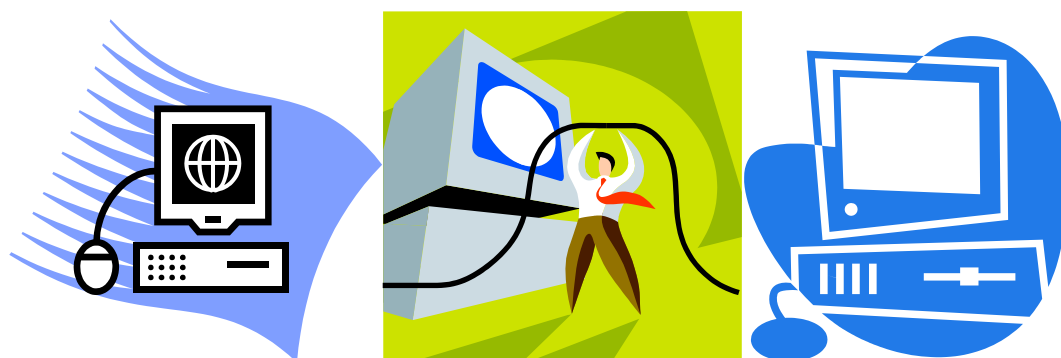


**O'ZBEKISTON RESPUBLIKASI
OLIY VA O'RTA MAXSUS TA'LIM VAZIRLIGI**

Samarqand davlat universiteti

Inatov A.I.

Algoritmlar va ma'lumotlar strukturasi



MA'RUZA MASHG'ULOTLARI

5110700 - Informatika o'qitish metodikasi

Samarqand-2019

Fanning o'quv uslubiy majmuasi ishchi o'quv reja va o'quv dasturiga muvofiq ishlab chiqildi

Tuzuvchi:

SamDU «Axborotlashtirish texnologiyalari» kafedrasida assistenti Inatov A.I.

Taqrizchilar:

SamDU, «Axborotlashtirish texnologiyalari» kafedrasida dotsenti Djumanov O.I.

SamDU «Axborotlashtirish texnologiyalari» kafedrasida k. o'qituvchisi Aminov I.B.

O'quv uslubiy majmua «Axborotlashtirish texnologiyalari» kafedrasining « » sentabr 2019 yildagi №1 yig'ilishida muhokama qilindi va fakultet ilmiy kengashida tasdiqlash uchun tavsiya qilindi.

Kafedra mudiri:

prof.I.I.Jumanov

***O'quv uslubiy majmua** “Amaliy matematika va informatika” fakulteti o'quv-uslubiy kengashining «___»_____ 2019 yildagi №1 yig'ilishida muhokama etildi va foydalanishga tavsiya qilindi.*

Fakultet uslubiy kengash raisi:_____ **Sh.Mamatov**

Fakultet kengashi raisi:

Babayarov A.I.

Kelishildi: O'quv uslubiy boshqarma boshlig'i:

Aliqulov B.

Ushbu ma'ruzlar matni fan o'qituvchisi tomonidan tayyorlangan o'quv-uslubiy majmuasidan olindi.

Annotatsiya

2002 yilda 30 mayda O'zbekiston Respublikasi Prezidenti tomonidan PF №3080 «Kompyuterlashtirish va axborot – kommunikasiya texnologiyalarini yanada rivojlantirish xaqidagi» qarori va «Kompyuterlashtirish va axborot – kommunikasiya texnologiyalarini 2002-2010 yillargacha rivojlantirish dasturi»ga asosan respublikamizda kompyuter va axborot texnologiyalarini rivojlantirish, ularni xalq xo'jaligida samarali qo'llash – dolzarb masalaga aylangan. Telekommunikasiya to'rlari, berilganlarni uzatish, Internet xizmatlariga kirish vositalari rivojlanib takomillashtirilmoqda.

Hozirgi kunda fan-texnikani jadal suratlar bilan rivojlanishi natijasida turli murakkab jarayonlarni, ularni matematik nuqtai nazardan tasavvur qilish, modellarini tuzish, algoritm va programma ta'minotini yaratish nafaqat nazariy jihatdan, balki amaliy jihatdan ham dolzarb bo'lgan muammolardan biri hisoblanadi. Fan nazariy va amaliy qismlardan iborat bo'lib, u Algoritmilar va ma'lumotlar strukturasi, C++ programmalash tilida programmalash asoslari, matematika masalalarini yechish uchun mo'ljallangan programmalar tuzish bo'yicha boshlang'ich tushunchalarini o'z ichiga olgan bo'limlaridan tashkil topgan.

«Algoritmilar va ma'lumotlar strukturasi» fani tabiiy fundamental fanlar majmuasiga taalluqli bo'lib, talabalar uni II semestrda o'rganishadi.

«Algoritmilar va ma'lumotlar strukturasi» fanining bosh maqsadi talabalarga qo'yilgan masalani yechuvchi kompyuter programmasini tuzishga o'rgatishdir. Bu maqsadda programmalash tillari haqida umumiy tushunchalar berish va bu tillardan foydalanishga o'rgatish hamda ixtisoslik fanlarini o'rganishga tayyorlashdan iborat. Berilgan masalalarni yechish uchun Algoritmilar va ma'lumotlar strukturasi qurish va ular asosida programma tuza olishbir qator fundamental bilimlarni muvaffaqiyatli o'zlashtirish uchun zarur bo'ladigan tayanch bilimlarni beradi.

«Algoritmilar va ma'lumotlar strukturasi» fanidan o'quv uslubiy majmua informatika o'qitish metodikasi ta'lim yo'nalishida bilim olayotgan 1-kurs talabalari uchun mo'ljallangan.

Sillabus

“Algoritmlar va ma’lumotlar strukturasi“ fanining qisqacha tavsifi

OTMning nomi va joylashgan manzili:	Samarqand davlat universiteti	Universitet hiyoboni, 15					
Kafedra:	Axborotlashtirish texnologiyalari	“Amaliy matematika va informatika” fakulteti tarkibida					
Ta’lim sohasi va yo’nalishi:	110000-pedagogika	5110700 - Informatika o‘qitish metodikasi					
Fanni (kursni) olib boradigan o‘qituvchi to‘g‘risida ma’lumot:	Inatov Abror Ismatovich	e-mail:	a-inatov@samdu.uz				
Dars vaqti va joyi:	Amali matematika va informatika o‘quv binosi 18 auditoriya	Kursning davomiy-ligi:	05.03.2020-4.07.2020				
Individual grafik asosida ishlash vaqti:	Dushanba, chorshanba va juma kunlari 14 ³⁰ dan 17 ⁰⁰ gacha						
Fanga ajratilgan soatlar	Auditoriya soatlari					Mustaqil ta’lim	96
	Ma’ruza	34	amaliyot	34	L		
Fanning boshqa fanlar bilan bog’liqligi (prerekvizitlari):	“Informatika”, Dasturlash asoslari, “Matematik tahlil”, “Algebra”, “Differensial tenglama”, “Umumiy fizika”.						

Fanning mazmuni

Fanning dolzarbligi va qisqacha mazmuni:	<u>Fanni o’qitishdan maqsad</u> Algoritmlar va ma’lumotlar strukturasi fanini o’qitishdan maqsad – informatika o’qituvchisining kasbiy sohasida egallashi lozim bo’lgan bilimlar va amalda qo’llash uchun ko’nikma va makalalarni shakllantirish va rivojlantirishdan iborat. Unda har bir kasb egasining faoliyati kerak bo’lgan tayanch nazariy va amaliy ma’lumotlarni o’z ichiga oladi. <u>Fanning vazifasi:</u> - algoritm tushunchasi va uning asosiy hossalari, ijrochilari, tasvirlash usullari, turlari, algoritmik tillar bilimlar bilan tanishtirish; -dasturlash tillari va ularning klassifikatsiyasi, dasturlash tilining alifbosi, buruqlar tizimi va operatorlari bilan tanishtirish va amaliy masalalarga dasturlar tuzishni o’rgatish; -obyektga yo’naltirilgan dasturlash tillari va ular
--	--

	ustida ishlashni o'rgatishdan iborat.
Talabalar uchun talablar	- o'qituvchiga va guruhdoshlarga nisbatan hurmat bilan munosabatda bo'lish; - universitet ichki tartib - intizom qoidalariga rioya qilish; - uyali telefonni dars davomida o'chirish; - berilgan uy vazifasi va mustaqil ish topshiriqlarini o'z vaqtida va sifatli bajarish; - ko'chirmachilik (plagiat) qat'iyan man etiladi; - darslarga qatnashish majburiy hisoblanadi, dars qoldirilgan holatda qoldirilgan darslar qayta o'zlashtirilishi shart; - darslarga oldindan tayyorlanib kelish va faol ishtirok etish; - talaba o'qituvchidan so'ng, dars xonasiga - mashg'ulotga kiritilmaydi; - talaba reyting ballidan norozi bo'lsa e'lon qilingan vaqtdan boshlab 1 kun mobaynida apellyasiya komissiyasiga murojat qilishi mumkin
Elektron pochta orqali munosabatlar tartibi	Professor-o'qituvchi va talaba o'rtasidagi aloqa elektron pochta orqali ham amalga oshirilishi mumkin, telefon orqali baho masalasi muhokama qilinmaydi, baholash faqatgina universitet hududida, ajratilgan xonalarda va dars davomida amalga oshiriladi. Elektron pochta ochish vaqti soat 15.00 dan 19.00 gacha

«Algoritmlar va ma'lumotlar strukturasi» mashg'ulotlarning mavzular va soatlar bo'yicha taqsimlanishi:

t/r	Mavzular nomi	Jami soat	Ma'ruza	Amaliy mashg'ulot	Laboratoriya	Mustaqil ta'lim
1.	Algoritmlar va ma'lumotlar strukturasi va ularning turlari	26	4	4	4	14
2.	Umumiy Algoritmlar va ma'lumotlar strukturasi nazariyasi	18	6	4	-	10
3.	Dasturlash tillari va ularning turlari	12	2	2	2	6
4.	Pascal dasturlash tili va uning imkoniyatlari	138	22	24	28	66
Jami		198	34	34	34	96

Baholashni 5 baholik shkaladan 100 ballik shkalaga o'tkazish

JADVALI

5 baholik shkala	100 ballik shkala	5 baholik shkala	100 ballik shkala	5 baholik shkala	100 ballik shkala
5,00 — 4,96	100	4,30 — 4,26	86	3,60 — 3,56	72
4,95 — 4,91	99	4,25 — 4,21	85	3,55 — 3,51	71
4,90 — 4,86	98	4,20 — 4,16	84	3,50 — 3,46	70
4,85 — 4,81	97	4,15 — 4,11	83	3,45 — 3,41	69
4,80 — 4,76	96	4,10 — 4,06	82	3,40 — 3,36	68
4,75 — 4,71	95	4,05 — 4,01	81	3,35 — 3,31	67
4,70 — 4,66	94	4,00 — 3,96	80	3,30 — 3,26	66
4,65 — 4,61	93	3,95 — 3,91	79	3,25 — 3,21	65
4,60 — 4,56	92	3,90 — 3,86	78	3,20 — 3,16	64
4,55 — 4,51	91	3,85 — 3,81	77	3,15 — 3,11	63
4,50 — 4,46	90	3,80 — 3,76	76	3,10 — 3,06	62
4,45 — 4,41	89	3,75 — 3,71	75	3,05 — 3,01	61
4,40 — 4,36	88	3,70 — 3,66	74	3,00	60
4,35 — 4,31	87	3,65 — 3,61	73	3,0 dan kam	60 dan kam

Asosiy adabiyotlar:	Abduqodirov A., Xaitov A., Shodiev R. Axborot texnologiyalari – T.: O'zbekiston, 2002 y. Aripov M. va boshqalar Informatika va informasion texnologiyalar. Oliy o'quv yurti talabalari uchun darslik T. 2005 y.
Qo'shimcha adabiyotlar:	1. Informatika. Prof. N. V. Makarova.-Toshkent; 2005. 2. В.М.Пестиков, А.Н.Маслобоев. Турбо Pascal 7.0. Изучаем на примерах. – С-Пт.: “БХВ-Петербург”, 2004. 3. П.Дарахвелидзе, Э.Марков. Программирование в Delphi7. Учебник.-СпБ-2003. 4. М. Ashurov, I. Mirmaxmudov, Sh. Sapaev Zamonaviy dasturlash tillari fanidan laboratoriya ishlari Toshkent 2008 5. В.Т.Безручко. Практикум по курсу информатики. «Финансы и статистика» М-2004.
Internet va Ziyonet saytlari	1. http:// www.nmarket.ru 2. www. pedagog. uz 3. www. Ziyonet. uz 4. www. edu. Uz 5. http://www.borlpasc.narod.ru/. 6. http:// www.intuit.ru

O'ZBEKISTON RESPUBLIKASI
OLIY VA O'RTA MAXSUS TA'LIM VAZIRLIGI
SAMARQAND DAVLAT UNIVERSITETI

“TASDIQLAYMAN”
O'quv ishlari bo'yicha prorektor
_____ prof. A.S. Soleev

“ ” _____ 2019 yil

ALGORITMLAR VA MA'LUMOTLAR STRUKTURASI
FANINING
ISHCHI O'QUV DASTURI

Bilim sohasi:	100000 –Gumanitar
Ta'lim sohasi:	110000 –Pedagogika
Ta'lim yo'nalishi:	5110700 – Informatika o'qitish metodikasi

Samarqand – 2019

Fanning ishchi o'quv dasturi o'quv, ishchi o'quv reja va o'quv dasturiga muvofiq ishlab chiqildi.

Tuzuvchilar:

Qobilov S.S. - SamDU «Axborotlashtirish texnologiyalari» kafedrasida dotsenti, t.f.n.

Inatov A.I. – SamDU «Axborotlashtirish texnologiyalari» kafedrasida assistenti

Taqrizchilar:

O'rinboyev E. O'. - SamDU «Matematik modellashtirish va kompleks dasturlash» kafedrasida dotsenti, t.f.n.;

Abdullayev A.N. - SamDU «Axborotlashtirish texnologiyalari» kafedrasida dotsenti, t.f.n.

Fanning ishchi o'quv dasturi SamDU “Axborotlashtirish texnologiyalari” kafedrasining 2019 yil “___” _____ dagi “___” - son yig'ilishida muhokamadan o'tgan va fakultet kengashida muhokama qilish uchun tavsiya etilgan.

Kafedra mudiri:_____ prof. I.I.Jumanov

Fanning ishchi o'quv dasturi “Amaliy matematika va informatika” fakulteti kengashida muhokama etilgan va foydalanishga tavsiya qilingan (2019 yil ___ avgustdagi ___-sonli bayonnoma).

Fakultet uslubiy kengash raisi:_____ Sh.Mamatov

Fakultet kengashi raisi: _____ A.I.Babayarovich

Kelishildi:

Kirish

Mazkur fan dasturi bakalavriat yo'nalishi: 5110700 – Informatika o'qitish metodikasi ta'lim yo'nalishida taxsil olayotgan talabalarning o'zlashtirishi lozim bo'lgan bilimlari va unga qo'yiladigan talablar asosida tuzilgan bo'lib, bo'lajak fan o'qituvchisi egallashi kerak bo'lgan bilimlar va ko'nikmalar majmuini o'z ichiga oladi:

- algoritmik tillar, algoritm va uning xossalari, algoritmnlarni tasvirlash usullari, rekursiya va iteratsiya, algoritmning murakkabligi tushunchasi, algoritm turlari, samarali algoritmnlar ishlab chiqishning asosiy usullari;
- steklar, navbatlar, daraxtlar, algoritmnlarning tahlili;
- algoritmnlarni saralash usullari, sodda saralash, tanlash usulida saralash, qo'shib saralash, almashish usulida saralash, saralashning Sheyker, Shella usullari, piramida usulida saralash, turnir usulida saralash;
- qidiruv usullari: binar qidiruv, Fibonachchi qidiruv, binar daraxt bo'yicha qidiruv, muvozanatlashtirilgan daraxt bo'yicha qidiruv, Rabin algoritmi, Boyer-Mur algoritmi, rekursiv algoritmnlar;
- pascal dasturlash tilli, dasturlash tilining alifbosi, kattaliklar va ularning tiplari, buruqlar tizimi va operatorlari, massivlar, chiziqli, tarmoqlanuvchi va takrorlanuvchi operatorlar, funktsiya va protseduralar, fayllar bilan ishlash, tilning grafik imkoniyatlari;

Fanning maqsad va vazifalari

Algoritmnlar va ma'lumotlar strukturasi fani umumkasbiy fanlar biri bo'lib, 2 semestrda o'qitiladi.

Algoritmnlar va ma'lumotlar strukturasi fanini o'qitishdan **maqsad** – informatika o'qituvchisining kasbiy sohasida egallashi lozim bo'lgan bilimlar va amalda qo'llash uchun kunikma va makalalarni shakllantirish va rivojlantirishdan iborat. Unda har bir kasb egasining faoliyati kerak bo'lgan tayanch nazariy va amaliy malumotlarni o'z ichiga oladi.

Algoritmnlar va ma'lumotlar strukturasi fanining **vazifasi**:

- algoritm tushunchasi va uning asosiy hossalari, ijrochilari, tasvirlash usullari, turlari, algoritmik tillar bilimlar bilan tanishtirish;
- dasturlash tillari va ularning klassifikatsiyasi, dasturlash tilining alifbosi, buruqlar tizimi va operatorlari bilan tanishtirish va amaliy masalalarga dasturlar tuzishni o'rgatish;
- obyektga yo'naltirilgan dasturlash tillari va ular ustida ishlashni o'rgatishdan iborat.

Fan bo'yicha talabaning malakasiga qo'yiladigan talablar

«Algoritmnlar va ma'lumotlar strukturasi» o'quv fanini o'zlashtirish jarayonida amalga oshiriladigan masalalar doirasida bakalavr:

- algoritm, va uning xossalari, algoritm ijrochilari, algoritmnlarni tasvirlash usullari, rekursiya va iteratsiya, algoritmning murakkabligi tushunchasi, algoritm turlari, samarali algoritmnlar ishlab chiqishning asosiy usullari, algoritmik tillar, dasturlash tillari va ularning klassifikatsiyasi, dasturlash tilining alifbosi, buyuqlar tizimi va operatorlarini, chiziqli, tarmoqlanuvchi va takrorlanuvchi dasturlar tuzishni, obyektga yo'naltirilgan dasturlash tillari haqidagi, obyektga yo'naltirilgan loyihalash; obyektlarni loyihalash: obyektga yo'naltirilgan dasturlash tili va unda dastur tuzishni **bilishlari kerak**;
- algoritmning asosiy hossalari, algoritm ijrochilari, algoritmnlarni tasvirlash usullari, algoritm turlarini amaliyotga tadbiq qilishi va ularga doir sodda masalalarni hal qila olishi, dasturlash tillari va ularning turlari, dasturlash tilining buyuqlar tizimi va operatorlarini qo'llagan holda chiziqli, tarmoqlanuvchi va takrorlanuvchi dasturlar tuza olish, dasturlashning obyektga

yo'naltirilgan paradigmasi, obyektga yo'naltirilgan loyihalash haqidagi masalalarni algoritmini tuza olish **ko'nikmalariga ega bo'lishlari kerak**;

Kasbiy sohasida uchraydigan turli hil masalalarga algoritmlar tuza olishi, algoritmnining turlarni farqlay olish, tasvirlash usullariga oid misollar keltira olish, rekursiya va iteratsiya, algoritmnining murakkabligi tushunchalarni ajrat olishi, samarali algoritmlar ishlab chiqishning asosiy usullari(balansirovka, dinamik dasturlash va boshqalar)ni amaliy qo'llay olishi, biror bir dasturlash tillari va ularning turlarini farqlay olish, dasturlash tillalari yordamida amaliy masallalarga dasturlar tuza olish, massivlar, grafik operatorlar, satriy kattaliklar bilan ishlash, funksiyalar va proseduralar, yozuvlar, ro'yxatlar, fayllar, modulli dasturlar haqidagi bilimlarni amalda qullay olish, obyektga yo'naltirilgan dasturlash tillaridan foydalana olish, boshqarish elementlari, oynalar, dialoglar; hodisalar va habarlar, obyektga yo'naltirilgan muhitlarda habarlarni uzatish va ularga ishlov berish, obyektlar iyerarxiyasi asosida dasturlarni loyihalash **malakalariga ega bo'lishlari kerak**.

O'quv rejadagi boshqa fanlar bilan bog'liqligi

Algoritmlar va ma'lumotlar strukturasi fani asosiy kasbiy fanlaridan biri hisoblanib, 2 semestrda o'qitiladi. Dasturni amalga oshirish o'quv rejasidagi rejalashtirilgan oliy matematika, fizika, informatika, malumotlar bazasi va ularni boshqarish tizimlari fanlaridan olingan nazariy va amaliy bilimlarga tayanadi.

Fanni o'qitishda zamonaviy axborot va pedagogik texnologiyalar

Talabalarning fanni muvafaqiyatli o'zlashtirishi uchun o'qitishning ilg'or va zamonaviy usullaridan foydalanish, yangi axborot-pedagogik texnologiyani tadbiq etish muhim ahamiyatga egadir. Fanni o'zlashtirishda darslik, o'quv va uslubiy qo'llanmalar, maruza matnlari, tarqatma materiallar, elektron materiallar, plakatlardan foydalaniladi.

Maruza va amaliy mashg'ulotlarda mos ravishda ilg'or pedagogik va kompyuter texnologiyalardan foydalaniladi.

O'quv jarayonida fanni o'tish sifatini belgilovchi quyidagi holatlar etiborga olinadi: yuqori ilmiy darajada dars berish, muammoli maruzalar o'qish, darslarni savol-javob tarzda qiziqarli tashkil qilish, ilg'or pedagogik texnologiyalardan va multimedia vositalardan foydalanish, tinglovchilarni undaydigan, o'ylantiradigan muammolarni ular oldiga quyish, erkin muloqot yuritishga, ilmiy izlanishga jalb qilish.

«Algoritmlar va ma'lumotlar strukturasi» kursini loyihalashtirishda quyidagi asosiy konseptual yondoshuvlardan foydalaniladi:

Shaxsga yo'naltirilgan talim. Bunda kelgusidagi mutaxassis faoliyati bilan bog'liq o'qitish, masalalar, mavzular ishchi dasturda ko'rilishi kerakligi nazarda tutilgan.

Tizimli yondoshuv. Ta'lim texnologiyasi tizimning barcha belgilarini o'zida mujassam etmog'i lozim: jarayonning mantiqiyliqi, uning barcha bo'g'inlarini o'zaro bog'langanligi, yaxlitligi.

Faoliyatga yo'naltirilgan yondoshuv. Mazkur dasturda kelgusidagi mutaxassis sifatlarini shakllantirish, aktivlashtirish va uning barcha qobiliyati va tashabbuskorligini ochishga etibor berilgan.

Dialogik yondoshuv. Fanning amaliyot darslarida shaxsning o'z-o'zini faollashtirish, o'zini ko'rsata olish kabi ijodiy faoliyatlarini rivojlantirish nazarda tutilgan.

Xamkorlikdagi ta'limni tashkil qilish. Talabalarning quyilgan masala yechimlarini olishda birgalikdagi ishlashni joriy etish zarurligi etiborga olingan.

Muammoli ta'lim. Talim oluvchi faoliyatini aktivlashtirish uchun fan dasturi bilan bog'liq qiziqarli mavzular muhokama qilinishligi, bunda ilmiy bilimning obektiv qarama-qarshiligi, uni hal etish usullari, amaliy faoliyatga ularni qo'llash masalalarni muhokama qilish ko'zda tutilgan.

Axborotni taqdim qilishning zamonaviy vositalari va usullarini qo'llash – yangi kompyuter va axborot texnologiyalarni o'quv jarayoniga qo'llash.

O'qitishning usullari va texnikasi. Ma'ruza (kirish, mavzuga oid, vizuallash), muammoli ta'lim, keys-stadi, pinbord, paradoks va loyihalash usullari, amaliy ishlar.

O'qitishni tashkil etish shakllari. Dialog, muloqot, xamkorlik, o'zaro o'rganishga asoslangan frontal, kollektiv va guruh.

O'qitish vositalari. Darslik, maruza matni, elektron kitob, elektron o'quv qo'llanmalar, elektron o'yinlar va shu bilan bir qatorda kompyuter va axborot texnologiyalari.

Kommunikasiya usullari. Tinglovchilar bilan operativ teskari aloqaga asoslangan bevosita o'zaro munosabatlar.

Teskari aloqa usullari va vositalari: kuzatish, blis-so'rov, oraliq, joriy, yakuniy nazorat tahlili.

Boshqarish usullari va vositalari: o'quv mashg'uloti bosqichlarini belgilab beruvchi texnologik xarita ko'rinishidagi o'quv mashg'ulotlarini rejalashtirish, quyilgan maqsadga erishishda o'qituvchi va tinglovchining birgalikdagi xarakati, auditoriya mashg'ulotlari va mustaqil ishlar nazorati.

Monitoring va baholash. Kurs ohirida test topshiriqlari yoki yozma ish varinatlarini bo'yicha talabalar bilimlari baholanadi.

Ayrim mavzular bo'yicha talabalar bilim baholash test asosida va kompyuter yordamida bajariladi. Internet tarmog'idagi rasmiy iqtisodiy ko'rsatkichlaridan foydalaniladi, tarqatma materiallar tayyorlanadi, tayanch so'z va iboralar asosida oraliq va yakuniy baholashlar o'tkaziladi.

“Algoritmlar va ma'lumotlar strukturasi” fanidan mashg'ulotlarning mavzular va soatlar bo'yicha taqsimlanishi:

t/r	Mavzular nomi	Jami soat	Ma'ruza	Amaliy mashg'ulot	Labora toriya	Mustaqil ta'lim
1.	Algoritmlar va ularning turlari	26	4	4	4	14
2.	Umumiy algoritmlar nazariyasi	20	6	4	-	10
3.	Dasturlash tillari va ularning turlari	12	2	2	2	6
4.	Pascal dasturlash tili va uning imkoniyatlari	140	22	24	28	66
Jami		198	34	34	34	96

Asosiy qism: Fanning uslubiy jihatdan uzviy ketma-ketligi

Asosiy qismda (ma'ruza) fanni mavzulari mantiqiy ketma-ketlikda keltiriladi. Har bir mavzuning mohiyati asosiy tushunchalar va tezislar orqali ochib beriladi. Bunda mavzu bo'yicha talabalarga DTS asosida yetkazilishi zarur bo'lgan bilim va ko'nikmalar to'la qamrab olinishi kerak. Asosiy qism sifatiga qo'yiladigan talab mavzularning dolzarbligi, ularning ish beruvchilar talablari va ishlab chiqarish ehtiyojlariga mosligi, mamlakatimizda bo'layotgan ijtimoiy-siyosiy va demokratik o'zgarishlar, iqtisodiyotni erkinlashtirish, iqtisodiy-huquqiy va boshqa sohalardagi islohatlarning ustuvor masalalarini qamrab olishi hamda fan va texnologiyalarning so'ngi yutuqlari e'tiborga olinishi tavsiya etiladi.

Ma'ruza mashg'ulotlari

Algoritmlar haqida umumiy ma'lumotlar. Algoritm haqida umumiy intuitiv tarif manosidagi tushuncha. Algoritmning kibernetik tarifi. Yevklid algoritmi. Algoritmning asosiy hossalari, ijrochilari, tasvirlash usullari, turlari. Algoritmning asosiy tiplari.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. Ma'ruza, namoyish etish, savol-javob, Charxpalak, metodlari.*

Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.

Algoritmning asosiy tiplari. Chiziqli, tarmoqlanuvchi va takrorlanuvchi algoritmlar.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim, namoyish etish, guruhlarda ishlash metodi.*

Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.

Algoritmlar nazariyasiga kirish. Umumiy algoritmlar nazariyasiga doir asosiy kashfiyotlar. Reduksiya prinsipi. Algoritm va hisoblash jarayoni orasidagi bog'lanish.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. Klaster, T-sxema, o'z-o'zini nazorat.*

Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.

Algoritmlarni baholash. Algoritmlarni baholash mezonlari va tahlil qilish usullari. Chiziqli ro'yhatlar va ular ustida asosiy amallar.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim, namoyish etish, interaktiv metodlar, kichik guruhlarda ishlash.*

Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.

Hisoblanuvchi funksiyalar va Tyuring tezisi. Tyuring mashinalari va EHMLar. Rekursiv funksiyalar. Rekursiya va Rekursiv funksiyalar.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. namoyish etish, kichik guruhlarda ishlash.*

Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.

Dasturlash tillari va ularning klassifikatsiyasi. Mashinaga mo'ljallangan va proseduraga mo'ljallangan dasturlash tillari, yuqori darjali dasturlash tillari, interpretatorlar va kompilyatorlar, dasturlarni translyasiyalash.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. Integrativ, munozara, o'z-o'zini nazorat, klaster.*

Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.

Pascal dasturlash tiliga kirish. Pascal tili sintaksisi. O'zgarmlar va o'zgaruvchilar. Ma'lumotlarning standart turlari. Fodalanuvchining turlari. Pascal dasturlash tilida dasturning tarkibiy qismi. Tarkibiy qismlarning xarakteristikasi. Nishonlarni, o'zgarmlarni, ma'lumotlarni, o'zgaruvchilarni, funksiya va proseduralarni, operatorlarni va izohlarni ifodalash bo'limlari.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. Klaster, Venn diagrammasi, T-sxemasi, o'z-o'zini nazorat.*

Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.

Pascal dasturlash tilida ifodalar, operatsiyalar va sodda operatorlar. Arifmetik ifodalar va operatsiyalar. Ma'lumotlarni kiritish-chiqarish operatorlari. Sodda operatorlar.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. namoyish etish, kichik guruhlarda ishlash.*

Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.

Tarmoqlanuvchi hisoblash jarayonlarini dasturlash. Mantiqiy amallar va ifodalar. Shartli va shartsiz o'tish operatorlari. Tanlash operatori.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim, namoyish etish, interaktiv metodlar, kichik guruhlarda ishlash.*

Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.

Takrorlanuvchi hisoblash jarayonlarini dasturlash. Parametrlilik operatorini tashkil etish. Parametrlilik takrorlanish jarayoni. Old shartli va so'ng shartli sikl operatorlari.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. Klaster, T-sxema, o'z-o'zini nazorat.*

Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.

Murakkab takrorlanish jarayonlarini dasturlash. Yig'indi va ko'paytmani, qator yig'indisini berilgan aniqlikda hisoblash algoritmi.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim, namoyish etish, interaktiv metodlar, kichik guruhlarda ishlash.*

Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.

Ma'lumotlarning skalyar turlari. Ma'lumotlarning chegaralangan turi. Ma'lumotlarning murakkab turi. Ma'lumotlarning muntazzam turi.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim, namoyish etish, interaktiv metodlar, kichik guruhlarda ishlash.*

Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.

Massivlar va ular bilan ishlash. Massiv turlarini ifodalash. Massiv elementlari ustida ishlash. Massiv elementlari ustida amallar. Bir o'lchovli va ikki o'lchovli massivlar ustida amallar.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim, namoyish etish, interaktiv metodlar, kichik guruhlarda ishlash.*

Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.

Satr tipidagi ma'lumotlar. Satrli ifodalar. Satrlar bilan ishlashda qo'llaniladigan prosedura va funksiyalar. Satrlarni ishlatilishiga misol.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. Ma'ruza, namoyish etish, savol-javob, Charxpalak, metodlari.*

Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.

Qism dasturlardan foydalanish . Prosedura va funksiyalar haqida umumiy ma'lumotlar. Maxsus qurilgan prosedura va funksiyalar.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. Klaster, T-sxema, o'z-o'zini nazorat.*

Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.

Foydalanuvchining prosedura va funksiyalari. Soxta va xaqiqiy parametrlar bilan ishlash. Lokal va global o'zgaruvchilar.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim, namoyish etish, interaktiv metodlar, kichik guruhlarda ishlash.*

Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.

Turbo Paskalning standart kutubxona modullari. Modul. Modul tushunchasi. Foydalanuvchi modulini yaratish. Ularning xarakteristikallari va xususiyatlari.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. Klaster, T-sxema, o'z-o'zini nazorat.*

Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.

CRT moduli. Ularning asosiy xarakteristikallari. Ekran adapterlarining xarakteristikallari va ularning imkoniyatlari.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. Klaster, T-sxema, o'z-o'zini nazorat.*

Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.

GRAPH moduli. GRAPH modulining imkoniyatlari va ulardan foydalanish. Koordinatalar, darchalar, ranglar, fon ranglari va obyektlar. Nuqta, to'g'ri chiziq, ko'pburchak, yoylar, aylanalar hosil qilish.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim, namoyish etish, interaktiv metodlar, kichik guruhlarda ishlash.*

Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.

“Algoritmlar va ma'lumotlar strukturasi” fani bo'yicha ma'ruza mashg'ulotlarining kalendar tematik rejasi

№	Maruza mavzusi	Ajratilgan soat
1	2	3
II-SEMESTR		
	Algoritm va uning turlari	
1.	Algoritmlar haqida umumiy tushunchalar. Algoritm tushunchasi, asosiy xossalari. Algoritm ijrochisi. Algoritmlarni tasvirlash usullari. Algoritm turlari	2
2.	Algoritmik tillar va algoritmlarni ishlab chiqish. Algoritmning murakkabligi. Rekursiya. Iteratsiya. Algoritmni ishlab chiqish usullari.	2

	Algoritmik tillar	
	Algoritmlar umumiy nazariyasi	
3.	Algoritm nazariyasining umumiy tushunchalari. Algoritm nazariyasiga kirish. Reduksiya prinsipi. Algoritm va hisoblash jaroyoni orasidagi bog'lanish	2
4.	Algoritmlarni baholash. Algoritmlarni baholash, ularni mezonlari va tahlil qilish. Ikki tamonlama algoritm.	2
5.	Ro'yxat va daraxtlar. Chiziqli ro'yxatlar va ular ustida amallar. Tarmoqlar. Daraxtlar va ularning turlari. Saralash va izlash algoritmlari. Tanlash va joylash algoritmlari. Izlashning asosiy prinsiplari. Izlash usullari. Saralash algoritmlarini taqqoslash	2
	Dasturlash tillari va ularning turlari	
6.	Dasturlash tillari va ularning klassifikatsiyasi. Mashinaga mo'ljallangan va proseduraga mo'ljallangan dasturlash tillari, yuqori darjali dasturlash tillari, interpretatorlar va kompilyatorlar, dasturlarni translyasiyalash.	2
	Paskal dasturlash tili va uning imkoniyatlari	
7.	Pascal dasturlash tiliga kirish. Pascal tili sintaksisi. O'zgarmlar va o'zgaruvchilar. Ma'lumotlarning standart turlari. Fodalanuvchining turlari. Pascal dasturlash tilida dasturning tarkibiy qismi Tarkibiy qismlarning xarakteristikasi. Nishonlarni, o'zgarmlarni, ma'lumotlarni, o'zgaruvchilarni, funksiya va proseduralarni, o'preatorlarni va izohlarni ifodalash bo'limi	2
8.	Pascal dasturlash tilida ifodalar, operatsiyalar va sodda operatorlar Arifmetik ifodalar va operatsiyalar. Ma'lumotlarni kiritish-chiqarish operatorlari. Sodda operatorlar.	2
9.	Tarmoqlanuvchi hisoblash jarayonlarini dasturlash Mantiqiy amallar va ifodalar. Shartli va shartsiz o'tish operatorlari. Tanlash operatori.	2
10.	Takrorlanuvchi hisoblash jarayonlarini dasturlash Parametrli sikl operatorini tashkil etish. Parametrli takrorlanish jarayoni. Old shartli va so'ng shartli sikl operatorlari	2
11.	Massivlar va ular bilan ishlash Massiv turlarini ifodalash. Massiv elementlari ustida ishlash. Massiv elementlari ustida amallar. Bir o'lchovli va ikki o'lchovli massivlar ustida amallar.	2
12.	Satr tipidagi ma'lumotlar Satrli ifodalar. Satrlar bilan ishlashda qo'llaniladigan prosedura va funksiyalar. Satrlarni ishlatilishiga misol.	2
13.	Qism dasturlardan foydalanish Prosedura va funksiyalar haqida umumiy ma'lumotlar. Maxsus qurilgan prosedura va funksiyalar.	2
14.	Foydalanuvchining prosedura va funksiyalari Soxta va xaqiqiy parametrlar bilan ishlash. Lokal va global o'zgaruvchilar.	2
15.	Paskalda ma'lumotlarning faylli turlari Faylli turlarning ifodalanishi. Fayllarni qayta ishlash vositalari. Tiplashgan fayllar.	2
16.	Turbo Paskalda modullar. Modul. Modul tushunchasi. Foydalanuvchi modulini yaratish. Ularning xarakteristikalar va xususiyatlari.	2
17.	GRAPH moduli	2

	GRAPH modulining imkoniyatlari va ulardan foydalanish. Koordinatalar, darchalar, ranglar, fon ranglari va obyektlar. Nuqta, to'g'ri chiziq, ko'pburchak, yoylar, aylanalar hosil qilish.	
JAMI:		34

Amaliy mashg'ulotlarning tavsiya etiladigan mavzulari

Amaliyot ishlari mavzulari talabalarda berilgan amaliy masalalarni yechishda algortmlar va dasturlar tuzish amaliy ko'nikmalarni shakllantirishga qaratilgan.

Algoritmning asosiy tiplari: chiziqli, tarmoqlanuvchi va takrorlanuvchi algortmlar.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. Klaster, T-sxema, o'z-o'zini nazorat.*

Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.

Tanlash va joylashtirish turkumida murakkablikdagi saralash algortmlari.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim, namoyish etish, savol-javob, Charxpalak, metodlari.*

Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.

Samarali mashina algortmlarini tuzish usullari. Rekursiya va itarasiya, algortmlar murakkabligi tushunchasi.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim, namoyish etish, interaktiv metodlar, kichik guruhlarda ishlash.*

Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.

Dasturlash tillari va ularning klassifikasiyasini o'rganish.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. Klaster, T-sxema, o'z-o'zini nazorat.*

Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.

Paskal dasturlash tilida chiziqli dastur tuzish.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. Klaster, T-sxema, o'z-o'zini nazorat.*

Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.

Mantiqiy amallar. Mantiqiy ifodalarning geometrik ma'nosi. Mantiqiy ifodalar ustida amallar bajarish.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim, namoyish etish, interaktiv metodlar, kichik guruhlarda ishlash.*

Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.

Tarmoqlanuvchi hisoblash jarayonlariga doir dasturlar tuzish.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. Klaster, T-sxema, o'z-o'zini nazorat.*

Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.

Takrorlanuvchi hisoblash jarayonlariga doir dasturlar tuzish.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim, namoyish etish, savol-javob, Charxpalak, metodlari.*

Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.

Murakkab takrorlanish jarayonlarini dasturlash.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim, namoyish etish, interaktiv metodlar, kichik guruhlarda ishlash.*

Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.

Pascalda satrli kattaliklarga doir dasturlar tuzish.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim, namoyish etish, interaktiv metodlar, kichik guruhlarda ishlash.*

Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.

Pascalda massivlarga doir dasturlar tuzish.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim, namoyish etish, interaktiv metodlar, kichik guruhlarda ishlash.*

Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.
Pascaldagi satrli kattaliklarga doir dasturlar tuzish.
Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. Klaster, T-sxema, o'z-o'zini nazorat.*

Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.
Pascaldagi qism dasturlar, funksiyaga doir dasturlar tuzish.
Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim, namoyish etish, interaktiv metodlar, kichik guruhlarda ishlash.*

Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.
Pascaldagi qism dasturlar, proseduraga doir dasturlar tuzish.
Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. namoyish etish, kichik guruhlarda ishlash.*

Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.
CRT moduli qism dasturlaridan foydalanish.
Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim.*

Integrativ, munozara, o'z-o'zini nazorat, klaster.

Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.
Pascaldagi grafik operatorlarga doir dasturlar tuzish va natija olish.
Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. Klaster, Venn diagrammasi, T-sxemasi, o'z-o'zini nazorat.*

Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.
Pascaldagi to'plamlarga doir dasturlar tuzish.
Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim.*

Integrativ, munozara, o'z-o'zini nazorat, klaster.

Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.
Pascaldagi yozuvlarga doir dasturlar tuzish.
Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. B/B/B jadvali, munozara, T-sxema, o'z-o'zini nazorat*

Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.
Tiplashgan fayllarga doir dasturlar tuzish.
Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. namoyish etish, kichik guruhlarda ishlash.*

Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.
Matnli fayllarga doir dasturlar tuzish.
Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim, namoyish etish, interaktiv metodlar, kichik guruhlarda ishlash.*

Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.

**“Algoritmlar va ma'lumotlar strukturasi” fani bo'yicha
amaliy mashg'ulotlarning kalendar tematik rejasi**

№	Mashg'ulot mavzusi	Ajratil-gan soat
II-SEMESTR		
1.	Algoritmning asosiy tiplari: chiziqli, tarmoqlanuvchi, takrorlanuvchi algoritmlar	2
2.	Ro'yxat va daraxtlar. Chiziqli ro'yxatlar va ular ustida amallar. Tarmoqlar. Daraxtlar va ularning turlari	2
3.	Tanlash va joylashtirish turkumida murakkablikdagi saralash algoritmlari	2
4.	Samarali mashina algoritmlarini tuzish usullari. Rekursiya va itarasiya, algoritmlar murakkabligi tushunchasi.	2
5.	Dasturlash tillari va ularning klassifikatsiyasini o'rganish.	2

6.	Pascal tilida ma'lumotlarning turlari va standart funksiyalar	2
7.	Sodda operatorlar va chiziqli dastur tuzish	2
8.	Mantiqiy amallar. Mantiqiy ifodalarning geometrik ma'nosi. Mantiqiy ifodalar ustida amallar bajarish.	2
9.	Tarmoqlanuvchi hisoblash jarayonlariga doir dasturlar tuzish	2
10.	Takrorlanuvchi hisoblash jarayonlariga doir dasturlar tuzish	2
11.	Murakkab takrorlanish jarayonlarini dasturlash	2
12.	Pascalda massivlarga va satrlarga doir dasturlar tuzish	2
13.	Pascalda qism dasturlar, funksiyaga doir dasturlar tuzish	2
14.	Pascalda qism dasturlar, proseduraga doir dasturlar tuzish	2
15.	Pascalda to'plamlarga va yozuvlarga doir dasturlar tuzish	2
16.	Standart modullar. CRT moduli qism dasturlaridan foydalanish. Foydalanuvchi modullari va ularni ishlab chiqish	2
17.	Pascalda grafik operatorlarga doir dasturlar tuzish va natija olish	2
JAMI:		34

Laboratoriya mashg'ulotlari

Asosiy algoritmik konstruksiyalar bilan ishlash. Chiziqli algoritmlar tuzish.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. Klaster, Venn diagrammasi, T-sxemasi, o'z-o'zini nazorat.*

Adabiyotlar: A1; A2; A3; A5; Q1; Q2; Q3; Q4; Q5.

Asosiy algoritmik konstruksiyalar bilan ishlash. Tarmoqlanuvchi algoritmlar tuzish.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. Klaster, Venn diagrammasi, T-sxemasi, o'z-o'zini nazorat.*

Adabiyotlar: A1; A2; A4; A5; Q1; Q2; Q3; Q4; Q5.

Asosiy algoritmik konstruksiyalar bilan ishlash. Takrorlanuvchi algoritmlar tuzish.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. Klaster, Venn diagrammasi, T-sxemasi, o'z-o'zini nazorat.*

Adabiyotlar: A1; A2; A3; A4; A5; Q2; Q3; Q4; Q5.

Pascalda chiziqli dastur tuzish va natija olish.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. namoyish etish, kichik guruhlarda ishlash.*

Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3;.

Pascal dasturlash tilida tarmoqlanuvchi dasturlar tuzish va natija olish.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. B/B/B jadvali, munozara, T-sxema, o'z-o'zini nazorat*

Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.

Pascalda takrorlanuvchi tarkibli dasturlar tuzish va natijalar olish.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. B/B/B jadvali, munozara, T-sxema, o'z-o'zini nazorat*

Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.

Pascalda murakkab takrorlanish tarkibli dasturlar tuzish va natijalar olish.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim.*

Integrativ, munozara, o'z-o'zini nazorat, klaster.

Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.

Pascalda satrli kattaliklarga doir dasturlar tuzish va natija olish.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. Klaster, Venn diagrammasi, T-sxemasi, o'z-o'zini nazorat.*

Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.

Bir o'lchovli massivlarga doir dasturlar tuzish va natija olish.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. Klaster, Venn diagrammasi, T-sxemasi, o'z-o'zini nazorat.*

Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.

Ikki o'lchovli massivlarga doir dasturlar tuzish va natija olish.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. Klaster, Venn diagrammasi, T-sxemasi, o'z-o'zini nazorat.*

Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.

Pascalda qism dasturlar, funksiya va proseduralarga doir dasturlar tuzish va natija olish.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. namoyish etish, kichik guruhlarda ishlash.*

Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.

Pascalda grafik operatorlarga doir dasturlar tuzish va natija olish.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. namoyish etish, kichik guruhlarda ishlash.*

Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.

To'plamlar va yozuvlarga doir dasturlar tuzish va natija olish.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. B/B/B jadvali, munozara, T-sxema, o'z-o'zini nazorat*

Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.

Fayllar bilan ishlashga doir dasturlar tuzish va natija olish.

Qo'llaniladigan ta'lim texnologiyalari: *dialogik yondoshuv, muammoli ta'lim. B/B/B jadvali, munozara, T-sxema, o'z-o'zini nazorat*

Adabiyotlar: A1; A2; A3; A4; A5; Q1; Q2; Q3; Q4; Q5.

“Algoritmash va dasturlash tillari” fani bo'yicha laboratoriya mashg'ulotlarining kalendar tematik rejasi

№	Laboratoriya mashg'ulotlarining mavzusi	Ajratil-gan soat
1	2	3
II-SEMESTR		
1.	Asosiy algoritmik konstruksiyalar va ma'lumotlar strukturasi bilan tanishish	4
2.	Saralash algoritmlar va ularning tahlili: Algoritmlar tahlili. Algoritmlarni saralash usullari, Sodda saralash. Tanlash usulida saralash. Qo'shib saralash. Almashish usulida saralash.	8
3.	Saralashning boshqa usullari va qidirish algoritmlari: Sheyker va Shell usulida saralash. Piramida va Turnir usulida saralash. Binar, Fibonachchi va Binar daraxt bo'yicha saralash. Muvozanatlashtirilgan daraxt bo'yicha qidiruv.	4
4.	Chiziqli, tarmoqlanuvchi va takrorlanuvchi jarayonlarni dasturlash	6
6.	Pascal tilida murakkab turlar bilan ishlash. Pascalda qismdasturlar va grafika	12
JAMI		34

Mustaqil ta'lim tashkil etishning shakli va mazmuni

“Algoritmlar va ma'lumotlar strukturasi” bo'yicha talabani mustaqil ta'limi shu fanni o'rganish jarayonining tarkibiy qismi bo'lib, uslubiy va axborot resurslari bilan to'la ta'minlangan.

Talabalar auditoriya mashg'ulotlarida professor-o'qituvchilarning ma'ruzasini tinglaydilar, misol va masalalar yechadilar. Auditoriyadan tashqarida talaba darslarga tayyorlanadi, adabiyotlarni konspekt qiladi, uy vazifa sifatida berilgan masalalarni hal qiladi.

Bundan tashqari ayrim mavzularni kengroq o'rganish maqsadida qo'shimcha adabiyotlarni o'qib referatlar tayyorlaydi hamda ayrim mavzular bo'yicha testlar yechadi. Mustaqil ta'lim natijalari reyting tizimi asosida baholanadi.

Uyga vazifalarni bajarish, qo'shimcha darslik va adabiyotlardan yangi bilimlarni mustaqil o'rganish, kerakli ma'lumotlarni izlash va ularni topish yo'llarini aniqlash, internet tarmoqlaridan foydalanib ma'lumotlar to'plash va ilmiy izlanishlar olib borish, ilmiy to'garak doirasida yoki mustaqil ravishda ilmiy manbalardan foydalanib ilmiy maqola va ma'ruzalar tayyorlash kabilar talabalarning darsda olgan bilimlarini chuqurlashtiradi, ularning mustaqil fikrlash va ijodiy qobiliyatini rivojlantiradi. Shuning uchun ham mustaqil ta'limsiz o'quv faoliyati samarali bo'lishi mumkin emas.

Uy vazifalarini tekshirish va baholash amaliy mashg'ulot olib boruvchi o'qituvchi tomonidan, konspektlarni va mavzuni o'zlashtirish darajasini tekshirish va baholash esa ma'ruza darslarini olib boruvchi o'qituvchi tomonidan har darsda amalga oshiriladi.

“Algoritmlar va ma'lumotlar strukturasi” fanidan mustaqil ish majmuasi fanning barcha mavzularini qamrab olgan va quyidagi 17 ta katta mavzu ko'rinishida shakllantirilgan.

Talabalar mustaqil ta'limining mazmuni va hajmi

№	Mustaqil ta'lim mavzulari	Berilgan topshiriqlar	Bajarish muddati	Hajmi (soatda)
II semestr				
1.	Algoritmning berilish usullari va ularga oid misollar	Adabiyotlardan konspekt. Individual topshiriqlar	1 - hafta	4
2.	Algoritmning turlariga oid misollar	Adabiyotlardan konspekt. Individual topshiriqlar	2 - hafta	6
3.	Murakkab turdagi algiritmlar va ularning qo'llanilishi	Adabiyotlardan konspekt. Individual topshiriqlar	3 - hafta	4
4.	Dasturlash tillari va ularning kelib chiqish tarixi	Adabiyotlardan konspekt. Individual topshiriqlar	4 - hafta	6
5.	Dasturlash tillarining turlari va bir biridan farqi	Adabiyotlardan konspekt. Individual topshiriqlar	5 - hafta	4
6.	Pascal dasturlash tilining kelib chiqish va qo'llanishi	Adabiyotlardan konspekt. Individual topshiriqlar	6 - hafta	6
7.	Pascal dasturlash tilidan chiziqli va tarmoqlanuvchi dasturlar tuzish va ularni tahlil qilish	Adabiyotlardan konspekt. Individual topshiriqlar	7 - hafta	6
8.	Pascal dasturlash tilida takrorlanuvchi dasturlar va murakkab sikl. a) Dars jadvalini ciqaruvchi. b) Alfavit bo'yicha saralovchi dastur. c) Test topshiriqlarini bajaruvchi (To'g'ri javobni belgilovchi) dasturlar tuzing	Adabiyotlardan konspekt. Individual topshiriqlar	8 - hafta	6
9.	Qism dastur. Massivlar. Mustaqil ravishda 10 tadan dastur tuzing va uning bajarilishini asoslab bering.	Adabiyotlardan konspekt. Individual topshiriqlar	9 – hafta	6
10.	Satriy kattalik bilan ishlash. Mustaqil ravishda 10 tadan dastur tuzing va uning bajarilishini asoslab	Adabiyotlardan konspekt. Individual topshiriqlar	10 – hafta	6

	bering.			
11.	Pascal dasturlash tilida grafik imkoniyatlar va ulardan foydalanish. Figuralarni harakatga keltiruvchi dasturlar tuzish.	Adabiyotlardan konspekt. Individual topshiriqlar	11– hafta	6
12.	Bir necha xil Protsedura va funksiyalarning qo‘llanishiga doir misollar.	Adabiyotlardan konspekt. Individual topshiriqlar	12 – hafta	6
13.	Sodda, Shartli va skilli dasturlar tuzish.	Adabiyotlardan konspekt. Individual topshiriqlar	13 – hafta	6
14.	Tasodifiy sonlar, Massivlar bilan ishlash. Sana-vaqt turi, Xarfiy kattaliklar bilan ishlash	Adabiyotlardan konspekt. Individual topshiriqlar	14 – hafta	6
15.	To‘plamlar, fayllar bilan ishlash	Adabiyotlardan konspekt. Individual topshiriqlar	15 – hafta	6
16.	Funksiya va protseduralarni yaratish	Adabiyotlardan konspekt. Individual topshiriqlar	16 – hafta	6
17.	Grafik uskunalar bilan ishlash. Nazorat qiluvchi dasturlar yaratish	Adabiyotlardan konspekt. Individual topshiriqlar	17 – hafta	6
Jami				96

Dasturning informasion -uslubiy ta’minoti

Mazkur fanni o‘qitish jarayonida ta’limning zamonaviy metodlari, pedagogik va axborot-kommunikatsiya texnologiyalarini qo‘llash nazarda tutilgan.

- ma’ruza darslarida zamonaviy kompyuter texnologiyalari yordamida prezentatsiya va elektron-didaktik texnologiyalaridan;

- amaliy mashg‘ulotlarda zamonaviy pedagogik va inovatsion texnologiyalardan;

- laboratoriya mashg‘ulotlarida zamonaviy kompyuter sinflaridan foydalanish ko‘zda tutiladi. Shuningdek, bugungi kun talabiga javob beradigan dasturlash tillari Pascal, Delphi tillarini o‘rnatuvchi disklar ham bo‘lishi lozim..

“Algoritmlar va ma’lumotlar strukturasi” fanidan talabalar bilimni reyting tizimi asosida baholash mezonlari

“Algoritmlar va ma’lumotlar strukturasi” fanidan talabalar bilimni baholash O‘zbekiston Respublikasi Oliy va o‘rta maxsus ta’lim vazirining 2018 yil 9 avgustdagi 19-2018-son buyrug‘i bilan tasdiqlangan **“Oliy ta’lim muassasalarida talabalar bilimni nazorat qilish va baholash tizimi to‘g‘risida”**gi Nizom asosida amalga oshiriladi.

Mazkur fandan talabalar bilimni nazorat qilish **oralik va yakuniy nazorat** turlarini o‘tkazish orqali amalga oshiriladi.

Talabalar bilimni baholash **5 baholik tizimda** amalga oshiriladi.

Talabalarning bilimi quyidagi mezonlar asosida:

- talaba mustaqil xulosa va qaror qabul qiladi, ijodiy fikrlay oladi, mustaqil mushohada yuritadi, olgan bilimni amalda qo‘llay oladi, fanning (mavzuning) mohiyatini tushunadi, biladi, ifodalay oladi, aytib beradi hamda fan (mavzu) bo‘yicha tasavvurga ega deb topilganda — 5 (a‘lo) baho;
- talaba mustaqil mushohada yuritadi, olgan bilimni amalda qo‘llay oladi, fanning (mavzuning) mohiyatni tushunadi, biladi, ifodalay oladi, aytib beradi hamda fan (mavzu) bo‘yicha tasavvurga ega deb topilganda — 4 (yaxshi) baho;
- talaba olgan bilimni amalda qo‘llay oladi, fanning (mavzuning) mohiyatni tushunadi, biladi, ifodalay oladi, aytib beradi hamda fan (mavzu) bo‘yicha tasavvurga ega deb topilganda — 3 (qoniqarli) baho;

- talaba fan dasturini o‘zlashtirmagan, fanning (mavzuning) mohiyatini tushunmaydi hamda fan (mavzu) bo‘yicha tasavvurga ega emas deb topilganda — 2 (qoniqarsiz) baho bilan baholanadi.
-

Baholashni 5 baholik shkaladan 100 ballik shkalaga o‘tkazish

JADVALI

5 baholik shkala	100 ballik shkala	5 baholik shkala	100 ballik shkala	5 baholik shkala	100 ballik shkala
5,00 — 4,96	100	4,30 — 4,26	86	3,60 — 3,56	72
4,95 — 4,91	99	4,25 — 4,21	85	3,55 — 3,51	71
4,90 — 4,86	98	4,20 — 4,16	84	3,50 — 3,46	70
4,85 — 4,81	97	4,15 — 4,11	83	3,45 — 3,41	69
4,80 — 4,76	96	4,10 — 4,06	82	3,40 — 3,36	68
4,75 — 4,71	95	4,05 — 4,01	81	3,35 — 3,31	67
4,70 — 4,66	94	4,00 — 3,96	80	3,30 — 3,26	66
4,65 — 4,61	93	3,95 — 3,91	79	3,25 — 3,21	65
4,60 — 4,56	92	3,90 — 3,86	78	3,20 — 3,16	64
4,55 — 4,51	91	3,85 — 3,81	77	3,15 — 3,11	63
4,50 — 4,46	90	3,80 — 3,76	76	3,10 — 3,06	62
4,45 — 4,41	89	3,75 — 3,71	75	3,05 — 3,01	61
4,40 — 4,36	88	3,70 — 3,66	74	3,00	60
4,35 — 4,31	87	3,65 — 3,61	73	3,0 dan kam	60 dan kam

Tavsiya etilgan adabiyotlar ro‘yxati

Asosiy darslik

- Abduqodirov A., Xaitov A., Shodiev R. Axborot texnologiyalari – T.: O‘zbekiston, 2002 y.
- Aripov M. va boshqalar Informatika va informasion texnologiyalar. Oliy o‘quv yurti talabalari uchun darslik T. 2005 y.

Qo‘shimcha adabiyotlar

- Informatika. Prof. N. V. Makarova.-Toshkent; 2005.
- В.М.Пестиков, А.Н.Маслобоев. Турбо Pascal 7.0. Изучаем на примерах. – С-Пт.: “БХВ-Петербург”, 2004.
- П.Дарахвелидзе, Э.Марков. Программирование в Delphi7. Учебник.-СПБ-2003.
- M. Ashurov, I. Mirmaxmudov, Sh. Sapaev Zamonaviy dasturlash tillari fanidan laboratoriya ishlari Toshkent 2008
- В.Т.Безручко. Практикум по курсу информатики. «Финансы и статистика» М-2004.

Elektron ta’lim resurslari

- [http:// www.nmarket.ru](http://www.nmarket.ru)
- [www. pedagog. uz](http://www.pedagog.uz)
- [www. Ziyonet. uz](http://www.Ziyonet.uz)
- [www. edu. Uz](http://www.edu.Uz)
- [http://www.borlpasc.narod.ru/.](http://www.borlpasc.narod.ru/)
- [http:// www.intuit.ru](http://www.intuit.ru)

Kalendar tematik reja

“Algoritmlar va ma’lumotlar strukturasi” fanidan ma’ruza mashg’ulotlari mavzularining kalendar ish rejasi

№	Ma’ruza mavzusi	Soat	Sana
1	2	3	4
	II - semestr		
	Algoritm va uning turlari		
1.	Algoritmlar va ma’lumotlar strukturasi haqida umumiy tushunchalar. Algoritm tushunchasi, asosiy xossalari. Algoritm ijrochisi. Algoritmlar va ma’lumotlar strukturasi tasvirlash usullari. Algoritm turlari	2	
2.	Algoritmik tillar va Algoritmlar va ma’lumotlar strukturasi ishlab chiqish. Algoritmning murakkabligi. Rekursiya. Iteratsiya. Algoritmni ishlab chiqish usullari. Algoritmik tillar	2	
	Algoritmlar va ma’lumotlar strukturasi umumiy nazariyasi		
3.	Algoritm nazariyasining umumiy tushunchalari. Algoritm nazariyasiga kirish. Reduksiya prinsipi. Algoritm va hisoblash jaroyoni orasidagi bog’lanish	2	
4.	Algoritmlar va ma’lumotlar strukturasi baholash. Algoritmlar va ma’lumotlar strukturasi baholash, ularni mezonlari va tahlil qilish. Ikki tamonlama algoritm. Ro’yxat va daraxtlar. Chiziqli ro’yxatlar va ular ustida amallar. Tarmoqlar. Daraxtlar va ularning turlari	2	
5.	Saralash va izlash Algoritmlar va ma’lumotlar strukturasi. Tanlash va joylash Algoritmlar va ma’lumotlar strukturasi. Izlashning asosiy prinsiplari. Izlash usullari. Saralash Algoritmlar va ma’lumotlar strukturasi taqqoslash	2	
	Dasturlash tillari va ularning turlari		
6.	Dasturlash tillari va ularning klassifikatsiyasi. Mashinaga mo’ljallangan va proseduraga mo’ljallangan dasturlash tillari, yuqori darjali dasturlash tillari, interpretatorlar va kompilyatorlar, dasturlarni translyatsiyalash.	2	
	Paskal dasturlash tili va uning imkoniyatlari		
7.	Paskal dasturlash tiliga kirish. Paskal tili sintaksisi. O’zgarmlar va o’zgaruvchilar. Ma’lumotlarning standart turlari. Fodalanuvchining turlari. Paskal dasturlash tilida dasturning tarkibiy qismi. Tarkibiy qismlarning xarakteristikasi. Nishonlarni, o’zgarmlarni, ma’lumotlarni, o’zgaruvchilarni, funktsiya va proseduralarni, operatorlarni va izohlarni ifodalash bo’limi	2	
8.	Paskal dasturlash tilida ifodalar, operatsiyalar va sodda	2	

	operatorlar Arifmetik ifodalar va operasiyalar. Ma'lumotlarni kiritish-chiqarish operatorlari. Sodda operatorlar.		
9.	Tarmoqlanuvchi hisoblash jarayonlarini dasturlash Mantiqiy amallar va ifodalar. Shartli va shartsiz o'tish operatorlari. Tanlash operatori.	2	
10.	Takrorlanuvchi hisoblash jarayonlarini dasturlash Parametrli sikl operatorini tashkil etish. Parametrli takrorlanish jarayoni. Old shartli va so'ng shartli sikl operatorlari	2	
11.	Massivlar va ular bilan ishlash Massiv turlarini ifodalash. Massiv elementlari ustida ishlash. Massiv elementlari ustida amallar. Bir o'lchovli va ikki o'lchovli massivlar ustida amallar.	2	
12.	Satr tipidagi ma'lumotlar Satrli ifodalar. Satrlar bilan ishlashda qo'llaniladigan prosedura va funksiyalar. Satrlarni ishlatilishiga misol.	2	
13.	Qism dasturlardan foydalanish Prosedura va funksiyalar haqida umumiy ma'lumotlar. Maxsus qurilgan prosedura va funksiyalar.	2	
14.	Foydalanuvchining prosedura va funksiyalari Soxta va xaqiqiy parametrlar bilan ishlash. Lokal va global o'zgaruvchilar.	2	
15.	Paskalda ma'lumotlarning faylli turlari Faylli turlarning ifodalanishi. Fayllarni qayta ishlash vositalari. Tiplashgan fayllar.	2	
16.	Turbo Paskalda modullar. Modul. Modul tushunchasi. Foydalanuvchi modulini yaratish. Ularning xarakteristikalar va xususiyatlari.	2	
17.	GRAPH moduli GRAPH modulining imkoniyatlari va ulardan foydalanish. Koordinatalar, darchalar, ranglar, fon ranglari va obyektlar. Nuqta, to'g'ri chiziq, ko'pburchak, yoylar, aylanalar hosil qilish.	2	
Jami:		34	

MODULNI O'QITISHDA FOYDALANILADIGAN INTERFAOL TA'LIM METODLARI

“SWOT-tahlil” metodi

Metodning maqsadi: mavjud nazariy bilimlar va amaliy tajribalarni tahlil qilish, taqqoslash orqali muammoni hal etish yo'llarini topishga, bilimlarni mustahkamlash, takrorlash, baholashga, mustaqil, tanqidiy fikrlashni, nostandart tafakkurni shakllantirishga xizmat qiladi.

S-(strength)	• Kuchli tomonlari
W – (weakness)	• Zaif, kuchsiz tomonlari
O – (opportunity)	• Imkoniyatlari
T – (threat)	• To'siqlar

Namuna: Windows OT ning SWOT tahlilini ushbu jadvalga tushiring.

S	Windows OT dan foydalanishning kuchli tomonlari	Ushbu operatsion tizimli dasturlash tili bo'lib, boshqa dasturlash tillarida ham tizimli masalalarni yechishda a murojaat etiladi.
W	Windows OT dan foydalanishning kuchsiz tomonlari	Windows OT ning strukturasi murakkabligi.
O	Windows OT dan foydalanishning imkoniyatlari (ichki)	Windows OT ni mukammal o'zlashtirgan dasturchilar muam-moli masalalarni yechishda va yuzaga keladigan xatoliklarni bartaraf etishda mukammal muhit hisoblanadi.
T	To'siqlar (tashqi)	Ma'lumotlar xavfsizligining to'laqonli ta'minlanmaganligi

Xulosalash» (Rezyume, Veyer) metodi

Metodning maqsadi: Bu metod murakkab, ko'ptarmoqli, mumkin qadar, muammoli xarakteridagi mavzularni o'rganishga qaratilgan. Metodning mohiyati shundan iboratki, bunda mavzuning turli tarmoqlari bo'yicha bir xil axborot beriladi va ayni paytda, ularning har biri alohida aspektlarda muhokama etiladi. Masalan, muammo ijobiy va salbiy tomonlari, afzallik, fazilat va kamchiliklari, foyda va zararlari bo'yicha o'rganiladi. Bu interfaol metod tanqidiy, tahliliy, aniq mantiqiy fikrlashni muvaffaqiyatli rivojlantirishga hamda o'quvchilarning mustaqil g'oyalari, fikrlarini yozma va og'zaki shaklda tizimli bayon etish, himoya qilishga imkoniyat yaratadi. "Xulosalash" metodidan ma'ruza mashg'ulotlarida individual va juftliklardagi ish shaklida, amaliy va seminar mashg'ulotlarida kichik guruhlardagi ish shaklida mavzu yuzasidan bilimlarni mustahkamlash, tahlili qilish va taqqoslash maqsadida foydalanish mumkin.

Metodni amalga oshirish tartibi:



trener tinglovchilarni 5-6 kishidan iborat kichik guruhlariga ajratadi;



trening maqsadi, shartlari va tartibi bilan ishtirokchilarni tanishtirgach, har bir guruhga umumiy muammoni tahlil qilinishi zarur bo'lgan qismlari tushirilgan tarqatma materiallarni tarqatadi;



har bir guruh o'ziga berilgan muammoni atroflicha tahlil qilib, o'z mulohazalarini tavsiya etilayotgan sxema bo'yicha tarqatmaga yozma bayon qiladi;



navbatdagi bosqichda barcha guruhlar o'z taqdimotlarini o'tkazadilar. Shundan so'ng, trener tomonidan tahlillar umumlashtiriladi, zaruriy axborotlar bilan to'ldiriladi va mavzu yakunlanadi

Namuna:

Operatsion tizimlar					
MS DOS		NC		WINDOWS	
afzalligi	kamchiligi	afzalligi	kamchiligi	afzalligi	kamchiligi
Xulosa:					

“Keys-stadi” metodi

«**Keys-stadi**» - inglizcha soʻz boʻlib, («case» – aniq vaziyat, hodisa, «stadi» – oʻrganmoq, tahlil qilmoq) aniq vaziyatlarni oʻrganish, tahlil qilish asosida oʻqitishni amalga oshirishga qaratilgan metod hisoblanadi. Mazkur metod dastlab 1921 yil Garvard universitetida amaliy vaziyatlardan iqtisodiy boshqaruv fanlarini oʻrganishda foydalanish tartibida qoʻllanilgan. Keysda ochiq axborotlardan yoki aniq voqyea-hodisadan vaziyat sifatida tahlil uchun foydalanish mumkin. Keys harakatlari oʻz ichiga quyidagilarni qamrab oladi: Kim (Who), Qachon (When), Qayerda (Where), Nima uchun (Why), Qanday/ Qanaqa (How), Nima-natija (What).

“Keys metodi” ni amalga oshirish bosqichlari

Ish bosqichlari	Faoliyat shakli va mazmuni
1-bosqich: Keys va uning axborot taʼminoti bilan tanishtirish	✓ yakka tartibdagi audio-vizual ish; ✓ keys bilan tanishish(matnli, audio yoki media shaklda); ✓ axborotni umumlashtirish; ✓ axborot tahlili; ✓ muammolarni aniqlash
2-bosqich: Keysni aniqlashtirish va oʻquv topshirigʻni belgilash	✓ individual va guruhda ishlash; ✓ muammolarni dolzarblik iyerarxiasini aniqlash; ✓ asosiy muammoli vaziyatni belgilash
3-bosqich: Keysdagi asosiy muammoni tahlil etish orqali oʻquv topshirigʻining yechimini izlash, hal etish yoʻllarini ishlab chiqish	✓ individual va guruhda ishlash; ✓ muqobil yechim yoʻllarini ishlab chiqish; ✓ har bir yechimning imkoniyatlari va toʻsiqlarni tahlil qilish; ✓ muqobil yechimlarni tanlash
4-bosqich: Keys yechimini yechimini shakllantirish va asoslash, taqdimot.	✓ yakka va guruhda ishlash; ✓ muqobil variantlarni amalda qoʻllash imkoniyatlarini asoslash; ✓ ijodiy-loyiha taqdimotini tayyorlash; ✓ yakuniy xulosa va vaziyat yechimining amaliy aspektlarini yoritish

Keys. Berilgan topshiriq asosida dastur algoritmi tuzilib C++ dasturlash tilida dastur matni yozildi. Dasturni acm.tuit.uz saytiga yuborilganda “kompilyatsiyada hatolik” habari chiqdi. Yaʼni Sistema yechimni qabul qilmadi.

Keysni bajarish bosqichlari va topshiriqlar:

- Keysdgi muammoni keltirib chiqargan asosiy sabablarni belgilang (individual va kichik guruhda).
- Xatolikni bartaraf etuvchi ishlar ketma-ketligini belgilang (juftliklardagi ish).

«FSMU» metodi

Texnologiyaning maqsadi: Mazkur texnologiya ishtirokchilardagi umumiy fikrlardan xususiy xulosalar chiqarish, taqqoslash, qiyoslash orqali axborotni o'zlashtirish, xulosalash, shuningdek, mustaqil ijodiy fikrlash ko'nikmalarini shakllantirishga xizmat qiladi. Mazkur texnologiyadan ma'ruza mashg'ulotlarida, mustahkamlashda, o'tilgan mavzuni so'rashda, uyga vazifa berishda hamda amaliy mashg'ulot natijalarini tahlil etishda foydalanish tavsiya etiladi.

Texnologiyani amalga oshirish tartibi:

- qatnashchilarga mavzuga oid bo'lgan yakuniy xulosa yoki g'oya taklif etiladi;
- har bir ishtirokchiga FSMU texnologiyasining bosqichlari yozilgan qog'ozlarni tarqatiladi:

F	• Fikringizni bayon eting
S	• Fikringizni bayoniga sabab ko'rsating
M	• Ko'rsatgan sababingizni isbotlab misol keltiring
U	• Fikringizni umumlashtiring

- ishtirokchilarning munosabatlari individual yoki guruhiiy tartibda taqdimot qilinadi.

FSMU tahlili qatnashchilarda kasbiy-nazariy bilimlarni amaliy mashqlar va mavjud tajribalar asosida tezroq va muvaffaqiyatli o'zlashtirilishiga asos bo'ladi.

Namuna.

Fikr: Dasturiy ta'minot komputerning asosiy tarkibiy qismi va asosiy tamoyillaridan biridir".

Topshiriq: Mazkur fikrga nisbatan munosabatingizni FSMU orqali tahlil qiling.

“Assesment” metodi

Metodning maqsadi: mazkur metod ta’lim oluvchilarning bilim darajasini baholash, nazorat qilish, o’zlashtirish ko’rsatkichi va amaliy ko’nikmalarini tekshirishga yo’naltirilgan. Mazkur texnika orqali ta’lim oluvchilarning bilish faoliyati turli yo’nalishlar (test, amaliy ko’nikmalar, muammoli vaziyatlar mashqi, qiyosiy tahlil, simptomlarni aniqlash) bo’yicha tashhis qilinadi va baholanadi.

Metodni amalga oshirish tartibi:

“Assesment” lardan ma’ruza mashg’ulotlarida talabalarning yoki qatnashchilarning mavjud bilim darajasini o’rganishda, yangi ma’lumotlarni bayon qilishda, seminar, amaliy mashg’ulotlarda esa mavzu yoki ma’lumotlarni o’zlashtirish darajasini baholash, shuningdek, o’z-o’zini baholash maqsadida individual shaklda foydalanish tavsiya etiladi. Shuningdek, o’qituvchining ijodiy yondashuvi hamda o’quv maqsadlaridan kelib chiqib, assesmentga qo’shimcha topshiriqlarni kiritish mumkin.

Namuna. Har bir katakdagi to’g’ri javob 5 ball yoki 1-5 balgacha baholanishi mumkin.



Test

- $1 \cdot 10^{-8}$ aniqlikda natijani chop etish qanday bajariladi?
- A. `printf("%.8f",x)`
- B. `cout<<x`
- C. `cout<<("%.8f",x)`



Qiyosiy tahlil

- Dasturiy ta’minot mahsulotlardan foydalanish ko’rsatkichlarini tahlil qiling?



Tushuncha tahlili

- STD qisqartmasini izohlang.



Amaliy ko’nikma

- Windows muhitida ishlash uchun qo’shimcha kutubxona fayllarini o’rnating va sozlang.

“Insert” metodi

Metodning maqsadi: Mazkur metod o’quvchilarda yangi axborotlar tizimini qabul qilish va bilimlarni o’zlashtirilishini yengillashtirish maqsadida qo’llaniladi, shuningdek, bu metod o’quvchilar uchun xotira mashqi vazifasini ham o’taydi.

Metodni amalga oshirish tartibi:

- o’qituvchi mashg’ulotga qadar mavzuning asosiy tushunchalari mazmuni yoritilgan input-matnni tarqatma yoki taqdimot ko’rinishida tayyorlaydi;
- yangi mavzu mohiyatini yorituvchi matn ta’lim oluvchilarga tarqatiladi yoki taqdimot ko’rinishida namoyish etiladi;

- ta'lim oluvchilar individual tarzda matn bilan tanishib chiqib, o'z shaxsiy qarashlarini maxsus belgilar orqali ifodalaydilar. Matn bilan ishlashda talabalar yoki qatnashchilarga quyidagi maxsus belgilardan foydalanish tavsiya etiladi:

Belgilar	1-matn	2-matn	3-matn
“V” – tanish ma'lumot.			
“?” – mazkur ma'lumotni tushunmadim, izoh kerak.			
“+” bu ma'lumot men uchun yangilik.			
“–” bu fikr yoki mazkur ma'lumotga qarshiman?			

Belgilangan vaqt yakunlangach, ta'lim oluvchilar uchun notanish va tushunarsiz bo'lgan ma'lumotlar o'qituvchi tomonidan tahlil qilinib, izohlanadi, ularning mohiyati to'liq yoritiladi. Savollarga javob beriladi va mashg'ulot yakunlanadi.

“Tushunchalar tahlili” metodi

Metodning maqsadi: mazkur metod talabalar yoki qatnashchilarni mavzu buyicha tayanch tushunchalarni o'zlashtirish darajasini aniqlash, o'z bilimlarini mustaqil ravishda tekshirish, baholash, shuningdek, yangi mavzu buyicha dastlabki bilimlar darajasini tashhis qilish maqsadida qo'llaniladi.

Metodni amalga oshirish tartibi:

- ishtirokchilar mashg'ulot qoidalari bilan tanishtiriladi;
- o'quvchilarga mavzuga yoki bobga tegishli bo'lgan so'zlar, tushunchalar nomi tushirilgan tarqatmalar beriladi (individual yoki guruhli tartibda);
- o'quvchilar mazkur tushunchalar qanday ma'no anglatishi, qachon, qanday holatlarda qo'llanilishi haqida yozma ma'lumot beradilar;
- belgilangan vaqt yakuniga yetgach o'qituvchi berilgan tushunchalarning tugri va tuliq izohini uqib eshittiradi yoki slayd orqali namoyish etadi;
- har bir ishtirokchi berilgan tugri javoblar bilan uzining shaxsiy munosabatini taqqoslaydi, farqlarini aniqlaydi va o'z bilim darajasini tekshirib, baholaydi.

Namuna: “Moduldagi tayanch tushunchalar tahlili”

Tushunchalar	Sizningcha bu tushuncha qanday ma'noni anglatadi?	Qo'shimcha ma'lumot
Fayl	Makro deriktivalarni belgilash	a directive that defines a macro.

katalog	Bir source fayl ichida boshqa bir faylag murojatni amalga oshirish mexanizmi	a mechanism for textual inclusion of one source file into another.
tizim	add-and-assign operatori; masalan $a+=b$ vazifazi jihatdan $a=a+b$ bilan bir xil	add-and-assign operator; $a+=b$ is roughly equivalent to $a=a+b$.
open	Dastur jodini o'zida jamlovchi fayl	file containing definitions.
copy	Dastur jodini o'zida jamlovchi fayl	file containing definitions.
delete	Sarlavha fayli	header file
address	Hotira manzili	a memory location

Izoh: Ikkinchi ustunchaga qatnashchilar tomonidan fikr bildiriladi. Mazkur tushunchalar haqida qo'shimcha ma'lumot glossariyda keltirilgan.

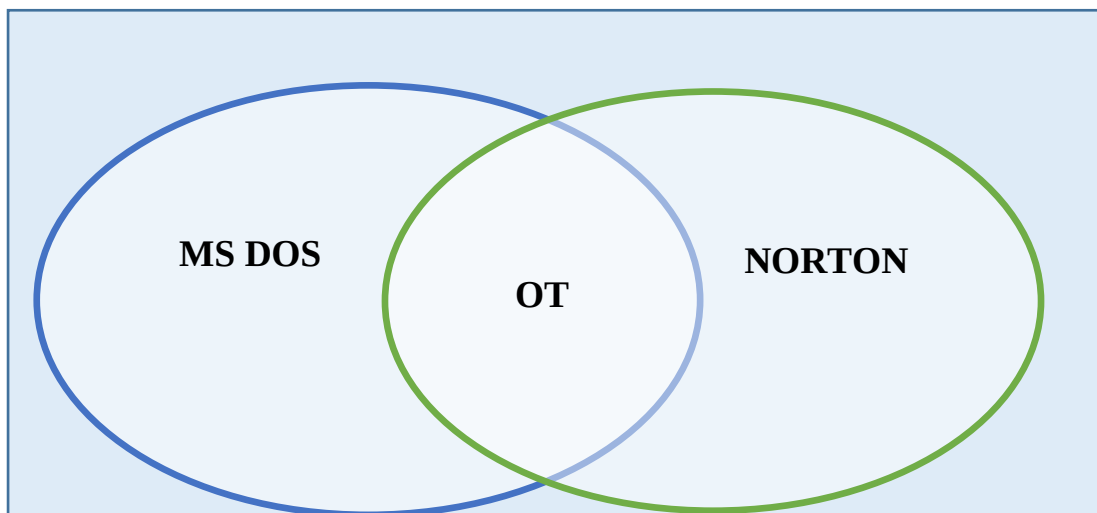
Venn Diagrammasi metodi

Metodning maqsadi: Bu metod grafik tasvir orqali o'qitishni tashkil etish shakli bo'lib, u ikkita o'zaro kesishgan aylana tasviri orqali ifodalanadi. Mazkur metod turli tushunchalar, asoslar, tasavurlarning analiz va sintezini ikki aspekt orqali ko'rib chiqish, ularning umumiy va farqlovchi jihatlarini aniqlash, taqqoslash imkonini beradi.

Metodni amalga oshirish tartibi:

- ishtirokchilar ikki kishidan iborat juftliklarga birlashtiriladilar va ularga ko'rib chiqilayotgan tushuncha yoki asosning o'ziga xos, farqli jihatlarini (yoki aksi) doiralari ichiga yozib chiqish taklif etiladi;
- navbatdagi bosqichda ishtirokchilar to'rt kishidan iborat kichik guruhlariga birlashtiriladi va har bir juftlik o'z tahlili bilan guruh a'zolarini tanishtiradilar;
- juftliklarning tahlili eshitilgach, ular birgalashib, ko'rib chiqilayotgan muammo yohud tushunchalarning umumiy jihatlarini (yoki farqli) izlab topadilar, umumlashtiradilar va doirachalarning kesishgan qismiga yozadilar.

Namuna: Dasturlash tillari turlari bo'yicha



“Blis-o’yin” metodi

Metodning maqsadi: o’quvchilarda tezlik, axborotlar tizmini tahlil qilish, rejalashtirish, prognozlash ko’nikmalarini shakllantirishdan iborat. Mazkur metodni baholash va mustahkamlash maksadida qo’llash samarali natijalarni beradi.

Metodni amalga oshirish bosqichlari:

1. Dastlab ishtirokchilarga belgilangan mavzu yuzasidan tayyorlangan topshiriq, ya’ni tarqatma materiallarni alohida-alohida beriladi va ulardan materialni sinchiklab o’rganish talab etiladi. Shundan so’ng, ishtirokchilarga to’g’ri javoblar tarqatmadagi «yakka baho» kolonkasiga belgilash kerakligi tushuntiriladi. Bu bosqichda vazifa yakka tartibda bajariladi.

2. Navbatdagi bosqichda trener-o’qituvchi ishtirokchilarga uch kishidan iborat kichik guruhlariga birlashtiradi va guruh a’zolarini o’z fikrlari bilan guruhdoshlarini tanishtirib, bahslashib, bir-biriga ta’sir o’tkazib, o’z fikrlariga ishonirish, kelishgan holda bir to’xtamga kelib, javoblarini «guruh bahosi» bo’limiga raqamlar bilan belgilab chiqishni topshiradi. Bu vazifa uchun 15 daqiqa vaqt beriladi.

3. Barcha kichik guruhlar o’z ishlarini tugatgach, to’g’ri harakatlar ketma-ketligi trener-o’qituvchi tomonidan o’qib eshittiriladi, va o’quvchilardan bu javoblarni «to’g’ri javob» bo’limiga yozish so’raladi.

4. «To’g’ri javob» bo’limida berilgan raqamlardan «yakka baho» bo’limida berilgan raqamlar taqqoslanib, farq bulsa «0», mos kelsa «1» ball quyish so’raladi. Shundan so’ng «yakka xato» bo’limidagi farqlar yuqoridan pastga qarab qo’shib chiqilib, umumiy yig’indi hisoblanadi.

5. Xuddi shu tartibda «to’g’ri javob» va «guruh bahosi» o’rtasidagi farq chiqariladi va ballar «guruh xatosi» bo’limiga yozib, yuqoridan pastga qarab qo’shiladi va umumiy yig’indi keltirib chiqariladi.

6. Trener-o’qituvchi yakka va guruh xatolarini to’plangan umumiy yig’indi bo’yicha alohida-alohida sharhlab beradi.

7. Ishtirokchilarga olgan baholariga qarab, ularning mavzu bo’yicha o’zlashtirish darajalari aniqlanadi.

«OT ni o'rnatish va sozlash» ketma-ketligini joylashtiring. O'zingizni tekshirib ko'ring!

Harakatlar mazmuni	Yakka baho	Yakka xato	To'g'ri javob	Guruh bahosi	Guruh xatosi
#include <iostream>			1		
using namespace std;			2		
int main() {			3		
Cout<< "Hello, World!" <<endl;			4		
return 0;			5		
}			6		

“Brifing” metodi

“Brifing”- (ing. briefing-qisqa) biror-bir masala yoki savolning muhokamasiga bag'ishlangan qisqa press-konferensiya.

O'tkazish bosqichlari:

Taqdimot qismi.

Muhokama jarayoni (savol-javoblar asosida).

Brifinglardan trening yakunlarini tahlil qilishda foydalanish mumkin. Shuningdek, amaliy o'yinlarning bir shakli sifatida qatnashchilar bilan birga dolzarb mavzu yoki muammo muhokamasiga bag'ishlangan brifinglar tashkil etish mumkin bo'ladi. Tinglovchilar yoki tinglovchilar tomonidan yaratilgan mobil ilovalarning taqdimotini o'tkazishda ham foydalanish mumkin. **“Portfolio” metodi** “Portfolio” – (ital. portfolio-portfel, ingl.hujjatlar uchun papka) ta'limiy va kasbiy faoliyat natijalarini autentik baholashga xizmat qiluvchi zamonaviy ta'lim texnologiyalaridan hisoblanadi. Portfolio mutaxassisning saralangan o'quv-metodik ishlari, kasbiy yutuqlari yig'indisi sifatida aks etadi. Jumladan, talaba yoki tinglovchilarning modul yuzasidan o'zlashtirish natijasini elektron portfoliolar orqali tekshirish mumkin bo'ladi. Oliy ta'lim muassasalarida portfolioning quyidagi turlari mavjud:

Faoliyat turi	Ish shakli	
	Individual	Guruhiy
Ta'limiy faoliyat	Tinglovchilar portfoliosi, bitiruvchi, doktorant, tinglovchi portfoliosi va boshq.	Tinglovchilar guruhi, tinglovchilar guruhi portfoliosi va boshq.
Pedagogik faoliyat	O'qituvchi portfoliosi, rahbar xodim portfoliosi	Kafedra, fakultet, markaz, OTM portfoliosi va boshq.

1- MA'RUZA MASHG'ULOTI:

**Algoritmlar haqida umumiy tushunchalar. Algoritm tushunchasi, asosiy xossalari.
Algoritm ijrochisi. Algoritmlarni tasvirlash usullari. Algoritm turlari.**

(ma'ruza – 2 soat)

1.1. Ma'ruza mavzusi bo'yicha reja-topshiriq va o'quv uslubiy xujjatlar.

Talabalar soni 50 ta	O'quv soati - 2 soat
Mashg'ulot shakli	Vizuallashtirilgan ma'ruza
Ma'ruza rejasi	1. Fanga kirish. 2. Algoritm haqida umumiy intuitiv ta'rif ma'nosidagi tushuncha. 3. Algoritmning kibernetik ta'rifi. Yevklid algoritmi. 4. Algoritmning asosiy hossalari, ijrochilari, tasvirlash usullari, turlari. Algoritmning asosiy tiplari.
<i>O'quv mashg'ulotining maqsadi</i>	O'quv kursi haqida umumiy tasavvurni berish.
<i>Pedagogik vazifalar:</i>	<i>O'quv faoliyati natijalari:</i>
Fanning ahamiyati va vazifalari, uni o'quv fanlar tizimida tutgan o'rni bilan tanishtirish;	Fanning ahamiyati va vazifalarini aytib bera oladilar;
Fanning tuzilishini va tavsiya etiladigan o'quv adabiyotlarini sharxlash;	Fanning tuzilishi va tavsiya etilayotgan o'quv adabiyotlari bilan tanishadilar;
Algoritm haqida umumiy tushuncha berish, algoritmning kibernetik ta'rifini tushuntirish, Yevklid algoritmi haqida tushuncha beriladi;	Algoritm haqida tushunchaga ega bo'ladilar, algoritmning kibernetik ta'rifi, Yevklid algoritmi haqida tasavvurga ega bo'ladilar;
haqida tushuncha beriladi;	haqida bahs yurita oladilar;
Algoritmning asosiy hossalari, ijrochilari, tasvirlash usullari, turlari va uning asosiy tiplari bilan tanishtiriladi;	Algoritmning asosiy hossalari, ijrochilari, tasvirlash usullari, turlari va uning asosiy tiplari haqida tushunchaga ega buladilar. Ularni farqlab bera oladilar;
<i>O'qitish usullari-texnikasi</i>	<i>Ma'ruza, namoyish, blis-so'rov, aqliy hujum</i>
<i>o'qitish vositalari</i>	<i>ma'ruza matni, kompyuter texnologiyasi, doska</i>
<i>o'qitish shakllari</i>	frontal, kollektiv ish
<i>o'qitish sharoiti</i>	<i>kompyuter, multimedia proektor (yoki proektor va LCD panel)</i>
<i>Monitoring va baholash</i>	<i>Og'zaki so'rov, tezkor so'rov. Nazorat so'rovlari. Boshlang'ich nazorat test savollari.</i>

Ma'ruza mashg'ulotining texnologik xaritasi.

Faoliyat bosqichlari	Faoliyatning mazmuni	
	O'qituvchi	Talaba
1-bosqich. Kursga va mavzuga kirish (20 min)	1.1. O'quv kursining nomi, kurs doirasida dastlabki umumiy tasavvur, hamda uslubiy va tashkiliy tomonlari, tinglovchilar bilimlarini baholash mezonlari bilan tanishtiradi (1.1-Ilova). O'quv kursining 1 – mavzusi bo'yicha dastlabki umumiy tasavvurni beradi va hamda uslubiy va tashkiliy tomonlari bilan tanishtiradi.	Tinglaydilar
	1.2. Mavzu rejasi va tayanch iboralar bilan tanishtiriladi. (1.2-va 1.3-ilovalar)	Yozadilar, tinglaydilar
	1.3. Blis-so'rov usulida mavzu bo'yicha ma'lum bo'lgan tushunchalarni sanab berishni so'raydi (1.4 -ilova).	Tushunchalarni sanab beradilar
2-bosqich. Asosiy bo'lim (50 min)	2.1. Ma'ruza rejasining 1-4 punktlari bo'yicha tushuntiradi, har bir punkt nihoyasida umumlashtirib boradi. Jarayon kompyuter slaydlarini namoyish qilish bilan olib boriladi. (1.6-ilova)	Tinglaydilar. konspekt qilib boradilar.
	2.2. Tayanch iboralarga qaytiladi. Ekranda nazorat uchun savollar va topshiriqlar namoyish etiladi. Savollarga javob beriladi va B/B/B jadvali va "Venn" diagrammasi to'ldiriladi (1.5-Ilova).	Har bir tayanch tushuncha va iboralarni muhokama qiladilar. Barcha axborotni tizimlashtiradilar. Konspekt qiladilar.
3-bosqich. Yakunlovchi (10 min)	3.1. Mavzu bo'yicha yakunlovchi xulosalar qiladi. Mavzuda olingan bilimlarni qayerda ishlatish mumkinligi ma'lum qiladi.	Savollar beradilar
	3.2. Mavzu maqsadiga erishishdagi tinglovchilar faoliyati tahlil qilinadi va baholanadi.	
	3.3. Mavzu bo'yicha mustaqil o'rganish uchun topshiriqlar beradi.	Mustaqil o'rganish uchun topshiriqlarni yozib oladilar

1.1-ilova

“Algoritmash va dasturlash tillari” fanidan 5 ballik sistema asosida quyidagi jadvalga ko'ra reyting nazorati amalga oshiriladi.



Nazorat turi	Nazorat shakllari	Har bir nazorat uchun belgilangan maksimal ball	Nazorat soni	Nazorat shakllari bo'yicha belgilangan maksimal ball
Joriy nazorat	1. Amaliy ish	5	2	10
	2. Lab.ishi	5	2	10
	3. Amaliy ish	5	2	10
	4. Lab.ishi	5	2	10
Jami:		50	8	40
Oraliq nazorat	1. Test	15	1	15
	2. Yozma ish	15	1	15
Jami:		20	2	30
Yakuniy nazorat	1.Yozma ish.	30	1	30
Jami:		30	11	100

1.2-ilova

Ma'ruza rejasi

1. Fanga kirish.
2. Algoritm haqida umumiy intuitiv ta'rif ma'nosidagi tushuncha.
3. Algoritmning kibernetik ta'rifi. Yevklid algoritmi.
4. Algoritmning asosiy hossalari, ijrochilari, tasvirlash usullari, turlari. Algoritmning

1.3-ilova

Tayanch iboralar

Algoritm, Ob'ekt, Tasvir, qadam, Alfavit, So'z. Algoritmik til. Algoritm hossalari.Chiziqli algoritm. Tarmoqlanuvchi algoritm. Takrorlanuvchi algoritm. Yevklid algoritmi.

1. Algoritm deyilganda nima tushuniladi? Algoritm so'zi qayerdan kelib chiqadi?
2. Algoritmning qanday xossalari bilasiz? Algoritmni yozishning qanday
3. usullari bilasiz?
4. Blok-sxema nima? Blok-sxemaning asosiy elementlarini ayting.
5. Algoritmning turlarini sanab o'ring.
6. Masalalarni EHM da yechish bosqichlarini sanab o'ring.
7. Al-Xorazmiyning xæti va ijodi xaqida nimalarni bilasiz?
8. Masalani echish algoritmini qanday usullar bilan ifodalash mumkin?
9. Algoritmga qo'yiladigan talablarni sanab o'ring.
10. Algoritmning qanday xossalari bilasiz?
11. Masala echimi algoritmini tuzishda qanday qoidalarga amal qilish lozim?
12. CHiziqli algoritm deb nimaga aytiladi?
13. Tarmoqlanuvchi algoritmni ta'riflang va unga xætiy masalalar keltiring.
14. Takrorlanuvchi algoritmning vazifasini tushuntiring.
15. Algoritmni ifodalashda blok-sxemalardan foydalanishning afzalliklarini aytib bering.
16. Blok-sxema elementlarini chizib ko'rsating va ularning vazifalarini ayting.

1. B/B/B jadvali

1.5-ilova

Bilaman	Bilmoqchiman	Bilib oldim

2. Uchta turdagi algoritmga mos «VENN» diagrammasini yaratish

1.6-ilova

Ma'ruzaning mazmuni bo'yicha ko'rgazmali slaydlari

“Algoritm” so'zi va tushunchasi IX asrda yashab ijod etgan buyuk bobolonimiz Muxammad Al – Xorazmiy nomi bilan uzviy bog'liq. Algoritm so'zi Al – Xorazmiyning arifmetikaga bag'ishlangan asarining dastlabki etidagi «Dixit Algoritmi» («Dedik Alxorazmiy» ning lotincha ifodasi) degan jumalardan kelib chiqqan desa bo'ladi. Al – Xorazmiy birinchi bo'lib o'nlik sanoq sistemasining printsiplarini va unda turli amallar bajarish qoidalarini asoslab berdi. Bu esa hisoblash ishlarini ixchamlash va osonlashtirish imkonini yaratdi, chunki bu bilan o'sha davrda qo'llangan rasm ko'rinishida bo'lgan ma'lumotlarni, rim raqamlari va sonlarni so'z orqali yozib bajarishdagi noqulayliklar bartaraf etildi.	Dastlab algoritm deyilganda o'nlik sanoq sistemasidagi sonlar ustida turli arifmetik amallar bajarish qoidalarini tushunib kelingan. Demak, Al – Xorazmiyning ilmiy asarlaridan kelib chiqqan algoritm tushunchasi juda keng ma'noda qo'llaniladi. Bular bilan, aziz talaba, siz – biz g'ururlansak bo'ladi. Ayni bir vaqtda kundalik hayotimizda bizni o'rab olgan algoritmshunchalik ko'p va xilma – xil bo'lsada, biz ularning turlarini ajrata olishimiz, xususiyatlari hamda tasvirlash va ishlash usullarini bilishimiz lozim. Algoritm nima? Algoritm – biz o'rganayotgan kursda asos etib olingan tushunchalardan biri hisoblanadi, chunki uning ilmiy ta'rifi yo'q. Biroq biz algoritmning mohiyatini aniq va qat'iyroq tushuntirishga xarakat qilamiz. Algoritm deganda biror maqsadga erishishga yoki ma'lum masalani echishga qaratilgan ko'rsatmalar (buyruqlarning) aniq, tushunarli, chekli hamda to'liq tizimi tushuniladi. Ko'rinib turibdiki, bunday ko'rsatmalar tizimi turmushimizda juda ko'p uchraydi. Algoritmga misol sifatida taomlar tayyorlash retseptlarini, turli avtomatik quрилmalarni (videomagnitofon, kir yuvish mashinasi, mexanik yoki elektron o'yinchoqlarni) ishlatish bo'yicha yo'riqnomalarni, sport musobaqalari yoki ko'cha harakati qoidalarini keltirishimiz mumkin.
--	---

Evklid algoritmi

Qadimgi Gresiyalik matematik Evklid 2 ta natural A va V sonlarning eng katta umumiy bo'luvchisini topish algoritmini taklif etdi. Uning ma'nosi quyidagicha:

Katta sondan kichigini ayirish, natijani katta son o'rniga kuyish va ikkala son tenglashguncha bu amalni takrorlash. Ushbu teng sonlar izlangan natijadir.

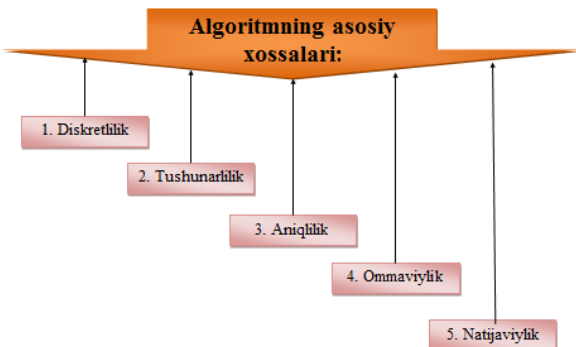
Evklid algoritmidan A va V sonlarning eng katta umumiy bo'luvchisi ushbu sonlar ayirmasining eng katta bo'luvchisi hamda ikkala A, V sonlarning ham umumiy eng katta bo'luvchisi bo'lish faktidan foydalanilgan.

Evklid algoritmining bu ifodasiga aniqlik etishmaydi, shuning uchun uning konkretlashtirish zarur bo'ladi.

Xaqiqiy Evklid algoritmi quyidagicha:

- ✓ A sonni birinchi son deb, V sonni ikkinchi son deb qaralsin. 2-punktga o'tilsin;
- ✓ Birinchi va ikkinchi sonlarni taqqoslang. Agar ular teng bo'lsa, 5-punktga o'tilsin, aks xolda 3-punktga o'tilsin;
- ✓ Agar birinchi son ikkinchi sondan kichik bulsa, ularning o'rnini almashtirilsin. 4-punktga o'tilsin;
- ✓ Birinchi sondan ikkinchi son ayirilsin va ayirma birinchi son deb xisoblansin. 2-punktga o'tilsin;
- ✓ Birinchi sonni natija sifatida qabul qilinsin. Tamom.

Algoritmnining asosiy xossalari:



Algoritmnini tasvirlash usullari.

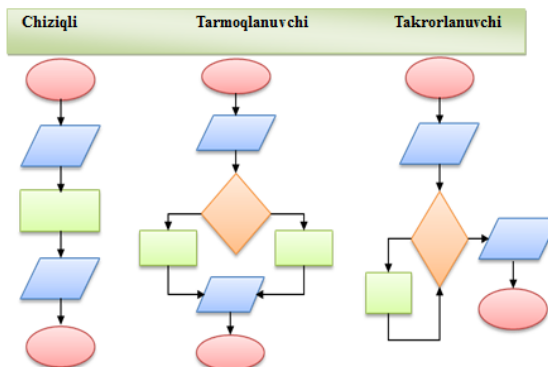
1. Algoritmnining so'zlar orqali ifodalanishi. Ushbu holda ijrochi uchun beriladigan har bir ko'rsatma jumalalar orqali buyruq mazmunida beriladi.
2. Algoritmnining formulalar yordamida berilishi. Algoritmnini formulalar bilan berilish usulidan matematika, fizika, kimyo kabi aniq fanlarni o'rganishda ko'plab foydalanamiz. Bu usulni ba'zan analitik ifodalash deb ham yuritishimiz mumkin.
3. Algoritmnining jadval ko'rinishida berilishi. Algoritmnining bu tarzda tasvirlanishidan ham ko'p foydalanamiz. Masalan, maktabda qo'llanilib kelinayotgan V. M. Bradisning to'rt xonali matematik jadvallari, turli lotoreya yutuq jadvallari, musobaqalar jadvallari. Funktsiyalarning grafiklarini chizishda ham algoritmlarning qiymatlar jadvali ko'rinishlaridan foydalanamiz. Bu kabi jadvallardan foydalanish algoritmlari sodda bo'lgani tufayli ularni o'zlashtirib olish osondir.
4. Algoritmlarning grafik shaklida tasvirlanishi. Algoritmnining bu shakli sizga avvaldan tanish, chunki matematika kursida chizilgan grafiklarning ko'pchiligi algoritmnining grafik usulda berilishiga misol bo'ladi. Bundan tashqari shahar yoki turar joy mavzolari joylashgan biror uy hamda inshootlarini izlash va xarakatlanish bo'yicha berilgan xarita – sxemalar ham shunga misol bo'la oladi. Biz algoritmnining informatikada ko'p qo'llaniluvchi yana bir grafik tasviri bilan tanishtirib o'tmoqchimiz. Bu maxsus vositaning nomi blok – sxema deb yuritiladi. Blok – sxemalar turli sodda geometrik figuralar shaklidagi oddiy elementlardan tashkil topadi.

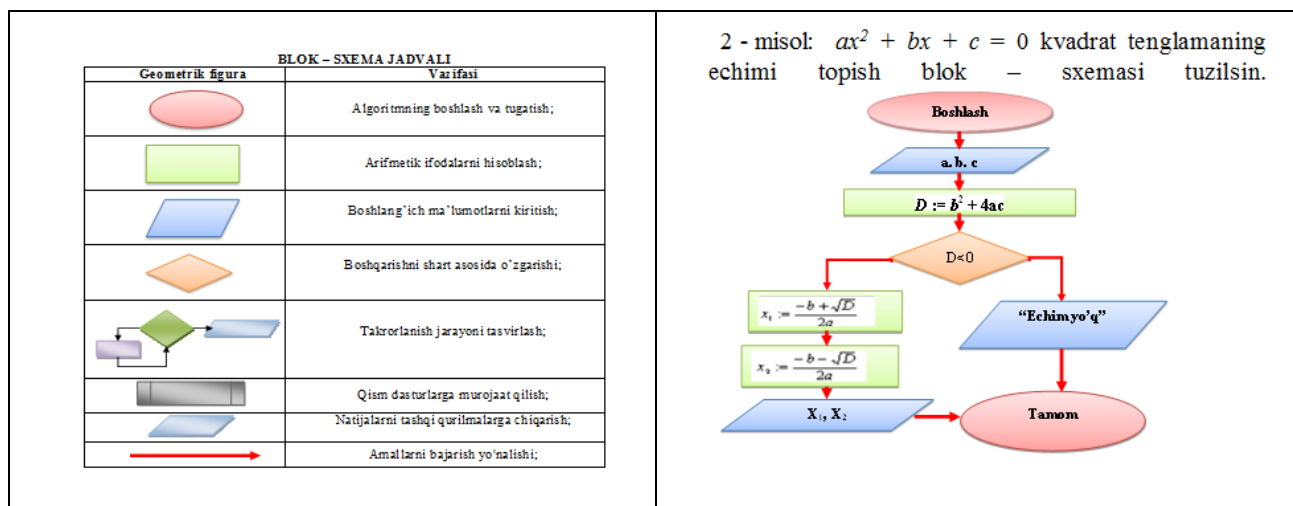
Algoritmnining quyidagi uch turi mavjud:

Hech qanday shart tekshirilmaydigan va tartib bilan faqat ketma – ket bajariladigan algoritmlar **chiziqli algoritmlar** deb yuritiladi.

Shartga muvofiq bajariladigan ko'rsatmalar bilan tuziladigan algoritmlar **tarmoqlanuvchi algoritmlar** deyiladi.

Masalani echish jarayonida bir hil amallarni takrorlanishini talab qiluvchi algoritmlar **takrorlanuvchi algoritmlar** deyiladi.





Asosiy darslik va o'quv qo'llanmalar

1. Abduqodirov A., Xaitov A., Shodiev R. Axborot texnologiyalari – T.: O'zbekiston, 2002 y.
2. Aripov M. va boshqalar Informatika va informasion texnologiyalar. Oliy o'quv yurti talabalari uchun darslik T. 2005 y.
3. Информатика. Проф. Н. В. Макарова.-Тошкент; 2005.
4. Sh.Nazirov. Delphi tilida dastrurlash asoslari T.2008 y.
5. В.М.Пестиков, А.Н.Маслобоев. Turbo PASCAL 7.0. Изучаем на примерах. – С-Пт.: "БХВ-Петербург", 2004.
6. П.Дарахвелидзе, Е.Марков. Программирование в Delphi7. Учебник.-СПБ-2003.

Qo'shimcha adabiyotlar

7. M. Ashurov, I. Mirmaxmudov, Sh. Sapaev Zamonaviy dasturlash tillari fanidan laboratoriya ishlari Toshkent 2008
8. В.Т.Безручко.практикум по курсу информатики. «Финанси и статистика» М-2004.

2-Ma'ruza: Algoritmik tillar va algoritmlarni ishlab chiqish

Reja:

1. Algoritmik tillar
2. Algoritmni ishlab chiqish
3. Algoritmning murakkabligi

1. Algoritmik tillar

Algoritmlarni yozish uchun qo'llaniladigan tillar **algoritmik tillar** deb ataladi. Algoritmik tilni EHM ham tushunsa, u holda bu til **dasturlash tili** deb ataladi. Demak, algoritmik til yoki dasturlash tili ham berilgan misol yoki masalani yechish algoritmining yozish usullaridan biri ekan. Shu yozish usullarining qanchalik mashinaga bog'liqligi yoki tabiiy tillarga yaqinligiga qarab ular turlicha nomlanadi.

Ma'lumki, har bir EHM o'zining buyruqlar (komandalar) sistemasi va mashina tiliga ega. Bu til ma'lum qonun-qoidalar asosida yozilgan 0 va 1 raqamlari ketma-ketligidan iboratdir. Bu qonun-qoidalar asosan, ma'lum bir EHM qurilmalarining tuzilishiga bog'liq bo'lib, aynan shu quruq mashinalari uchun o'rinnidir. Har bir EHM o'zining tilida yozilgan dasturnigina tushuna oladi va uni ijro etadi.

2. Algoritmni ishlab chiqish

Algoritmni yaratish ijobiy ish, shuning uchun ixtiyoriy zarur algoritmni tuzish imkonini beradigan bir umumiy usul mavjud emas. Lekin algoritmni ishlab chiqishni asoslangan oddiy sxemalarini beradigan ko'pgina algoritmlashtirish nazariyalari bor. Bunday sxemalar va yangi algoritmni paydo qilishning o'rtasida qattiq bog'liqlik kuzatiladi. Tez uchraydigan va ko'p foydalaniladigan usullarni quyidagicha ajratib olish mumkin:

1. Algoritmni konstruksiyalash. Bu usulda yangi algoritm mavjud algoritmardan tarkibiy qismlar sifatida foydalanib, bir-biriga moslab bir butunlik hosil qilish yo'li bilan ishlab chiqiladi.

2. Algoritmni ekvivalent qayta ishlash. Ikki algoritm ekvivalent hisoblanishi uchun quyidagi shartlar bajarilish kerak:

- Bittasi uchun mumkin bo'lgan dastlabki berilganlar varianti, ikkinchisi uchun ham mumkin bo'lishi kerak.

- Bir algoritmni qandaydir dastlabki ma'lumotga qo'llanilishi, ikkinchi algoritmni ham shu berilganga qo'llanilishiga kafolat beradi.

- Bir xil dastlabki berilgan ma'lumot uchun ikkala algoritm ham bir xil natija berishi. Lekin bu algoritmni ikki xil shakllarini ekvivalent deb nomlash noto'g'ridir.

Shunday qilib, algoritmni ekvivalent qayta ishlash deb, natijada dastlabki algoritmga ekvivalent algoritmni paydo qiladigan o'zgartirilishlarga aytiladi.

Misol tariqasida, algoritmni bir tildan boshqa tilga o'tkazishni keltirish mumkin. Shu bilan birgalikda algoritmni ekvivalent qayta ishlash usuli bilan keskin o'zgartirish mumkin, lekin bu holda asosiy e'tiborni dastlabki algoritmga nisbatan yahshi algoritmni yaratishga berish kerak.

3. Toraytiruvchi o'zgartirishlar. Bunday o'zgartirishlar natijasida dastlabki algoritm yechish kerak bo'lgan masalalarning xususiy holati yechimi algoritmni ishlab chiqiladi. Odatda, bu usulda ekvivalent qayta ishlash jarayonida algoritmni ixchamlashtirish maqsadda foydalaniladi.

4. Formal usulni matematikaga bog'liq bo'lmagan muammoga qo'llash. Bu yerda matematik muammo matematik ko'rinishga o'tkazilib, uning algoritmini ishlab chiqishga urinishi. Agar o'xshash matematik masala yechimining algoritmi ma'lum bo'lsa, undan foydalaniladi.

Algoritm lashtirish jarayoni uslublari bo'yicha matematik modellarni tuzish jarayoniga juda yaqin. Har bir algoritmni ishlab chiqish bevosita o'ziga xos yondashishni talab qilishiga qaramasdan, bu faoliyatni umumiy uslub va bosqichlari ham mavjud. Ba'zan dasturlarni tezroq yozib boshlashga hohish paydo bo'ladi. Lekin bu xatoli, chunki aynan algoritmni ishlab chiqish bosqichiga va uning to'g'riligiga masalaning to'liq yechimi bog'liqdir. Algoritmni tuzish turli xil uslublari mavjud.

Algoritmni to'g'riligini tekshirish. Dastur to'g'riligini isbotlashning eng keng tarqalgan turi – bu uni testlardan o'tkazishdir. Algoritmni tekshirishda nazoratchi boshlang'ich ma'lumotlarni majmui algoritmik test deb nomlanadi. To'g'ri deb shunday algoritmga aytiladiki, u masalaning qo'yilishida talab qilinadigan natijani har qanday ruxsat etilgan boshlang'ich ma'lumotlar bilan ham

shakllantirib biladi. Odatda, dastur bergan natijalar ma'lum bo'lgan yoki qo'lda hisoblangan ma'lumotlar bilan taqqoslanadi va ular to'g'riligi aniqlansa dastur to'g'ri ishlaydi degan hulosaga kelish mumkin. Ammo bu usul bilan foydalanuvchini hamma shubhalardan xalos qilib bo'lmaydi, ya'ni dastur ishlamaydigan hamma holatlarni hisobga olib bo'lmaydi.

Gudman va Xidetniyemilar tomonidan algoritm to'g'riligini isbotlash uchun quyidagi uslubiyat taklif qilingan.

Algoritm 0 dan m gacha bo'lgan qadamlar ketma-ketligi ko'rinishida tavsiflangan deb tahmin qilaylik. Har bir qadam uchun qandaydir asoslanishni taklif etamiz. Xususan, qadamdan oldin va keyin ishlaydigan shartlar haqida lemma kerak bo'lishi mumkin. Shu bilan birgalikda, algoritm chekliligining isbotini ham taklif etamiz va hamma ruxsat etilgan kiritish ma'lumotlarini tekshirib, hamma mumkin bo'lgan chiqarish ma'lumotlarni olamiz. Algoritmni to'g'riligi bilan samaradorligi o'rtasida hech qanday aloqa yo'qligini ta'kidlab o'tamiz. Aslida hamma talablarga bir xil yahshi javob beradigan algoritm kamdan-kam ishlab chiqiladi.

Algoritmni amalga oshirish

Algoritmni amalga oshirish deganda, EHM uchun dasturni yozish deb tushuniladi. Buning uchun quyidagi savollarga javob berish kerak:

1. Asosiy o'zgaruvchilarni aniqlash.
2. O'zgaruvchilarning turlarini aniqlash.
3. Nechta massiv yoki fayllar va qanday kattalikda ular kerak bo'ladi?
4. Bog'lanilgan ro'yhatlardan foydalanish ma'nolimi?
5. Qanday dasturiy qismlar kerak bo'lishi mumkin (tayyor bo'lsa ham)?
6. Qaysi dasturlash tilini tanlash?

Dastur yozish yoki tuzishning hilma-hil usillari va uslublari mavjud.

3. Algoritmni murakkabligi

Algoritmni tahlil qilishdan maqsad – algoritmga ma'lumotlarni aniq muvaffaqiyatli qayta ishlash uchun kerak bo'ladigan xotira hajmi va ishlash vaqtining baholari va chegaralarini olish. Bir masalani yechadigan ikki algoritmni taqqoslash uchun qandaydir sonli mezon topish kerak.

Faraz qilaylik, A – qandaydir bir turkumdagi masalalarni yechadigan algoritm, n – esa shu turkumdagi alohida bir masalaning kattaligi. Umumiy holda, n – oddiy skalyar yoki massiv yoki kiritiladigan ketma – ketlikning uzunligi bo'lishi mumkin. $f_A(n)$ - n kattalikdagi ixtiyoriy masalani yechadigan algoritm A bajarish kerak bo'lgan asosiy amallarni (qo'shish, ayirish, taqqoslash,...) yuqori chegarasini beradigan ishchi funksiya. Algoritmning sifatini baholash uchun quyidagi mezonni ishlatamiz.

Agar $f_A(n)$ o'sish tartibi n dan bog'liq bo'lgan polinomdan katta bo'lmasa, A algoritm **polinomial** deb aytiladi, aks holda algoritm A eksponensial hisoblanadi.

Shular bilan birgalikda tahlil jarayonida ko'p matematik fanlarda standart bo'lgan iboralar ishlatiladi.

$f_A(n)$ funksiya $O[g(n)]$ deb belgilanadi, va $\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = \text{const} \neq 0$ bo'lganda, uni tartibi katta n lar uchun $g(n)$ deb qabul qilinadi. Demak $f(n) = O[g(n)]$.

$f_A(n)$ funksiyasi $o[z(n)]$ deb katta n lar uchun belgilanadi, va unda $\lim_{n \rightarrow \infty} \frac{h(n)}{z(n)} = 0$

sharti bajariladi.

Bu begilar “katta O” va “kichik o” deb nomlanadi. Agar $f(n)=O[g(n)]$ bo’lsa, ikkala funksiya ham $n \rightarrow \infty$ bo’lganda bir xil tezlikda o’sadi.

Agar $f(n)=O[g(n)]$ bo’lsa, unda $g(n)$, $f(n)$ nisbatan ancha tez o’sadi.

Demak, $P_k(n)$ - qandaydir n o’zgaruvchidan bog’liq va k darajadagi polinom uchun $f_A(n) = O[P_k(n)]$ yoki $f_A(n) = oP_k(n)$ bo’lganda algoritm polynomial hisoblanadi, aks holda algoritm eksponensial.

Eksponensial algoritm yahshi ishlamaydigan deb hisoblanadi. Agar algoritmlar eksponensial bo’lsa, ular orasida eng samaralisini topish kerak, n kattalikdagi masalani $O(2^n)$ qadamda yechadigan algoritm $O(n!)$ yoki $O(n^n)$ qadamda masalani yechadigan algoritmdan afzalroq.

Dasturni tekshirish

Biz dasturni har bir qismini tekshiradigan kirituvchi ma’lumotlar to’plamini tanlashimiz kerak. Ko’p murakkab algoritmlarni matematik tomondan tadqiq qilish yoki juda qiyin yoki mumkin emas. Bunday holatlarda algoritmni faoliyat jarayonida va qiyinligi bo’yicha tekshiradi. Bundan tashqari dasturlarni hisoblash imkoniyatlarini aniqlash uchun ham testlash maqsadga muvofiq. Ko’p dasturlar qandaydir kiritiladigan ma’lumotlar bilan yahshi ishlasa, boshqalari bilan yomon ishlaydi. “Yahshi” lardan “yomon” larga o’tish “mayin” bo’lish kerak. Testlash uchun ma’lumotlar dasturning qiyinligiga, mavjud vaqt resurslariga, kiritish-chiqarish soniga bog’liq holda tanlanadi. Bu yerda analitik va eksperimental tahlil bir-birini to’ldiradi.

Hujjatlashtirish

O’zingiz yozmagan dastur kodini o’qish juda qiyin. Bu muammoni hujjatlashtirish yordamida yechsa bo’ladi. Hujjatlashtirish o’z ichiga hamma yordamchi ma’lumotlarni oladi va dasturda nima bajarilishini tushuntirib beradi, xususan, blok-sxemalardagi boshqarishni uzatish, berilganlarni kiritish-chiqarish shaklini batafsil tavsif qilish, siklning parametrlari, yordamchi lokal va global proseduralarni bajarilishi va boshqalar.

Hujjatlashtirishning eng asosiy qoidasi bu “boshqalar yozgan dasturlarni qanday ko’rishni istasangiz, o’zingiz ham dasturni shunday ko’rinishda rasmiylashtiring”.

Masalalar yechish.

1 - misol. Berilgan to’rt xonali butun sonning raqamlari ko’paytmasini toping.

Test tartibi	Tekshirish	Son	Natija
1	Musbat son	2314	$P = 24$
2	Manfiy son	-1245	$P = 40$

Algoritmi:

alg Butun_son (**but** Num, P)

arg Num

natija P

boshlab butun i, j, k, l
 $Num := \text{abs}(Num)$
 $i := Num \text{ div } 1000$
 $j := ((Num \text{ div } 100) \text{ mod } 10)$
 $k := ((Num \text{ div } 10) \text{ mod } 10)$
 $l := Num \text{ mod } 10$
 $P := i * j * k * l;$
Tamom

3-MA'RUZA MASHG'ULOTI:
Algoritm nazariyasining umumiy tushunchalari. Algoritm nazariyasiga kirish. Reduksiya prinsipi. Algoritm va hisoblash jaroyoni orasidagi bog'lanish

(ma'ruza – 2 soat)

2.1. Ma'ruza mavzusi bo'yicha reja-topshiriq va o'quv uslubiy xujjatlar.

Talabalar soni 30-40 ta	O'quv soati - 2 soat
Mashg'ulot shakli	Vizuallashtirilgan ma'ruza
Ma'ruza rejasi	2.1. Umumiy algoritmlar nazariyasiga doir tarixiy ma'lumotlar. 2.2. Reduktsiyaprintsipi. 2.3. Algoritm va hisoblash jarayoni orasidagi bog'lanish. 2.4. Algoritm tushunchasini formallashtirish
<i>O'quv mashg'ulotining maqsadi</i>	O'quv kursi haqida umumiy tasavvurni berish.
<i>Pedagogik vazifalar:</i>	<i>O'quv faoliyati natijalari:</i>
Umumiy algoritmlar nazariyasiga doir tarixiy ma'lumotlar beriladi;	Umumiy algoritmlar nazariyasiga doir tarixiy ma'lumotlarni aytib bera oladilar;
Reduksiya prinsipi haqida tushuncha beriladi;	Reduksiya prinsipi haqida tasavvurga ega bo'ladilar;
Algoritm va hisoblash jarayoni orasidagi bog'lanish tushuncha beriladi;	Algoritm va hisoblash jarayoni orasidagi bog'lanish haqida fikr yuritiladi;
Algoritm tushunchasini formallashtirishga doir ma'lumotlar beriladi;	Algoritm tushunchasini formallashtirish haqida tasavvurga ega bo'ladilar.
<i>O'qitish usullari-texnikasi</i>	<i>Ma'ruza, namoyish, blis-so'rov, aqliy hujum,</i>
<i>o'qitish vositalari</i>	<i>ma'ruza matni, kompyuter texnologiyasi, doska</i>
<i>o'qitish shakllari</i>	frontal, kollektiv ish
<i>o'qitish sharoiti</i>	kompyuter, multimedia proektor (yoki proektor va LCD panel)
<i>Monitoring va baholash</i>	<i>Test, og'zaki savollar</i>

Ma'ruza mashg'ulotining texnologik xaritasi.

Faoliyat	Faoliyatning mazmuni
----------	----------------------

bosqichlari	O'qituvchi	Talaba
1-bosqich. Kursga va mavzuga kirish (20 min)	1.1. O'quv kursining 2 – mavzusi bo'yicha dastlabki umumiy tasavvurni beradi va hamda uslubiy va tashkiliy tomonlari bilan tanishtiradi.	Tinglaydilar
	1.2. Mavzu rejasi va tayanch iboralar bilan tanishtiriladi. (2.1-va 2.3-ilovalar)	Yozadilar, Tinglaydilar
	1.3. Blis-so'rov usulida mavzu bo'yicha ma'lum bo'lgan tushunchalarni sanab berishni so'raydi (2.4-ilovalar).	Tushunchalarni sanab beradilar
2-bosqich. Asosiy bo'lim (50 min)	2.1. Mavzu rejasi va tayanch tushunchalar bilan tanishtiradi. Ma'ruza rejasining har bir punkt nihoyasida umumlashtirib boradi. Aqliy hujum usuli yordamida ushbu mavzu bo'yicha ma'lum bo'lgan tushunchalarni aytishni taklif etadi. Jarayon kompyuter slaydlarini namoyish qilish bilan olib boriladi.(2.6-ilova)	Tinglaydilar. konspekt qilib boradilar.
	2.2. Ma'ruzarejasining 1-4 punktlarigadoir nazariy ma'lumotlarni (2.6-Ilova) tushuntiradi, har bir punkt nihoyasida umumlashtirib boradi. Jarayon komp'yuter slaydlarini namoyish qilish bilan olib boriladi. Slaydlarni PowerPoint tartibidan namoyish va sharhlash bilan mavzu bo'yicha asosiy nazariy holatlarni bayon qiladi. Jalb qiluvchi savollar beradi. Berilayotgan ma'lumotlarni daftarga qayd etishlarini eslatadi.	Har bir tayanch tushuncha va iboralarni muhokama qiladilar. Barcha axborotni tizimlashtiradilar. Konspekt qiladilar.
	2.3. Ekranda nazorat uchun savollar va topshiriqlar namoyish etiladi. Savollarga javob beriladi va B/B/B jadvali va "Venn" diagrammasi to'ldiriladi (2.5-Ilova).	
3-bosqich. Yakunlovchi (10 min)	3.1. Mavzu bo'yicha yakunlovchi xulosalar qiladi. Mavzuda olingan bilimlarni qayerda ishlatish mumkinligi ma'lum qiladi.	Savollar beradilar
	3.2. Mavzu maqsadiga erishishdagi tinglovchilar faoliyati tahlil qilinadi va baholanadi.	
	3.3. Mavzu bo'yicha mustaqil o'rganish uchun topshiriqlar beradi.	Mustaqil o'rganish uchun topshiriqlarni yozib oladilar

2.1-ilova

Mavzuni jonlantirish uchun savollar

1. Algoritm asoschisi kim?
2. Algoritm nima uchun kerak?
3. Algoritm qanday xususiyatlarga ega?

2.2-ilova

Ma'ruza rejası

1. Umumiy algoritmlar nazariyasiga doir tarixiy ma'lumotlar.
2. Reduktsiyaprintsipi.
3. Algoritm va hisoblash jarayoni orasidagi bog'lanish.
4. Algoritm tushunchasini formallashtirish.

2.3-ilova

Tayanch iboralar

Alan Tyuring, Aloiz Chyorch, Emil Post, Tyuring mashinasi, Post mashinasi, Chyorchning lyamda- hisoblanuvchanlik usuli, PqNP muammosi.

2.4 -

1. Algoritm so'zi qayerdan kelib chiqadi?
2. Algoritmning qanday xossalari bilasiz?
3. Algoritmni yozishning qanday
4. usullari bilasiz?
5. Blok-sxema nima? Blok-sxemaning asosiy elementlarini ayting.
6. Algoritmning turlarini sanab o'ring.
7. Al-Xorazmiyning xati va ijodi xaqida nimalarni bilasiz?
8. Masalani echish algoritmini qanday usullar bilan ifodalash mumkin?
9. Algoritmga qo'yiladigan talablarni sanab o'ring.
10. Algoritmning qanday xossalari bilasiz?

ilova

1. B/B/B jadvali

2.5-ilova

Bilaman	Bilmoqchiman	Bilib oldim

2. Uchta turdagi algoritmlarga mos «VENN» diagrammasini yaratish

Ma'ruzaning mazmuni bo'yicha ko'rgazmali slaydlari

2.6-ilova

Umumiy algoritmlar nazariyasiga doir tarixiy ma'lumotlar Bizgacha etib kelgan intuitiv ma'nodagi algoritm eramizdan avvalgi III- asrda Evklid tomonidan taklif qilingan. Ushbu algoritm juda mashhur bo'lib, XX-asr boshlarigacha «algoritm» so'zining o'zi «Evklid algoritmi» ma'nosida ishlatilib kelindi. Boshqa matematika masalalari bosqichli echishni tasvirlash uchun esa «usul» so'zidan foydalanilgan. Zamonaviy algoritmlar nazariyasi rivojida boshlang'ich nuqta deb, nemis matematigi Kurt Gyodelning ilmiy ishini ko'rsatib o'tish mumkin (1931 y. Simvolik mantiqlarning to'lasligi to'g'risidagi teorema). Ushbu ishda ba'zi matematik muammolarni qaysidir sinfga taalluqli algoritmlar yordamida hal etib bo'lmazligi ko'rsatib berilgan. ... 1936 yilda Algoritm nazariyasi bo'yicha birinchi fundamental ilmiy ishlar bir-biridan alohida tarzda Alan Tyuring, Aloiz Chyorch va Emil Postlar e'lon qildilar. Ular tomonidan taklif etilgan Tyuring mashinasi, Post mashinasi va Chyorchning lyamda-hisoblanuvchanlik usuli algoritm formalizmining ekvivalent shakllaridir. Ular tomonidan taklif etilgan tezislar algoritm intuitiv tushunchasi va formal tizimlarning ekvivalentligini ta'kidlab berdi. Algoritmik echimsiz muammolarning formulirovkasi va isboti ushbu ishlarning muhim natijasi bo'ldi.	Umumiy algoritmlar nazariyasiga doir tarixiy ma'lumotlar • 1950- yillarda Algoritm nazariyasi rivojlanishiga rus matematiklari Kolmogorov va Markovlari z hissalari qo'shdilar. 60-70-yillarga kelib Algoritm nazariyasi fanida quyidagi mustaqil yo'nalishlar ajralib chiqdi: - Klassik algoritm nazariyasi (formal tillar terminlarida masalalarni ifodalash, echimli masala tushunchasi, 1965 yilda Edmonds tomonidan ta'riflangan PqNP muammosi, NP to'liq masalalar sinfining o'chilishi va tekshirishi); - Algoritmning asimptotik analizi nazariyasi (asimptotik baholash usullari, algoritmning murakkabligi, algoritm baho kriteriyalari va h.k.z.). Ushbu yo'nalish rivojiga Knut, Axo, Xopkroft, Ulman, Karp kabi olimlar o'z hissalari qo'shdilar; - hisoblash algoritmlarining praktik analizi nazariyasi (algoritmning mehnat talabligi oshkor funksiyasini topish, funksiyalarning chegaraviy analizi, rasional algoritmni tanlash metodikasi). Ushbu yo'nalish rivojlanishiga sabab bo'lgan ilmiy ish D.Knutning "Isskustvo programmirovaniya dlya EVM" kitobidan iborat.
--	--

Algoritm tushunchasini formallashtirish

Inson o'zining barcha faoliyat sohaslarida, jumladan axborotlarni qayta ishlashda ham masalalarni echishning turli usul va vositalari bilan to'qnashadi. Ular pirovard natijaga erishish uchun bajariladigan xarakatlar tartibini aniqlaydi. Buni intuitiv ma'nodagi algoritm tushunchasi deb qarashimiz mumkin. Ushbu tushunchaga qo'yiladigan ba'zi talablar esa algoritmni noformal aniqlash imkonini beradi:

Algoritm - qaysidir tilda berilgan masalani echish uchun bajariladigan boshlang'ich berilganlar ustida bajariladigan amallarning chekli ketma-ketligi.

D - masalaning boshlang'ich berilganlar sohasi (to'plami), R - mumkin bo'lgan natijalar to'plami bo'lsin. Bu holda algoritm $D \rightarrow R$ akslantirishni bajaradi deb hisoblashimiz mumkin. Ushbu akslantirish to'la bo'lmashligi mumkin bo'lgani uchun quyidagi tushunchalarni kiritamiz:

Algoritm qisman deyiladi, agar natija faqat ba'zi D lar uchun olinishi mumkin bo'lsa, to'la algoritm deyiladi, gar barcha D lar uchun natija olinishi mumkin bo'lsa.

Algoritm tushunchasini formallashtirish

• Olimlarning izchil faoliyatlariga qaramay, Algoritm tushunchasiga bitta konkret ta'rif berishning imkoni bo'lmadi. Algoritm nazariyasida algoritmning turli formal ta'riflari kiritilgan bo'lib, ularning ekvivalentligi isbotlangan.

• A.N. Kolmogorov ta'rif. Algoritm - bu qo'yilgan masala natijasiga qandaydir sondagi qadamlardan keyin olib keluvchi ma'lum qoidalar bo'yicha bajariluvchi har qanday hisoblash tizimi.

• A.A. Markov ta'rif. Algoritm - bu boshlang'ich berilganlardan izlangan natijaga olib keluvchi hisoblash jarayonini aniqlovchi aniq ko'rsatmalar.

- Algoritm tushunchasining turli ta'riflari bir qator talablarga javob berishi kerak:
 - algoritm chekli sondagi elementar bajariluvchi ko'rsatmalardan iborat bo'lishi kerak;
 - algoritm chekli sondagi qadamlardan iborat bo'lishi kerak;
 - algoritm barcha boshlang'ich berilganlar uchun umumiy bo'lishi kerak;
 - algoritm to'g'ri echimga olib kelishi kerak.

Tavsiya etiladigan adabiyotlar

1. **V.I. Igoshin. Matematicheskaya logika i teoriya algoritmov. Izdatelstvo Saratovskogo Universiteta, 1991, 209-211s.**
2. **O.P. Kuznesov. Diskretnaya matematikadlya injenera. M: Energoatomizdat, 1982, 144-155s.**
3. **N.A. Krinitskiy, G.A. Mironov, G.D. Frolov. Programmirovaniye i algoritmiteskie yaziki, M: Nauka, 1979, 63-66s.**
4. **E.Z. Lyubimskiy, V.V. Martinyuk, N.P. Trifonov. Programmirovaniye, M: Nauka, 1980, 13-17s.**
5. <http://structur.h1.ru/hash.htm>
6. <http://intsys.msu.ru/stuff/vnosov/theorald.htm#top>

4-MA'RUZA MASHG'ULOTI:

Algoritm larni baholash. Algoritm larni baholash, ularni mezonlari va tahlil qilish. Ikki tamonlama algoritm.

(ma'ruza – 2 soat)

1.1. Ma'ruza mavzusi bo'yicha reja-topshiriq va o'quv uslubiy xujjatlar.

Talabalar soni 30-40 ta	O'quv soati - 2 soat
Mashg'ulot shakli	Vizuallashtirilgan ma'ruza
Ma'ruza rejasi	1. Algoritmning tasvirlash usullari 2. Chiziqli algoritmlar 3. Tarmoqlanuvchi algoritmlar 4. Takrorlanuvchi algoritmlar 5. Algoritm ijrosini tekshirish
<i>O'quv mashg'ulotining maqsadi</i>	O'quv kursi haqida umumiy tasavvurni berish.
<i>Pedagogik vazifalar:</i>	<i>O'quv faoliyati natijalari:</i>
Algoritmning tasvirlash usullari bilantishtirish;	Algoritmning tasvirlash usullarini aytib bera oladilar;
Chiziqli algoritmlar haqida ma'lumot beriladi;	Chiziqli algoritmlarni tushuntirib bera oladilar;
Tarmoqlanuvchi algoritmlar haqida tushuncha beriladi;	Tarmoqlanuvchi algoritmlar haqida tasavvurga ega bo'ladilar;
Takrorlanuvchi algoritmlar haqida tushuncha beriladi;	Takrorlanuvchi algoritmlar haqida bahs yuritiladi;
Algoritm ijrosini tekshirish tushuntiriladi;	Algoritm ijrosini tekshirish haqida tushunchaga ega buladilar.

O'qitish usullari-texnikasi	Ma'ruza, namoyish, blis-so'rov, aqliy hujum, pinbord texnikasi
o'qitish vositalari	ma'ruza matni, kompyuter texnologiyasi, doska
o'qitish shakllari	frontal, kollektiv ish
o'qitish sharoiti	kompyuter, multimedia proektor (yoki proektor va LCD panel)
Monitoring va baholash	Test, og'zaki savollar

Ma'ruza mashg'ulotining texnologik xaritasi.

Faoliyat bosqichlari	Faoliyatning mazmuni	
	O'qituvchi	Talaba
1-bosqich. Kursga va mavzuga kirish (20 min)	1.1. O'quv kursining 3 – mavzusi bo'yicha dastlabki umumiy tasavvurni beradi va hamda uslubiy va tashkiliy tomonlari bilan tanishtiradi. Mavzuni jonlantirish uchun savollar beriladi. (1.1-ilova)	Tinglaydilar
	1.2. Mavzu rejasi va tayanch iboralar bilan tanishtiriladi. (1.2- va 1.3-ilovalar)	Yozadilar, tinglaydilar
	1.3. Blis-so'rov usulida mavzu bo'yicha ma'lum bo'lgan tushunchalarni sanab berishni so'raydi (1.4 -ilovalar).	Tushunchalarni sanab beradilar
2-bosqich. Asosiy bo'lim (50 min)	2.1. Ma'ruza rejasining 1-4 punktlari bo'yicha tushuntiradi, har bir punkt nihoyasida umumlashtirib boradi. Jarayon kompyuter slaydlarini namoyish qilish bilan olib boriladi.(1.6-ilova).	Tinglaydilar. konspekt qilib boradilar.
	2.2. Tayanch iboralarga qaytiladi. Tinglovchilar ishtirokida ular yana bir bor takrorlanadi Mavzuga oid tushuncha va iboralarni qay darajada organganligi B.B.B jadvali va Venn diagrammasini aniqlanadi.(1.5-ilova)	Har bir tayanch tushuncha va iboralarni muhokama qiladilar. Barcha axborotni tizimlashtiradilar. Konspekt qiladilar.
3-bosqich. Yakunlovchi (10 min)	3.1. Mavzu bo'yicha yakunlovchi xulosalar qiladi. Mavzuda olingan bilimlarni qayerda ishlatish mumkinligi ma'lum qiladi.	Savollar beradilar
	3.2. Mavzu maqsadiga erishishdagi tinglovchilar faoliyati tahlil qilinadi va baholanadi.	
	3.3. Mavzu bo'yicha mustaqil o'rganish uchun topshiriqlar beradi.	Mustaqil o'rganish uchun topshiriqlarni yozib oladilar

1.1-ilova

Mavzuni jonlantirish uchun savollar

1. Algoritm so'zi qayerdan kelib chiqadi?
2. Algoritmning qanday xossalari bilasiz?
3. Algoritmni yozishning qanday usullari bilasiz?
5. Blok-sxema nima? Blok-sxemaning asosiy elementlarini ayting.

1.2-ilova

Ma'ruza rejası

1. Algoritmning tasvirlash usullari
2. Chiziqli algoritmlar
3. Tarmoqlanuvchi algoritmlar
4. Takrorlanuvchi algoritmlar
5. Algoritm ijrosini tekshirish

1.3-ilova

Tayanch iboralar

Aniqlik, Diskretlik, Ommaviylik, Tushunarliklik, Natijaviylik, Blok-sxema, Algoritmik notasiya, chiziqli algoritmlar, tarmoqlanuvchi algoritmlar, takrorlanuvchi algoritmlar, algoritm ijrosini tekshirish

1.4-ilova

1. Intuitiv algoritm tushunishining moyiyati nimada?
2. Algoritmning intuitiv ta'riflaridan birini keltiring?
3. Formal algoritm deganda nimani tushunamiz?
4. Algoritm ob'ekti deganda nimani tushunamiz?
5. Algoritm ob'ektining tasviri deganda nimani tushunamiz?

1. B/B/B jadvali

2.5-ilova

Bilaman	Bilmoqchiman	Bilib oldim

2. Uchta turdagi algoritmlarga mos «VENN» diagrammasini yaratish

Ma'ruzaning mazmuni bo'yicha ko'rgazmali slaydlari

1.6-ilova

Algoritmning tasvirlash usullari

Yuqorida ko'rilgan misollarda odatda biz masalani yechish algoritmini so'zlar va matematik formulalar orqali ifodaladik. Lekin algoritm boshqa ko'rinishlarda ham berilishi mumkin. Biz endi algoritmning eng ko'p uchraydigan turlari bilan tanishamiz.

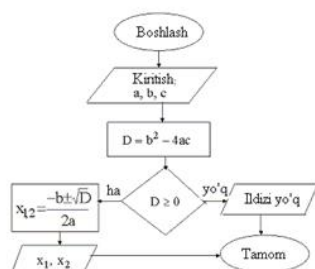
1. **Algoritmning so'zlar orqali ifodalanishi.** Bu usulda ijrochi uchun beriladigan har bir ko'rsatma jumalar, so'zlar orqali buyruq shaklida beriladi.
2. **Algoritmning formulalar bilan berilish** usulidan matematika, fizika, kimyo kabi aniq fanlardagi formulalarni o'rganishda foydalaniladi. Bu usulni ba'zan analitik ifodalash deyiladi.
3. **Algoritmning grafik shaklida tasvirlanishi**da algoritmlar maxsus geometrik figuralar yordamida tasvirlanadi va bu grafik ko'rinishi blok-sxema deyiladi.
4. **Algoritmning jadval ko'rinishida berilishi.** Algoritmning bu tarzda tasvirlanishidan ham ko'p foydalanamiz. Masalan, maktabda qo'llanib kelinayotgan to'rt xonali matematik jadvallar yoki turli xil lotereyalarda jadvali. Funksiyalarning grafiklarini chizishda ham algoritmning qiymatlari jadvali ko'rinishlaridan foydalanamiz. Bu kabi jadvalardan foydalanish algoritmlari sodda bo'lgan tufayli ularni o'zlashtirib olish oson.

Yuqorida ko'rilgan algoritmning tasvirlash usullarining asosiy maqsadi, qo'yilgan masalani yechish uchun zarur bo'lgan amallar ketma-ketligining eng qulay holatini aniqlash va shu bilan odam tomonidan programma yozishni yanada osonlashtirishdan iborat. Aslida programma ham algoritmning boshqa bir ko'rinishi bo'lib, u insonning kompyuter bilan muloqotini qulayroq amalga oshirish uchun mo'ljallangan.

Blok-sxemalarni tuzishda foydalaniladigan asosiy sodda geometrik figuralar quyidagilardan iborat:

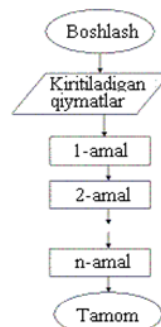
Nomi	Belgilanishi	Bajaradigan vazifasi
Jarayon		Bir yoki bir nechta amallarni bajarish natijasida ma'lumotning uzgariishi
Karor		Biror shartga bog'liq ravishda algoritmning bajarilish yonalishini tanlash
Shakl uzgartirish		Destini uzgartiruvchi buyruq yoki buyruqlar nuqtasini uzgartirish amali bajarish
Avval aniqlangan jarayon		Oldindan ishlab chikilgan dastur yoki algortmdan foydalanish
Kiritish CHikarish		Anborotlarni katta ishlash mumkin bo'lgan shakliga otkazish yoki olingan natijani tasvirlash
Display		Ekraniga ulangan displeydan anborotlarni kiritish yoki chikarish
Mujjat		Anborotlarni kogonga chikarish yoki kogondan kiritish
Anborotlar okimi chikishi		Blocklar orasidagi boglanishlarni tasvirlash
Boglagich		Uchlab kolgan anborot okimlarini ulash belgisi
Boshlash Tugatish		Anborotni katta ishlashni boshlash, yakunlash yoki butunlay nuktatish
Isx		Blocklarga tegishli nudi xildagi tushuntirishlar

$ax^2+bx+c=0$ kvadrat tenglamani yechish algoritmining blok-sxemasi



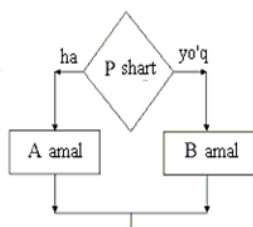
Chiziqli algoritmlar blok - sxemasining umumiy strukturasi

Faqat ketma-ket bajariladigan amallardan tashkil topgan algoritmlarga-chiziqli algoritmlar deyiladi. Bunday algoritmnii ifodalash uchun ketma-ketlik strukturasi ishlatiladi. Strukturada bajariladigan amal mos keluvchi shakl bilan ko'rsatiladi.



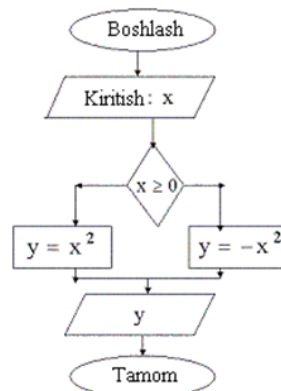
Tarmoqlanuvchi algoritmlar

Agar hisoblash jarayoni biror bir berilgan shartning bajarilishiga qarab turli tarmoqlar bo'yicha davom ettirilsa va hisoblash jarayonida har bir tarmoq faqat bir marta bajarilsa, bunday hisoblash jarayonlariga tarmoqlanuvchi algoritmlar deyiladi. Tarmoqlanuvchi algoritmlar uchun ayri strukturasi ishlatiladi. Tarmoqlanuvchi strukturasi berilgan shartning bajarilishiga qarab ko'rsatilgan tarmoqdan faqat bittasining bajarilishini ta'minlaydi.



1- Misol: $y = \begin{cases} x^2 & \text{agar } x \geq 0 \\ -x^2 & \text{agar } x < 0 \end{cases}$

Berilgan x ning qiymatiga bog'lik holda, agar u musbat bo'lsa «ha» tarmoq bo'yicha $y = x^2$ funksiyaning qiymati, aks holda $y = -x^2$ funksiyaning qiymati hisoblanadi.



Takrorlanuvchi algoritmlar

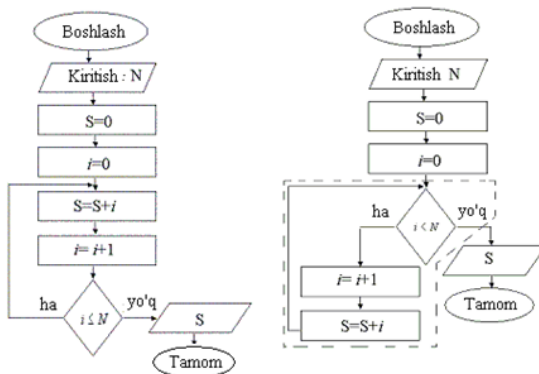
Agar biror masalani yechish uchun tuzilgan zarur bo'lgan amallar ketma-ketligining ma'lum bir qismi biror parametrga bog'liq ko'p marta qayta bajarilsa, bunday algoritmlar takrorlanuvchi algoritmlar yoki siklik algoritmlar deyiladi. Takrorlanuvchi algoritmlarga tipik misol sifatida odatda qatorlarning yig'indisi yoki ko'patmasini hisoblash jarayonlarini qarash mumkin. Quyidagi yig'indini hisoblash algoritmini tuzaylik.

$$S = 1 + 2 + 3 + \dots + N = \sum_{i=1}^N i$$

Bu yig'indini hisoblash uchun $i \geq 0$ da $S \geq 0$ deb olamiz va $i \leq 1$ da $S \leq 1$ ni hisoblaymiz. Bu yerda birinchi va ikkinchi qadamlar uchun yig'indi hisoblandi va keyingi qadamda i parametr yana bittaga o'rtiriladi va navbatdagi raqam avvalgi yig'indi S ning ustiga qo'shiladi va bu jarayon shu tartibda to $i \leq N$ sharti bajarilmaguncha davom ettiriladi va natijada izlangan yig'indiga ega bo'lamiz. Bu fikrlarni quyidagi algoritm sifatida ifodalash mumkin:

N - berilgan bo'lsin,
 $i=0$ berilsin,
 $S=0$ berilsin,
 $i=i+1$ hisoblansin,
 $S=S+i$ hisoblansin,
 $i \leq N$ tekshirilsin va bu shart bajarilsa, 4-satrga qaytilsin, aks holda keyingi qatorga o'tilsin,
 S ning qiymati chop etilsin.

1 dan n gacha bo'lgan sonlar yig'indisini hisoblash algoritmi



Ichma-ich joylashgan siklik algoritmlar.

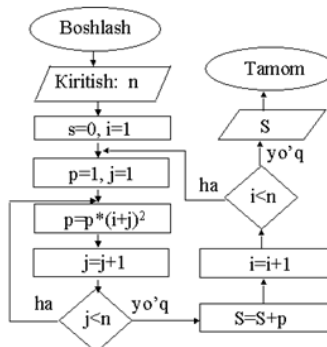
Ba'zan, takrorlanuvchi algoritmlar bir nechta parametrlarga bog'liq bo'ladi. Odatda bunday algoritmlarni ichma-ich joylashgan algortmlar deb ataladi.

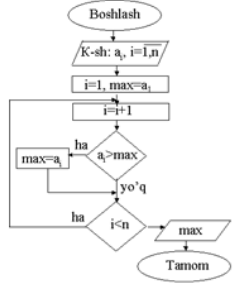
Misol sifati berilgan $n \times m$ o'lchovli a_{ij} -matritsa elementlarining yig'indisini hisoblash masalasini qaraylik.

$$S = \sum_{i=1}^n \sum_{j=1}^m (i+j)^2$$

Bu yig'indi hisoblash uchun, i ning har bir qiymatida j bo'yicha ko'paytmani hisoblab, avval yig'indi ustiga ketma-ket qo'shib borish kerak bo'ladi. Bu jarayon quyidagi blok-sxemada aks ettirilgan. Bu yerda i -tashqi sikl - yig'indi uchun, j -esa ichki sikl-ko'paytmani hosil qilish uchun foydalanilgan.

Ichma-ich joylashgan siklik algoritmgaga doir blok-sxema



Algoritm ijrosini tekshirish Kompyuter uchun tuzilgan algoritm ijrochisi-bu kompyuterdir. Biror programmalash tilida yozilgan algoritm kodlashtirilgan oddiy ko'rsatmalar ketma-ketligiga o'tadi va mashina tomonidan avtomatik ravishda bajariladi. Metodik nuqtayi-nazardan qaraganda algoritmning birinchi ijrochisi sifatida o'quvchining o'zini olish muhim ahamiyatga ega. O'quvchi tomonidan biror masalani yechish algoritmi tuzilganda bu algoritmni to'g'ri natija berishini tekshirish juda muhimdir. Buning yagona usuli o'quvchi tomonidan algoritmni turli boshlang'ich ma'lumotlarda qadamba - qadam bajarib (ijro etib) ko'rishdir. Algoritmni bajarish natijasida xatolar aniqlanadi va to'g'rilanadi. Ikkinchi tomonidan, masalani yechishga qiyalayotgan o'quvchi uchun tayyor algoritmni bajarish – masalani yechish yo'llarini tushunishga xizmat qiladi.	Algoritm ijrosini quyidagi <i>misol</i>da ko'raylik. Berilgan sonlarning eng kattasini topish algoritmini tuzaylik. Buning uchun, berilgan sonlardan birinchisi ni eng katta qiymat deb faraz qilaylik va uni max nomli yangi o'zgaruvchiga uzataylik: $max=a_1$. Parametr i ning qiymatini bittaga oshirib, ya'ni $i=i+1$ a_1 ni a_2 bilan taqqoslaymiz va qaysi biri katta bo'lsa uni max o'zgaruvchisiga uzatamiz va jarayonni shu tarzda to $i=n$ bo'lguncha davom ettiramiz. Bu fikrlar quyidagi blok-sxemada o'z aksini topgan. Vektor elementlarining eng kattasini topish algoritmi. 
Endi bu blok-sxema yoki algoritmning ijrosini , , aniq sonlarda ko'rib o'taylik: $i=1$ da $max=3$ bo'ladi. $i=i+1=2$ ni topamiz, $a_2 > max$, ya'ni $5 > 3$ ni tekshiramiz, shart bajarilsa, $max=5$ bo'ladi. $i < n$, ya'ni $2 < 3$ ni tekshiramiz. Shart bajarilsa, i ni yana bittaga oshiramiz, va $i=3$ bo'ladi, va $a_3 > max$, ya'ni $1 > 5$, ni tekshiramiz. Shart bajarilmadi, demak, keyingi $i < n$ shartni, ya'ni $3 < 3$ ni tekshiramiz. Shart bajarilmadi. Demak $max=5$ chop etiladi. Biz blok-sxemani tahlil qilish davomida uning to'g'riligiga ishonch hosil qildik. Endi ixtiyoriy n lar uchun bu blok-sxema bo'yicha eng katta elementni topish mumkin.	

5-MA'RUZA MASHG'ULOTI:

Ro'yxat va daraxtlar. Chiziqli ro'yxatlar va ular ustida amallar.

(ma'ruza – 2 soat)

1.1. Ma'ruza mavzusi bo'yicha reja-topshiriq va o'quv uslubiy xujjatlar.

Talabalar soni 30-40 ta	O'quv soati - 2 soat
Mashg'ulot shakli	Vizuallashtirilgan ma'ruza
Ma'ruza rejasi	1. Algoritmnlarni baholash ma'zonlari 2. Tahlil qilish usullari. 3. Chiziqli ro'yhatlar 4. Chiziqli ro'yhatlar ustida asosiy amallar.
<i>O'quv mashg'ulotining maqsadi</i>	O'quv kursi haqida umumiy tasavvurni berish.
<i>Pedagogik vazifalar:</i>	<i>O'quv faoliyati natijalari:</i>
Algoritmnlarni baholash ma'zonlari bilan tanishtirish;	Algoritmnlarni baholash ma'zonlarini aytib bera oladilar;
Tahlil qilish usullari haqida ma'lumot beriladi;	Tahlil qilish usullarini tushuntirib bera oladilar;
Chiziqli ro'yhatlar haqida tushuncha beriladi;	Chiziqli ro'yhatlar haqida tasavvurga ega bo'ladilar;
Chiziqli ro'yhatlar ustida bajariladigan asosiy amallarhaqida tushuncha beriladi;	Chiziqli ro'yhatlar ustida bajariladigan asosiy amallarhaqida fikrlay oladilar;
O'qitish usullari-texnikasi	Ma'ruza, namoyish, blis-so'rov, aqliy hujum,

	pinbord texnikasi
o'qitish vositalari	ma'ruza matni, kompyuter texnologiyasi, doska
o'qitish shakllari	frontal, kollektiv ish
o'qitish sharoiti	kompyuter, multimedia proektor (yoki proektor va LCD panel)
Monitoring va baholash	Test, og'zaki savollar

Ma'ruza mashg'ulotining texnologik xaritasi.

Faoliyat bosqichlari	Faoliyatning mazmuni	
	O'qituvchi	Talaba
1-bosqich. Kursga va mavzuga kirish (20 min)	1.1. O'quv kursining 4 – mavzusi bo'yicha dastlabki umumiy tasavvurni beradi va hamda uslubiy va tashkiliy tomonlari bilan tanishtiradi. Mavzuni jonlantirish uchun savollar beriladi. (1.1-ilova)	Tinglaydilar
	1.2. Mavzu rejasi va tayanch iboralar bilan tanishtiriladi. (1.2- va 1.3-ilovalar)	Yozadilar, tinglaydilar
	1.3. Blis-so'rov usulida mavzu bo'yicha ma'lum bo'lgan tushunchalarni sanab berishni so'raydi (1.4 -ilovalar).	Tushunchalarni sanab beradilar
2-bosqich. Asosiy bo'lim (50 min)	2.1. Ma'ruza rejasining 1-4 punktlari bo'yicha tushuntiradi, har bir punkt nihoyasida umumlashtirib boradi. Jarayon kompyuter slaydlarini namoyish qilish bilan olib boriladi.(1.6-ilova).	Tinglaydilar. konspekt qilib boradilar.
	2.2. Tayanch iboralarga qaytiladi. Tinglovchilar ishtirokida ular yana bir bor takrorlanadi Mavzuga oid tushuncha va iboralarni qay darajada organganligi B.B.B jadvali va Venn diagrammasini aniqlanadi.(1.5-ilova)	Har bir tayanch tushuncha va iboralarni muhokama qiladilar. Barcha axborotni tizimlashtiradilar. Konspekt qiladilar.
3-bosqich. Yakunlovchi (10 min)	3.1. Mavzu bo'yicha yakunlovchi xulosalar qiladi. Mavzuda olingan bilimlarni qayerda ishlatish mumkinligi ma'lum qiladi.	Savollar beradilar
	3.2. Mavzu maqsadiga erishishdagi tinglovchilar faoliyati tahlil qilinadi va baholanadi.	
	3.3. Mavzu bo'yicha mustaqil o'rganish uchun topshiriqlar beradi.	Mustaqil o'rganish uchun topshiriqlarni yozib oladilar

1.1-ilova

Mavzuni jonlantirish uchun savollar

1. Algoritm so'zi qayerdan kelib chiqadi?
2. Algoritmning qanday xossalari bilasiz?
3. Algoritmni yozishning qanday usullari bilasiz?
5. Blok-sxema nima? Blok-sxemaning asosiy elementlarini ayting.

1.2-ilova

Ma'ruza rejasi

1. Algoritmning tasvirlash usullari
2. Chiziqli algoritmlar
3. Tarmoqlanuvchi algoritmlar
4. Takrorlanuvchi algoritmlar
5. Algoritm ijrosini tekshirish

1.3-ilova

Tayanch iboralar

Aniqlik, Diskretlik, Ommaviylik, Tushunarliklik, Natijaviylik, Blok-sxema, Algoritmik notasiya, chiziqli algoritmlar, tarmoqlanuvchi algoritmlar, takrorlanuvchi algoritmlar,

1.4-ilova

1. Intuitiv algoritm tushunmasining mohiyati nimada?
2. Algoritmning intuitiv ta'riflaridan birini keltiring?
3. Formal algoritm deganda nimani tushunamiz?
4. Algoritm ob'ekti deganda nimani tushunamiz?
5. Algoritm ob'ekting tasviri deganda nimani tushunamiz?

1. B/B/B jadvali

2.5-ilova

Bilaman	Bilmoqchiman	Bilib oldim

2. Uchta turdagi algoritmlarga mos «VENN» diagrammasini yaratish

Ma'ruzaning mazmuni bo'yicha ko'rgazmali slaydlari

1.6-ilova

Ro'yxat Ro'yxat – bu berilganlarning ketma-ket tashkillashtirilgan strukturasi. Chiziqli ro'yxatlarning massivlardan farqi shundaki, ular dastur bajarilishi jarayonida o'z xajmini o'zgartirish imkoniyatiga ega. Binobarin ro'yxatlarning xajmi oldindan aniqlanmaydi. Chiziqli ro'yxatni zanjir qismlari ko'rinishida tasvirlash mumkin. Massivda elementlarni ketma-ket joylashtirish bevosita (indekslash orqali) amalga oshiriladi. Ro'yxat elementlari esa maxsus usulda joylashtirilib, uning elementlari axborotlar hamda keyingi qism adresini saqlovchi tugunlarda saqlanadi. Ushbu tugun va adresni quyidagicha e'lon qilish mumkin: <pre> Type Link = ^Node; Node = record Data: integer; Next: Link; End; </pre>	Ro'yxatni e'lon qilish uchun ikkita qo'ishimcha head va z tugunlaridan foydalanamiz. Head ro'yxatning birinchi elementini ko'rsatadi, z esa oxirgi elementini ko'rsatadi. Bunda ro'yxatni quyidagicha ifodalash mumkin bo'ladi: Berilganlarning bunday strukturasi ma'lumotlar ustida amallar bajarishning massivlardan ko'ra ancha effektivroq usullarni qo'llashga imkon beradi. Masalan, agar 1-elementni ro'yxat boshidan oxiriga o'tqazmoqchi bo'lsak, massivning barcha elementlarini 1- elementga joy bo'shatish uchun 1 pozitsiya o'ngga siljitishga to'g'ri keladi. Ro'yxatda esa shu amalni bajarish uchun faqat adreslar o'zgartirilishi kerak holos. Bunda 1-elementni saqlovchi tugun ko'rsatkichini 2-elementni saqlovchi tugunga o'rnatib, head bo'sh tugun ko'rsatkichini esa 1-element ni saqlovchi tugunga o'rnatamiz. Bundan tashqari berilganlarning ro'yxat strukturasi ketma-ketlikka yangi element qo'shish imkoniyatini yaratadi. Bunda ro'yxat uzunligi bitta elementga uzayadi. Quyidagi rasmda ro'yxatga yangi element qo'shish prosedurasini ifodalangan. Avvalo ushbu dinamik element yaratiladi, so'ngra yangi elementning ko'rsatkichi q tugunga to'g'irlanadi va oxirida R tugunning ko'rsatkichi yangi tugunga to'g'irlanadi. Xuddi shuningdek, ro'yxatdan element olib tashlash prosedurasini ham oson bajarish mumkin. Bunda r elementning ko'rsatkichi q dan keyin keluvchi elementga to'g'irlanadi.
--	---

Boshqa tomondan qaraganda, shunday amallar borki, berilganlarning ro'yxat strukturasi ularni bajarishda ma'lum noqulayliklarni tug'diradi. Bunday proseduralarga misol sifatida k-elementni topish masalasini keltirish mumkin. Massivda bu prosedura a[k] ga murojaat bilan oson hal etiladi. Ro'yxatda esa k ta adresni ko'rib chiqishga to'g'ri keladi. Xuddi shunday berilgan element oldidagi elementni topish ro'yxat uchun notabiiy amal bo'lib hisoblanadi. Bu muammoni hal etish uchun masalalarning formulirovkasi berilgan elementni olib tashlash, qo'shish o'rniga berilgan elementdan keyingi elementni olib tashlash yoki berilgan elementdan keyin element qo'shish shakliga almashtiriladi. Paskal tilida chiziqli ro'yxatlarni yaratish va ular ustida amal bajarish vositalari mavjud bo'lib, quyidagi proseduralar yuqorida to'xtalib o'tilgan ro'yxatlar ustida bajariluvchi sodda amallarni realizatsiya qiladi:	<pre> Type Link = ^Node; Node = record Data: integer; Next: Link; End; Var Head, z: link; procedure list_initialize; begin new(head); new(z); head^.next := z; z^.next := nil; end; procedure insert_after(v : integer; t : link); var x : link; begin new(x); x^.data := v; x^.next := t^.next; t^.next := x; end; procedure delete_next(t : link); var del: link; begin del := t^.next; t^.next := t^.next^.next; dispose(del); end; </pre>
Ushbu proseduralarni batafsilroq ko'rib o'taylik. Link = ^Node; - bu yerda yangi Link tipi yaratilib, u Node toifasidagi ko'rsatkichdan iboratdir. Ko'rsatkich bu - butun toifali o'zgaruvchi bo'lib, berilganlarning qandaydir elementini saqlovchi xotira bayti adresini saqlaydi. Ushbu terminning ma'nosiga alohida to'xtalamiz. Kompyuter xotirasini quyidagicha tasvirlash mumkin: xotira segment deb ataluvchi alohida bloklardan iborat. Dos da har segment nomeri maksimal 16 bitda iborat bo'lishi mumkin. Ixtiyoriy segment [0; \$FFFF] oralig'idagi nomerga ega bo'ladi. Bunda \$ belgi 16 lik sanoq sistemasidagi sonni ifodalaydi. \$592B, \$592C, \$592D segmentli xotira sohasini ko'rib chiqaylik. har bir segment ichidagi xotira yacheykalari ham o'z navbatida adreslarga ega. Bu adreslar ham [0; \$FFFF] oralig'idagi nomerlardan iborat. Masalan, \$592C segmentidagi xotira yacheykasining nomeri \$B401 bo'lsa, ushbu xotira yacheykasiga ko'rsatkichning qiymati \$592CB401 dan iborat bo'ladi.	(procedure list_initialize; new(head); - new prosedurasi node toifasidagi yangi dinamik o'zgaruvchi yaratadi va head o'zgaruvchisining qiymatini yangi yaratilgan o'zgaruvchini ko'rsatadigan qilib belgilaydi, ya'ni dastur xotirada 6(\$6) bayt uzunlikdagi bo'sh joy qidirib topib, bu sohani band deb e'lon qiladi. So'ngra dastur head o'zgaruvchisiga ushbu rezervlangan joy adresini o'zlashtiradi. Faraz qilaylik, dastur \$592CB401 adresli xotira sohasini topib, bu nomerni head o'zgaruvchisiga o'zlashtirsin. new(z) - dastur xotiradagi \$592D000A adresli sohani topib, uning adresini z ga o'zlashtirsin; Xotira adresi mazmuniga murojaat qanday amalga oshiriladi, degan savolga quyidagi misol vositasida javob beramiz: Masalan, head^.key:=10; operatori bajarilganda \$592CB401 adresli xotira yacheykasining key deb nomlanuvchi 2 baytli qismiga 10 soni kiritiladi. head^.next := z; - bu yerda esa \$592CB401 yacheykasining Next deb ataluvchi qismiga \$592D000A sonini kiritamiz. z^.next := nil - z bilan adreslanuvchi xotira yacheykasini nollar bilan to'ldiramiz.

5-MA'RUZA MASHG'ULOTINING DAVOMI:

Tarmoqlar. Daraxtlar va ularning turlari. Saralash va izlash algoritmlari. Tanlash va joylash algoritmlari. Izlashning asosiy prinsiplari. Izlash usullari. Saralash algoritmlarini taqqoslash.

(ma'ruza – 2 soat)

1.1. Ma'ruza mavzusi bo'yicha reja-topshiriq va o'quv uslubiy xujjatlar.

Talabalar soni 30-40 ta	O'quv soati - 2 soat
Mashg'ulot shakli	Vizuallashtirilgan ma'ruza
Ma'ruza rejasi	1. Hisoblanuvchi funksiyalar va sanoqli to'plamlar 2. Hisoblanuvchi funksiyalarga oid Tyuring tezisi 3. Tyuring mashinalari va EHmlar
<i>O'quv mashg'ulotining maqsadi</i>	O'quv kursi haqida umumiy tasavvurni berish.
<i>Pedagogik vazifalar:</i>	<i>O'quv faoliyati natijalari:</i>
Algoritmlarni baholash ma'zonlari bilan tanishtirish;	Algoritmlarni baholash ma'zonlarini aytib bera oladilar;
Tahlil qilish usullari haqida ma'lumot beriladi;	Tahlil qilish usullarini tushuntirib bera oladilar;

Chiziqli ro'yhatlar haqida tushuncha beriladi;	Chiziqli ro'yhatlar haqida tasavvurga ega bo'ladilar;
Chiziqli ro'yhatlar ustida bajariladigan asosiy amallarhaqida tushuncha beriladi;	Chiziqli ro'yhatlar ustida bajariladigan asosiy amallarhaqida fikrlay oladilar;
O'qitish usullari-texnikasi	Ma'ruza, namoyish, blis-so'rov, aqliy hujum, pinbord texnikasi
o'qitish vositalari	ma'ruza matni, kompyuter texnologiyasi, doska
o'qitish shakllari	frontal, kollektiv ish
o'qitish sharoiti	kompyuter, multimedia proektor (yoki proektor va LCD panel)
Monitoring va baholash	Test, og'zaki savollar

Ma'ruza mashg'ulotining texnologik xaritasi.

Faoliyat bosqichlari	Faoliyatning mazmuni	
	O'qituvchi	Talaba
1-bosqich. Kursga va mavzuga kirish (20 min)	1.1. O'quv kursining 4 – mavzusi bo'yicha dastlabki umumiy tasavvurni beradi va hamda uslubiy va tashkiliy tomonlari bilan tanishtiradi. Mavzuni jonlantirish uchun savollar beriladi. (1.1-ilova)	Tinglaydilar
	1.2. Mavzu rejasi va tayanch iboralar bilan tanishtiriladi. (1.2-va 1.3-illovalar)	Yozadilar, tinglaydilar
	1.3. Blis-so'rov usulida mavzu bo'yicha ma'lum bo'lgan tushunchalarni sanab berishni so'raydi (1.4 -illovalar).	Tushunchalarni sanab beradilar
2-bosqich. Asosiy bo'lim (50 min)	2.1. Ma'ruza rejasining 1-4 punktlari bo'yicha tushuntiradi, har bir punkt nihoyasida umumlashtirib boradi. Jarayon kompyuter slaydlarini namoyish qilish bilan olib boriladi.(1.6-ilova).	Tinglaydilar. konspekt qilib boradilar.
	2.2. Tayanch iboralarga qaytiladi. Tinglovchilar ishtirokida ular yana bir bor takrorlanadi Mavzuga oid tushuncha va iboralarni qay darajada organganligi B.B.B jadvali orqali aniqlanadi.(1.5-ilova)	Har bir tayanch tushuncha va iboralarni muhokama qiladilar. Barcha axborotni tizimlashtiradilar. Konspekt qiladilar.
3-bosqich. Yakunlovchi (10 min)	3.1. Mavzu bo'yicha yakunlovchi xulosalar qiladi. Mavzuda olingan bilimlarni qayerda ishlatish mumkinligi ma'lum qiladi.	Savollar beradilar
	3.2. Mavzu maqsadiga erishishdagi tinglovchilar faoliyati tahlil qilinadi va baholanadi.	
	3.3. Mavzu bo'yicha mustaqil o'rganish uchun topshiriqlar beradi.	Mustaqil o'rganish uchun topshiriqlarni yozib oladilar

1.1-ilova

Mavzuni jonlantirish uchun savollar

1. Intuitiv algoritm tushunmasining moiylati nimada?
2. Algoritmning intuitiv ta'riflaridan birini keltiring?
3. Formal algoritm deganda nimani tushunamiz?
4. Algoritm ob'ekti deganda nimani tushunamiz?

1.2-ilova

Ma'ruza rejasi

1. Hisoblanuvchi funksiyalar va sanoqli to'plamlar
2. Hisoblanuvchi funksiyalarga oid Tyuring tezisi
3. Tyuring mashinalari va EHMlar

1.3-ilova

Tayanch iboralar

Hisoblanuvchi funksiyalar, Tezis, EHM

1.4-ilova

1. Xisoblanuvchi funksiya nima?
2. Tyuring buyicha xisoblanuvchi funksiyalar deganda nimani tushunamiz?
3. Sanoqli to'plam deb nimaga aytiladi?
4. Echimli to'plam deb nimaga aytiladi?

3. B/B/B jadvali

2.5-ilova

Bilaman	Bilmoqchiman	Bilib oldim

Ma'ruzaning mazmuni bo'yicha ko'rgazmali slaydlari

1.6-ilova

$\{0,1,2,\dots,n,\dots\}$ Natural sonlar to'plamida berilgan bir yoki bir necha argumentli f funksiyalarni ko'rib o'taylik Hamda ushbu funksiyalar shu to'plamda qiymatlar qabul qiladi deb hisoblaylik. f funksiyaning aniqlanish sohasi $D_f = N^n = N \times N \times \dots \times N$ to'plamning qism to'plami bo'lsin. $D_f = \{(x_1, x_2, \dots, x_n) : f(x_1, x_2, \dots, x_n) \text{ aniqlangan}\}$ f ning qiymatlar sohasi N to'plamning qism to'plami bo'ladi: $R_f = \{y \in N : \exists (x_1, x_2, \dots, x_n) (f(x_1, x_2, \dots, x_n) = y)\}$ 1-Ta'rif. $f(x_1, x_2, \dots, x_n)$ Funksiyaning qiymatlarini o'zi aniqlangan argumentlar naborlari uchun hisoblovchi algoritm mavjud bo'lsa, hamda ushbu argumentlar nabori uchun funksiya aniqlanmagan bo'lgan holda ushbu algoritm cheksiz bajarilsa, funksiya hisoblanuvchi deyiladi. $M \subseteq N^n = N \times N \times \dots \times N$ bo'lsin. 2-Ta'rif. M to'plam echimli deyiladi, agar ixtiyoriy elementni unga tegishli yoki tegishli emasligini aniqlab beruvchi algoritm mavjud bo'lsa. M to'plamning xarakteristik funksiyasi degan tushunchani kiritaylik. X_M funksiya to'plamning xarakteristik funksiyasi deb ataladi, qachonki X_M M to'plamda berilgan bo'lsa, hamda 2 elementli $\{0,1\}$ to'plamda qiymatlar qabul qilib, quyidagicha aniqlansa: $X_M(x_1, x_2, \dots, x_n) = \begin{cases} 0, & \text{agar } (x_1, x_2, \dots, x_n) \notin M \\ 1, & \text{agar } (x_1, x_2, \dots, x_n) \in M \end{cases}$	Bundan kelib chiqadiki, M to'plam echimli bo'ladi, qachonki uning xarakteristik funksiyasi X_M hisoblanuvchi bo'lsa. $M \subseteq N$ bo'lsin 3-Ta'rif $M \subseteq N$ to'plam algoritmik (Rekursiv) sanoqli deyiladi, qachonki, M bo'sh to'plam bo'lsa, yoki qandaydir hisoblanuvchi funksiyaning aniqlanish sohasi mavjud bo'lib, shunday algoritm topiladiki, M to'plamning barcha elementlarini hosil qilsa. Misol. $M_q = \{1, 4, 9, 16, 25, 36, \dots\}$ to'plam berilgan bo'lsin. Bu to'plam natural sonlarning kvadratlari to'plamidir. Uning elementlarini hosil qilish uchun $1, 2, \dots$ sonlarni kvadratga ko'tarish kerak. Boshqacha aytganda M to'plam $f(x) = x^2 : M \rightarrow \{1^2, 2^2, \dots\}$ funksiyaning qiymatlar sohasidir. Bundan tashqari bu to'plam echimlidir ham. Buni isbotlash mumkin. Ta'rifdan kelib chiqqan xolda, ixtiyoriy elementni to'plamga tegishli ekanligini tekshirish uchun uni tub ko'paytuvchilarga ajratib, biror sonning aniq kvadrati ekanligini, yoki aksinchaligini tekshirib ko'rish mumkin. 1-Teorema. Agar M va to'plamlar sanoqli bo'lsa, $M \cap L$ va $M \cup L$ to'plamlar ham sanoqli. Isbot. Agar M va L to'plamlar elementlarini hosil qiluvchi algoritmlar mavjud bo'lsa, $M \cap L$ to'plam elementlari berilgan algoritmlarning bir vaqtda qo'llanilishi orqali hosil qilinadi. Q_M algoritm M to'plam elementlarini hosil qilsin; Q_L algoritm esa L to'plam elementlarini hosil qilsin;
--	--

U holda $M \cap L$ tuplam elementlarni hosil qiluvchi algoritmnin mohiyati quyidagicha bo'ladi: Q_M, Q_L algoritmlar yordamida navbat bilan $m_1, m_2, m_3, \dots$ elementlar hosil qilinadi. Har bir hosil qilingan m_i element oldin hosil qilingan i elementlar bilan taqqoslanadi. Agar m_i birorta i bilan mos tushib qolsa, u $M \cap L$ to'plamga kiritiladi. Aks holda i element hosil qilinib, uni oldin hosil qilingan m_i lar bilan taqqoslash kerak bo'ladi va h.z. Ushbu jarayon $M \cap L$ to'plamning barcha elementlarni sanab o'tish imkonini beradi. Bu esa teoremaning isbotidan iborat. 2-Teorema. $M \subseteq N$ bo'lsin. M to'plam echimli buladi faqat va faqat uning o'zi va uning kengaytmasi sanoqli bo'lsa. Isbot. Zaruriyligi. M echimli to'plam bo'lsin. M to'plam bo'sh bo'lmasin, ya'ni shunday a element mavjud bo'lsinki, $a \in M$ bo'lsin. U holda uning xarakteristik funksiyasi X_M hisoblanuvchi, ya'ni uni hisoblovchi algoritim mavjud. M to'plamni hosil qiluvchi algoritim ko'raylik: $x, \text{ agar } X_M(x)=1$ $F(x)=$ $a, \text{ agar } X_M(x)=0$ Bundan $M = \{f(0), f(1), \dots\}$, ya'ni M to'plam f funksiyaning qiymatlar to'plami ekanligi kelib chiqadi. f funksiyaning X_M ning hisoblanuvchi ekanligidan hisoblanuvchi ekanligi kelib chiqadi. Bundan ko'rinadi, M to'plam sanoqlidir. Xudi shuningdek, $b \in M$ element tanlab, $b, \text{ agar } X_M(x) \neq 1$	Eslatib utishimiz kerakki, algoritmlarning asosiy xususiyatlaridan biri shuki, u kandaydir bir tipdagi masalalarning cheksiz tuplamidan xar bir masalani echish usulidan iborat bulib, bu usul ushbu masala echimini chekdi kadamda topish imkonini beradi. Ammo algoritim tushunchasiga boshqa nuktai nazardan xam karash mumkin. SHu cheksiz bir tipdagi masalalar tuplamidan olinadigan xar bir masalani kandaydir alfavitning kaysidir suzi bilan ifodalash (kodlash) mumkin. Uning echimini esa shu alfavitning boshka kandaydir suzi bilan ifodalash mumkin. Natijada tanlangan alfavitning barcha suzlari tuplamining biror kism tuplamida aniklangan funksiyaga ega bulamiz. Bu funksiyaning kiymatlar soxasi shu alfavit barcha suzlari tuplamidan iborat buladi. Bundan shu kelib chikadiki, biror masalaning echimini topish deganda, uni bu funksiyaning berilgan masalani kodlovchi suzdagi kiymatini topish tushuniladi. Berilgan masalar sinfini echish algoritimga ega bulish deganda esa kurilgan funksiya argumentlarining funksiya aniklanish soxasidan olingan ixtiyoriy kiymatlari uchun funksiya kiymatini chekdi kadamda xisoblovchi usulga ega bulish demakdir. SHu tarzda algoritmik muammo bu – kandaydir alfavitda berilgan funksiyani xisoblash muammosidir degan xulosaga kelish mumkin. Endi funksiya kiymatini xisoblash deganda nimani tushunmok kerak? degan savolga javob berishimiz kerak. Bu savolga shunday javob berishimiz mumkin: funksiya kiymatini mos Tyuring mashinasi yordamida topish. Kanday funksiyalarni Tyuring buyicha xisoblanuvchi deya olamiz? Olimlarning tadqiqotlari, juda kur amaliy ishlar bunday funksiyalar sinfining ulkan ekanligini kursatdi. Kiymatini xisoblovchi algoritimga ega bulgan xar bir funksiya Tyuring buyicha xisoblanuvchi bulib chikdi. Bu xulosalar Tyuringa algoritmlar nazariyasi asosiy gipotezasi deb ataluvchi kuyidagi iezisini taklif etish imkonini berdi.:
Tyuring tezisi: Kandaydir alfavitda berilgan funksiya kiymatini xisoblovchi algoritim fakat va fakat funksiya Tyuring buyicha xisoblanuvchi bulganda, ya'ni uni mos Tyuring mashinasi yordamida xisoblash mumkin bulgandagina mavjud buladi. Bu tezis amalda aksioma, postulat bulib, prinsipial jixatdan matematik usullar bilan isbotlanishi mumkin emas. Uning kanchalik tugriligi fakat amaliy usullar bilan tekshirilishi mumkin. Ammo Tyuring tezisining inkor etilishi imkoniyati xam prinsipial mavjud bulib, buning uchun kandaydir algoritim bilan xisoblanuvchi, ammo xech kanday Tyuring mashinasi bilan xisoblanmaydigan funksiya kursatilishi kerak. Tyuring mashinasi va EXM lar. Tyuring mashinalarini urganish va ular uchun dasturlar tuzish algoritmik fikrlash fundamentini xosil kiladi. Uning mazmuni shundan iboratki, u yoki bu jarayonni elementar tashkil kiluvchi kadamlarga ajrata olishdir. Tyuring mashinasida xisoblash jarayonini kismalarga bulish (analiz) ning eng oxirgi imkoniyatlari xam amalda kullanganilgan. Bular: berilgan birlik informasiya simvollarini «ukish», elementar simvolda kuzatish ob'ektlarini uzgartirish; berilgan axborotlarni uzgartirishdan iborat. Albatta, xisoblash jarayonini bunday mayda bulaklash uni anchagina uzaytirishi tabiiy. Ammo shu bilan birga bunday elementar kadamlarga bulingan jarayonning mantiqiy strukturasini anchagina soddilashadi va nazariy tadqiqotlar uchun kulay shaklini oladi.	SHunday kilib, Tyuring mashinasi tushunchasi algoritmik jarayon analizining nazariy kuroli bulib, bundan esa ushbu tushunchaning algoritmik fikrlash moxiyati analizining nazariy kuroli sifatida maydonag keldi, deb xisoblash mumkin. Xozirgi zamon EXM larida algoritmik jarayon Tyuring mashinalaridagidek unchalik mayda tshkil etuvchilarga bulinmaydi. Aksincha, EXM yaratuvchilari mashina tomonidan bajariladigan amallarni imkon boricha kattalashtirishga intiladilar (bu yulda ma'lum chegaralarga rioya etiladi). Jumladan, Tyuring mashinasida «kushish» amali butun bir dasturni tashkil etadi., xozirgi zamon EXMlarida esa bu amal elementar xisoblanadi. Bundan tashkari Tyuring mashinasi cheksiz tashki xotiraga ega (ikki tomondan chegaralanmagan yacheykalarga bulingan lenta). Ammo biror real mavjud bulgan mashinadi cheksiz xotira bulishi mumkin emas. Tyuring mashinasi xozirgi zamon EXMlarining xotirasining cheksiz kattalashtirilib borilishi potentsial imkoniyatini aks ettiradi. Va nixoyat, Tyuring mashinalari bilan xozirgi zamon EXMlari ishining kiyosiy analizini utkazish mumkin. Kupchilik EXMlarda uch adresli buyruklar sistemasi kabul kilingan, bu xotiraning birdaniga uchta yacheykasi ichidagilari katmashuvchi binar amallarning bajarilishi bilan belgilanadi. Maslan, a yacheykadagi son v yacheykadagi songa kupaytiriladi, natija s yacheykaga junatiladi. Ikki adresli va bir adresli EXMlar xam mavjud.
Bir adresli EXMlar kuyidagicha ishlaydi: summatorga a yacheykadagi son chakiriladi; summatorada, masalan, v yacheykadagi songa ushbu son kupaytiriladi, natija summatoradan s yacheykaga uzatiladi. Tyuring mashinasini bir adresli mashina deb, xisoblash mumkin. Bu erda bir adresli buyruklar yanada soddalashtirilgan: mashina ishining xar bir kadamida kurilayotgan yacheykadagi birgina belgi uzgartiriladi, adres esa fakat 1 ga uzgarishi mumkin(chapdan yoki undan kushni yacheykaga utish) yoki umuman uzgarmasligi mumkin. Bu jarayonni anchagina chuzadi, ammo uni standartlashtiradi. Xulosa kilib shuni aytish mumkinki, xozirgi zamon EXMlari Tyuring mashinalarining real fizik modellari bulib, xisoblash jarayonlarining realizasiyasi uchun yaratilgandir. SHu bilan birga Tyuring mashinalari tushunchasi va bu mashinadar nazariyasi xozirgi zamon EXM larining nazariy fundamenti bulib xisoblanadi.	

6-MA'RUZAMASHG'ULOTI:

Dasturlash tillari va ularning klassifikatsiyasi. Mashinaga mo'ljallangan va proseduraga mo'ljallangan dasturlash tillari, yuqori darjali dasturlash tillari, interpretatorlar va kompilyatorlar, dasturlarni translyasiyalash.

MA'RUZA DARSINI OLIB BORISH TEXNOLOGIYASI

Talabalar soni:	50 – 60 talaba
Vaqt:	2 soat
Dars shakli	Mavzu bo'yicha ma'ruza
Ma'ruza rejasi	- Kompyuter dasturlari. - Dasturlash tili haqida tushuncha. Dasturlash tillari turlari.

	3. Dasturlash tillarining sintaktik jihatdan turlari. 4. Kompilyatsiya va interpretatsiya qilinuvchi tillar. 5. Komp'yuterda masala echish bosqichlari.
O'quv mashg'ulotining maqsadi: Talabalarda dasturlash tillari, ularning turlari, dastur kompilyatsiyasi va interpretatorlari tushunchalarini hosil qilish.	
Tayanch tushuncha va iboralar	Kompyuter dasturlari; amaliy dasturlar; tizimli dasturlar; uskunaviy tizimlar;Dasturlash; dasturchi; Dasturlash tillari turlari; assembler tili; fortran tili; beysik tili; quyi darajadagi DT; o'rta darajadagi DT; yuqori darajadagi DT;Kompilyatsiya va interpretatsiya qilinuvchi tillar; Komp'yuterda masala echish bosqichlari.
Pedagogik vazifalar:	O'quv faoliyati natijalari:
Kompyuter dasturlari haqida tushuncha beradi	Kompyuter dasturlarining 3 turi haqida tushunchaga ega bo'ladi
Dasturlash tili haqida tushuncha berish.	Dasturlash tillari va ularning turlari haqida bilimga ega bo'ladilar.
Dasturlash tillarining sintaktik jihatdan turlari.	Dasturlash tillari sintaktik jihatdan quyi, o'rta va yuqori darajaga bo'linish haqida ma'lumot oladilar.
Kompilyatsiyaqilinuvchi va interpretatsiya qilinuvchi tillar haqida ma'lumot berish.	Dasturlash tillari komputerda bajarilishiga qarab kompilyatsiya qilinuvchi va interpretatsiya qilinuvchi tillarga bo'linishi haqida ma'lumotga ega bo'ladilar.
Komp'yuterda masala yechish bosqichlarini tushuntirish.	Komp'yuterda masala yechish bosqichlari haqida bilimga ega bo'ladilar.
O'qitishvositalari	ma'ruza matni, tarqatma material, komp'yuter slaydlari, doska
O'qitishusullari	ma'ruza, tushuntirish, instruksiya berish, namoyish, Blits-so'rov, "Aqliy hujum", B/B/B (Bilaman/ Bilmoqchiman/ Bilib oldim) usuli
O'qitish shakllari	frontal, kollektiv ish
O'qitishsharoiti	komp'yuter, mul'timedia proektor (yoki proektor va LCD panel)

«Dasturlash tillari va ularning klassifikatsiyasi, mashinaga mo'ljallangan va protseduraga mo'ljallangan dasturlash tillari. Yuqori darjali dasturlash tillari, interpretatorlar va kompilyatorlar, dasturlarni translyatsiyalash»

MAVZUSIDAGI MA'RUZA DARSIGA TEXNOLOGIK KARTA

Ish bosqichlari	O'qituvchifaoliyatining mazmuni	Tinglovchi faoliyatining mazmuni
1-bosqich. Mavzuga kirish (10min)	1.1. O'quvmashg'uloti mavzusi, rejasiva maqsadi tushuntiriladi (1 - ilova)	Mavzu nomini yozib oladilar
	1.2. Blits-so'rov usulida mavzu bo'yicha ma'lum bo'lgan tushunchalarni faollashtiradi. (1-ilova)	Savollarga javob beradilar
2 -bosqich. Asosiy bo'lim (60 min)	2.1. Mavzu rejalari asosida tayanch tushunchalar bilan tanishtiradi.(2-ilova)	Mavzu rejasini yozib oladilar Tinglaydilar.
	2.2. Mavzuning 1- va 2- reja savollari bo'yicha "Aqliy hujum" uyushtiriladi. Har bir talaba fikri tinglanadi. Har bir savol muhokamasidan so'ng to'g'ri fikrlar tasdiqlanadi. Jarayon komp'yuter slaydlarini namoyish qilish bilan olib boriladi. (3-ilova)	Tinglaydilar. "Aqliy hujum" da ishtirok etadilar

3-bosqich. YAkunlovchi (10 min)	3.1. Mavzu bo'yicha yakunlovchi xulosalar qiladi. Mavzu bo'yicha olingan bilimlarni qacda ishlatish mumkinligi ma'lum qiladi.	Savollar beradilar
	3.2. Mavzu bo'yicha mustaqil o'rganish uchun topshiriqlar beradi.	Mustaqil o'rganish uchun topshiriqlarni yozib oladilar
	3.3. Mavzu bo'yicha bilimlarni chuqurlashtirish uchun adabiyotlar ro'yxatini beradi	Yozadilar
	3.4. Mavzuning keyingi savollari buyicha tayyorlanib kelish topshiriladi.	Yozadilar

Dasturlash tillari va ularning klassifikatsiyasi, mashinaga mo'ljallangan va protseduraga mo'ljallangan dasturlash tillari. Yuqori darjali dasturlash tillari, interpretatorlar va kompilyatorlar, dasturlarni translyatsiyalash.

Reja:

1-ilova

1. Kompyuter dasturlari.
2. Dasturlash tili haqida tushuncha. Dasturlash tillari turlari.
3. Dasturlash tillarining sintaktik jihatdan turlari.
4. Kompilyatsiya va interpretatsiya qilinuvchi tillar.
5. Komp'yuterda masala echish bosqichlari.

TAYANCH TUSHUNCHALAR:

2-ilova

Kompyuter dasturlari; amaliy dasturlar; tizimli dasturlar; uskunaviy tizimlar; Dasturlash; dasturchi; Dasturlash tillari turlari; assembler tili; fortran tili; beysik tili; quyi darajadagi DT; o'rta darajadagi DT; yuqori darajadagi DT; Kompilyatsiya va interpretatsiya qilinuvchi tillar; Komp'yuterda masala echish bosqichlari.

Tayanch tushunchalar:

Algoritm. Algoritmik til. Algoritm hossalari.
Chiziqli algoritm. Tarmoqlanuvchi algoritm.
Takrorlanuvchi algoritm.

Komp'yuter dasturlari uch turga bo'linadi.

Komp'yuter dasturlari- amaliy, tizimli, uskunaviy tizimlarga bo'linadi.

Amaliy dasturlar- bu foydalanuvchi bevosita ishlashi uchun mo'ljallangan dasturlardir.

Tizimli dasturlar- bu kompyuter qurilmalarining ishchi xolatini nazorat qiluvchi va boshqaruvchi dasturlar. Bundan tashqari u ko'p tizimli qobiqli dasturlar bo'lib, foydalanuvchini kompyuter bilan qulay muloqatini ta'minlaydi. Tizimli dasturning asosiy sinfi, bu drayver bo'lib, u OT xususan tashqi yoki ichki qurilmalar bilan ishlash imkonini beradi. Hozirgi kunda dasturlash tilining bir necha turlari mavjud:

Dasturlash tillari haqida

Tezkor elektron hisoblash mashinalarining paydo bo'lishi **dasturlash tili** deb ataluvchi turli-tuman belgilar sistemalarining paydo bo'lishiga olib keldi. Shunday qilib, hisoblash mashinalarida bajarilishi kerak bo'lgan jarayonlarni tavsiflash uchun qo'llaniladigan belgilar (simvollar) sistemasini **dasturlash tili** deb yuritamiz.

EHMLar uchun dastur tuzish va uni ishlatish juda murakkab va ko'p mehnat talab etadigan jarayon bo'lib, uning uchun ko'p aqliy mehnat va vaqt talab qilinadi. Shuning uchun yangi algoritmik tillarni yaratuvchilar dasturlashni sodda va xayotimizning turli sohalarida mehnat qiluvchilar uchun tushunarli bo'lishiga xarakat qilishadi. Hozirgi kunda dasturlash tillarining bir necha turi mavjud:

Bundan ko'rinadiki dasturlash tillarining soni bir necha yuzdan ortiq ekanligi.

Beysik dasturlash tili 1964 y. AQSh ning Dortmund kollejining ilmiy xodimlari Jon Kemeni va Toms Kurts tomonidan yaratilgan. Bu til turli hisoblashlarga doir masalalarni kompyuter bilan muloqat holda xal qilish uchun yaratildi.

Basic so'zi Beginners Allpurpose Symbolic Instruction Code –degan kengaytmasidan iborat bo'lib, ya'ni boshlovchilar uchun mo'ljallangan ko'p maqsadli, belgili ko'rsatmalar tili degan ma'noni bildiradi. Ushbu til kompyuter xotirasiga qo'yiladigan talablarning juda kamligi sababli, ShK larda ishlatiladigan til bo'lib qoldi. Bu tilning bir necha ko'rinishi mavjud bo'lib, maqsad foydalanuvchilarning muloqotida osonlik tug'dirish.

Assembler tili inglizchasiga Assembler- bu EHM uchun mashina kodlarida dastur yozish ishini yengillashtirish maqsadida 40 yillar ohiri 50 yillarning boshida yaratilgan.

Fortran tili –inglizcha Formula Translator – formulani translyatsiya qilish so'zlaridan qisqartirib olingan bo'lib, formula tarjimoni degan ma'noni bildiradi va bu til muhandislik, ilmiy-texnik masalalarni yechishga mo'ljallangan til hisoblanadi. Ushbu tilning qator variantlari yaratilgan, ulardan eng mashhurlari Fortran II va Fortran IV. Fortran tilining asosiy g'oyalari keyingi Algol-60, 68, PL-1 va boshqa tillarda rivojlantirilgan. Bu til 1954 yil yaratilgan.

Shunday maqsadni 1970 yili Shvesiyalik Oliy texnika o'quv yurtining professori Niklaus Virt o'z oldiga qo'ydi va u tomonidan tavsiya etilgan algoritmik tilni yaratdi va ulug' fransuz olimi B.Paskal (1623-1662) nomi bilan atadi.

Dasturlash tillarining sintaktik jihatdan turlari

Dasturlash tillarining sintaktik jihatdan turlari 3 turga(Informatika darsliklariga asosan) bo'linadi:



Quyi darajadagi dasturlash tili “Mashina tili” deb ham ataladi. Ushbu tilda dasturlar to'g'ridan-to'g'ri Operativ Xotira(OX) katakchalari va protsessor reyestrlari bilan ishlab tuziladi. Ushbu

tildagi buyruqlar Markaziy Protessor(MP)ning operatsiyalariga to'g'ri keladi. Buyruqlar ikkilik kodda yozilgan:

1: 10110000 01100001

2: (AL reyestriga 16-lik 61 sonini joylash)

Bir paytlar perfokartalar yordamida aynan mashina tilida dasturlar yozilgan.

O'rta daraja dasturlash tillarida protsessor buyruqlarini mnemonik kodlarga(buyruqqa mos qisqartirilgan so'zlar) almashtirilgan. Assembler tili bunga misoldir. O'rta darajadagi dasturlash tillarida ham bir protsessor operatsiya deyarli bir buyruqqa mos keladi. Masalan, yuqoridagi mashina kodi Assemblerda quyidagicha yoziladi:

1: MOV AL, 61h

2: (Yuqoridagidan ko'ra tushunarliroq)

Ko'rib turganingizdek, Assembler tili mashina kodidan bir pog'ona yuqorida turadi xolos.

ESDA TUTING!

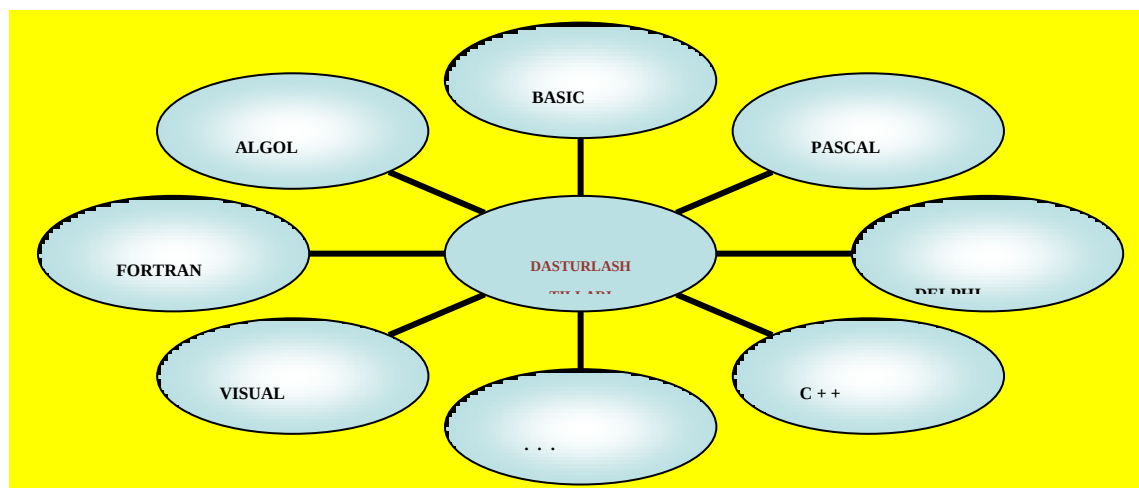
Keyingi ikki tilda tuzilgan programmalar ham har doim birinchi ko'rinishga(obyekt kodi) keltiriladi.

Yuqori darajadagi dasturlash tillari esa, asosan, dasturlash jarayonini tezlashtirish uchun yaratilgan. Shuni eslatib o'tish lozimki, har qanday dastur bajarilishidan oldin mashina kodiga o'tkaziladi. Ushbu darajadagi dasturlash tillarida yozilgan dastur ma'lum ma'noli

1: if (var == 6) {
2: print "Salom";

so'zlardan(odatda ingliz tilidagi) tashkil topadi. Masalan:

Ko'rib turibsizki, dastur qismi ingliz tilidagi ma'noli so'zlardan tashkil topgan. Hozirgi zamonaviy tillarning barchasi yuqori darajaga mansub.



Kompilyatsiya va interpretatsiya qilinuvchi tillar



Dasturlash tillari komputerda bajarilishiga qarab kompilyatsiya qilinuvchi va interpretatsiya qilinuvchi tillarga bo'linadi.

Kompilyatsiya qilinuvchi dasturlash tillarida dastur kodi kompilyator tomonidan mashina kodiga o'tkaziladi. Operatsion tizim(OT) esa, shu kodni to'g'ridan-to'g'ri ishlataveradi. Kompilyatsiya jarayoni komputer protsessori va OT talablariga mos ravishda amalga oshiriladi. Shuning uchun, bir OT uchun kompilyatsiya qilingan dasturning mashina kodi ikkinchi OT da ishlamaydi. Ushbu turdagi tillarga quyidagilarni misol qilib keltirishimiz mumkin: C, C++, Pascal va h.k.

Microsoft Windows OTlarida kompilyatsiya qilingan dastur nomi *.exe ko'rinishidagi fayl bo'ladi. Linux, Unix(va shularning davomchilari) kabi OT larda esa fayl kengaytmasining ahamiyati yo'q.

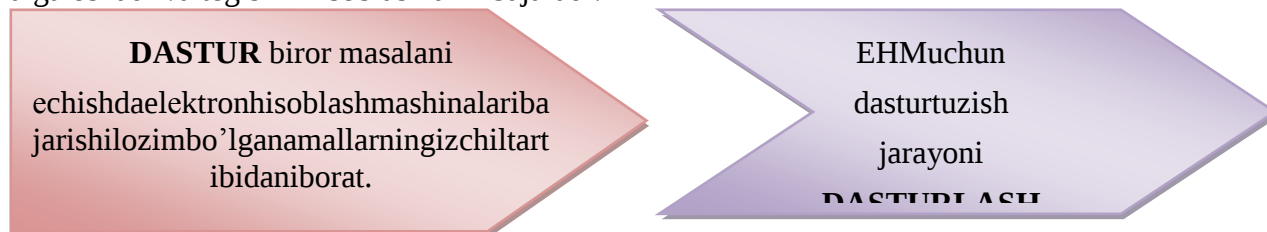
Kompilyatsiya qilinuvchi dasturlash tillarining asosiy yutuqlaridan biri — u OT dan boshqa biror dastur yoki kutubxona(Library, mas. DLL) o'rnatishni talab qilmaydi. Bundan tashqari, interpretatsiya qilinuvchi tillarga nisbatan ancha tez ishlaydi.

Interpretatsiya qilinuvchi dasturlash tillarida tuzilgan dastur kodi kompilyatsiya qilinmaydi. Ushbu turdagi dasturni ishlatishdan oldin dastur kodi interpretatsiya qilinadi. Interpretatsiya qilinuvchi dasturlash tillarida tuzilgan dastur mos interpretator o'rnatilgan komputerlardagina ishlaydi. Ushbu turdagi dasturlash tillariga PHP, Python, Ruby kabi tillar kiradi.

Interpretatsiya qilinuvchi dasturlash tillari kompilyatsiya qilinuvchilaridan, asosan, yozilgan dasturning deyarli hamma platformalarda ishlashi bilan ajralib turadi. Dastur biror turdagi OT yoki protsessor uchun yozilmaydi — faqat interpretatorgina turli platformalar uchun yoziladi.

Interpretatsiya qilinuvchi dastur kodi bajarilishidan oldin interpretator tomonidan oraliq kodga "kompilyatsiya" qilinadi. Shu oraliq kod interpretator tomonidan bajariladi. Python kabi tillar oraliq kodni saqlab qo'yadi, dastur kodi o'zgarmaguncha shu oraliq kodni ishlatadi.

Dastur biror masalani yechishda elektron hisoblash mashinalari bajarishi lozim bo'lgan amallarning izchil tartibidan iborat. EHM uchun dastur tuzish jarayoni dasturlash deyiladi. Dasturlash yechilishi kerak bo'lgan masala algoritmini EHM tiliga, ya'ni «mashina tili»ga o'tkazishdir. EHM uchun dastur tuzish – masalani yechish usulini mashina buyruqlarining shunday majmui (dasturi)ga, keltirish demakki, bu buyruqlar xotiraga joylashib, tartib bilan amalga oshadi va tegishli hisoblashlarni bajaradi.



KOMPYUTERDA MASALA YECHISH BOSQICHLARI (KMEB).

1. Masalaning qo'yilishi- bunda berilgan masalaning to'g'ri qo'yilganligi va uni yechish uchun qanday ma'lumotlar kerakligi va qanday natija olinishini bilish kerak.
2. Masalaning matematik modelini tuzish-bu bosqichda matematik formula tanlanadi matematik modeli tuziladi.
3. Masalani yechishni sonli usuli-bu bosqichda Hosil qilingan mat-k masalaning yechish usuli Tanlanadi va buning uchun sonli usuldan foydalaniladi.
4. Algoritmini tuzish-bunda mashina tushunadigan biror bir algoritmik tilda dG'tuzish.

Har bir til ham o'z alfavitiga ega.

Agar tuzilgan programmada alfavitda yo'q xarf yoki belgilar uchrasa, mashina bunday belgini tushunmaganligi haqida xabar beradi.

Biz ishlayotgan har qanday dastur biror -bir shaxs tomonida yozilgan bo'lib, ular dasturchilar deb aytiladi. Dasturchilar dasturni maxsus tillarda yozishadi –ularni dasturlash tillari deb aytiladi. Dastur ko'rinishlariga nisbatan quyi va yuqori dasturlash tillari sinflariga bo'linadi. Quyi dasturlash tillariga misollar “quyi darajadagi dasturlash” assambler tilini keltirishimiz mumkin. Yuqori dasturlash tillariga Paskal, TSi, SQQ, S# tillarini keltirishimiz mumkin.

BBB JADVALI

BILAMAN	BILMOQCHIMAN	BILIB OLDIM

BLITS – SO'ROV:3-ilova

1. Kompyuter dasturlari deganda nimani tushunasiz?
2. Dasturlash tillarining qanday turlarini bilasiz va ularning roli.
3. Dasturlash tillarining bir-biridan farqi?
4. Zamonaviy dasturlash tillari qaysilar va ularning xususiyatlari?
5. Dasturlarni kompilyatsiya qilish.
6. Dasturlarni interpretatsiya qilish.
7. OTlarida kompilyatsiya qilingan dastur nomi kengaytmasi qanday bo'ladi

7-MA'RUZA MASHG'ULOTI:

Pascal dasturlash tiliga kirish. Pascal tili sintaksisi. O'zgarmaslar va o'zgaruvchilar. Ma'lumotlarning standart turlari. Fodalanuvchining turlari. Pascal dasturlash tilida dasturning tarkibiy qismi. Tarkibiy qismlarning xarakteristikasi. Nishonlarni, o'zgarmaslarni, ma'lumotlarni, o'zgaruvchilarni, funktsiya va proseduralarni, opreatorlarni va izohlarni ifodalash bo'limi.

MA'RUZA DARSINI OLIB BORISH TEXNOLOGIYASI

Talabalar soni:	50 – 60 talaba
Vaqt:	2 soat
Dars shakli	Kirish ma'ruzasi
Ma'ruza rejasi	1. Paskal dasturlash tili. 2. Paskal dasturlash tili alifbosi. 3. Paskal dasturlash tilining asosiy tushunchalari.

	4. Paskal dasturlash tilida asosiy algoritmik konstruksiyalar va ularni dasturlash tilida amalga oshirish. – Chiziqli algoritm; – Tarmoqlanuvchi algoritm; – Takrorlanuvchi algoritm
Dars maqsadi : Paskal dasturlash tili haqida umumiy ma'lumotlar: alifbosi, miqdorlar, algoritmik konstruktorelar, asosiy tushunchalar bo'yicha talabalarda tushuncha xosil kilishga o'rgatish.	
Pedagog vazifalari: Paskal dasturlash tili haqida umumiy ma'lumotlar berish.	O'quv natijalari: Dasturlash tillari haqida ma'lumot beradi va Paskal tili, tarixi haqida izoh beradi.
Paskal dasturlash tili alifbosini kiritadi.	Paskal dasturlash tili alifbosiga nimalarni kiritish mumkin, nim alarni kiritish mumkin emasligini aniq ko'rsatib, sanab beradi.
Kattaliklar, o'zgaruvchi va o'zgarmaslar tushunchasini kiritish.	Miqdortushunchasini kiritib, ularning turlarga ajratadi. Xar bir turdagi miqdorning qabul qiluvchi qiymatlarini turlarga ajratib, ularni o'zaro farqlab ko'rsatadi. Paskalda ulardan foydalanish qoidalarini tushuntiradi.
Paskalda o'zgaruvchi tiplari, ularning nomlanishini tushuntirish.	Paskalda o'zgaruvchilarni nomlanishi, ularning tiplari, qiymatlarni qabul qilish doiralari sxematik tushuntiriladi.
Paskal dasturlash tilida asosiy algoritmik konstruksiyalar haqida ma'lumotlar berish.	Algoritm tushunchasi, uning chiziqli, tarmoqlanuvchi, tarmoqlanuvchi turlariga mos konstruksiyalar tushuntiriladi, ularga mos dasturlarlar tuzish uchun Paskal dasturlash tilining operatorlari bilan tanishtiradi.
O'qitishning usul, texnikasi:	Ma'ruza, suxbat, baxs-munozara muhokama qilish, fikr almashish.
O'qitish vositalari:	Ma'ruza matni, tarqatma materiallar, slaydlar
O'qitish shakli:	Frontal
O'qitish shart – sharoitlari:	Texnik vositalarni qo'llashga mo'ljallangan auditoriya. Proektor
Monitoring va baholash:	Og'zaki so'roq, savol-javob, tahlil natijalari.

MA'RUZA DARSIGA TEXNOLOGIK KARTA

Faoliyat bosqichlari	Faoliyat mazmuni	
	Pedagog	Talaba
1-bosqich. Kirish (5 daqiqa)	1.1. Ma'ruzaning mavzusini e'lon qiladi, o'quv mashg'ulotining maqsadi va natijalarini tushuntiradi. 1.2. Ma'ruza mashg'ulotining mavzusiga ma'ruza rejasini e'lon qiladi (1- ilova).	Tinglaydi, mavzu nomini yozib oladilar
2-bosqich. Asosiy bosqich (55 daqiqa)	2.1 Paskal dasturlash tili alifbosini kiritadi (2 - ilova). 2.2. Kattaliklar, o'zgaruvchi va o'zgarmaslar tushunchasini kiritadi. (3 - ilova) 2.3. Paskalda o'zgaruvchi tiplari, ularning nomlanishini tushuntiradi. (4 - ilova) 2.4. Paskal dasturlash tilining umumiy strukturasi va undagi xizmatchi so'zlar bilan. (5 - ilova) 2.5. Talabalarga savollar bilan murojaat etadi. (6-ilova). 2.6. FSMU texnologiyasidan foydalanib "Nima uchun	Eshitadilar kerakli ma'lumotlarni yozib, sxemalarni chizib oladilar. Savollarga javob beradilar. Juftli guruxlarga

	dastur tuzishni har birimiz bilishimiz shart?” savolini	bo’linib FSMU
		advalini
		o’ldiradi
3-bosqich		Eshtadilar
YAkuniy.	Talabalarning dars davomidagi faoliyatini tahlil etadi va	Musta hlash
(10 daqica)	baholaydi. Mustaqil ishlash uchun vazifa beradi (8 – ilova)	uchun
		lar

Paskal tili sintaksisi.Kattaliklarning asosiy turlari.Paskal tilidagi programmalarining strukturasi.Paskal tilining operatorlari.qiyimatlarning skalyar turlari: tartib-langan, sanab o’tiluvchi va chegaralangan turlar.

1-ilova

1. Paskal dasturlash tili.
2. Paskal dasturlash tili alifbosi.
3. Paskal dasturlash tilining asosiy tushunchalari.
4. Paskal dasturlash tilida asosiy algoritmik konstruksiyalar va ularni dasturlash tilida amalga oshirish.
 - a. Chiziqli algoritm;
 - b. Tarmoqlanuvchi algoritm;
 - c. Takrorlanuvchi algoritm.

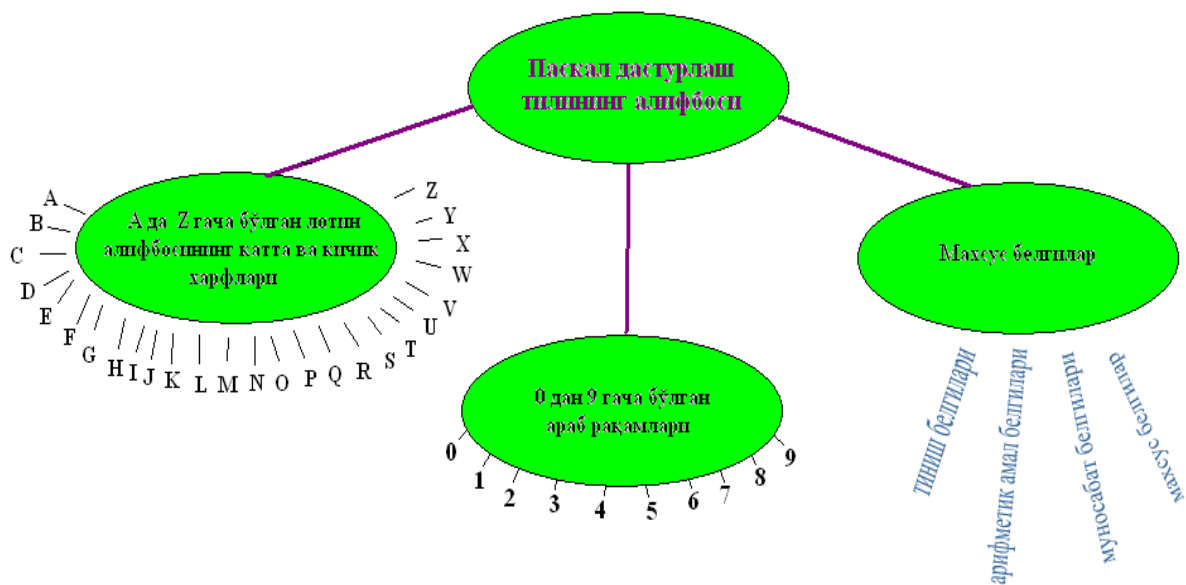
Faollastiruvchisavollar:

2. *Algoritm deb nimaga aytiladi?*
3. *Algoritmning qanday turlari mavjud?*
4. *Blok – sxema elementlarini sanab bering?*
5. *Argumentlarni kiritish operatorini tushuntirib bering*

Dasturlash tillari. EHM lar uchun dastur tuzish va uni ishlatish juda murakkab va ko’p mehnat talab etadigan jarayon bo’lib, uning uchun ko’p aqliy mexnat va vaqt talab qilinadi. Shuning uchun yangi algoritmik tillarni yaratuvchilar dasturlashni sodda va xayotimizning turli sohalarida mexnat qiluvchilar uchun tushunarli bo’lishiga xarakat qilishadi.

Bundan ko’rinadiki dasturlash tillarining soni bir necha yuzdan ortiq ekanligi. 1970 yili Shvesiyalik Oliy texnika o’quv yurtining professori Niklaus Virt o’z oldiga qo’ydi va u tomonidan tavsiya etilgan algoritmik tilni yaratdi va ulug’ fransuz olimi B.Paskal (1623-1662) nomi bilan atadi.

Paskal dasturlash tilining alifbosi



2-Ilova

Pascal tili ham boshqa dasturlash tillari kabi o'z alfavitiga va belgilariga ega.

U 26 bosh lotin harflarini, 0 dan 9 gacha bo'lgan arab raqamlarini va quyidagi belgilarni ishlatadi: bo'shliq belgisi; 4 ta arifmetik amallar + , - , * , / ; mantiqiy amallarni bajarish uchun < , > , <= , >= , <> , = belgilarini ishlatadi. Bulardan tashqari vergul, nuqta, ikki nuqta, kichik qavs, katta va o'rta qavs.

Tuzilgan programmada alfavitda yo'q xarf yoki belgilar uchrasa, mashina bunday belgini tushunmaganligi xaqida xabar beradi.

Paskal tilining asosiy tushunchalari.

Paskal tili ham boshqa tillar kabi o'zining grammatikasi, qonun va qoidalariga ega bo'lib, ular yordamida yuqorida keltirilgan alfavit belgilaridan nomlar va gap tuziladi. Paskalda nomlar ikki xil bo'ladi:

- 1) tildagi standart nomlar;
- 2) dastur tuzuvchining nomlari.

Standart nomlar ingliz tilida yoziladi va malum manoni anglatadi. Dastur tuzilayotganda bu nomlar ingliz tilidagi ko'rinishda yoziladi va faqat o'z manosida ishlatiladi. Shuning uchun ham bu nomlar standart nomlar deyiladi (1-jadval.). Dastur tuzuvchilarning nomlari Paskaldagi simvollardan malum qoida yoki qonun asosida tuziladi, ular hisoblash jarayonida o'zgarmaslar hamda o'zgaruvchi qiymatlarni va hokazolarni ifodalash uchun ishlatiladi. Shuni eslatib o'tish kerakki, dastur tuzuvchilarning nomlari sifatida standart nomlar va xizmatchi so'zlardan foydalanish mumkin emas.

Tarix va ma'lumot

Pascal tili Shvetsariyalik olim N.Virt tomonidan yaratilib, keyinchalik Borland korporatsiyasi tomonidan rivojlantirildi. Bu til rivojlantirilib Turbo Pascal, Borland Pascal va keyinchalik esa Object Pascal nomini oldi. Hozirgi kunda Object Pascal tili asosi bo'lgan Windows muhitida ishlovchi Delphi dasturiy vositasida murakkab professional dasturlar ishlab chiqilmoqda.

Kattalik – qiymatning tipini, miqdorini ifodalovchi ko'rsatkichdir. U EHM da saqlanishga va ishlatishga qarab ikkiga bo'linadi: o'zgarmaslar va o'zgaruvchilar.

O'zgarmas kattaliklar – dasturning bajarilish jarayonida qiymatlari saqlanuvchi kattaliklar bo'lib, ular o'z navbatida sonli, satriy va mantiqiy tiplarga bo'linadi. Sonli kattaliklar butun, xaqiqiy va eksponensial ko'rinishda berilishlari mumkin.

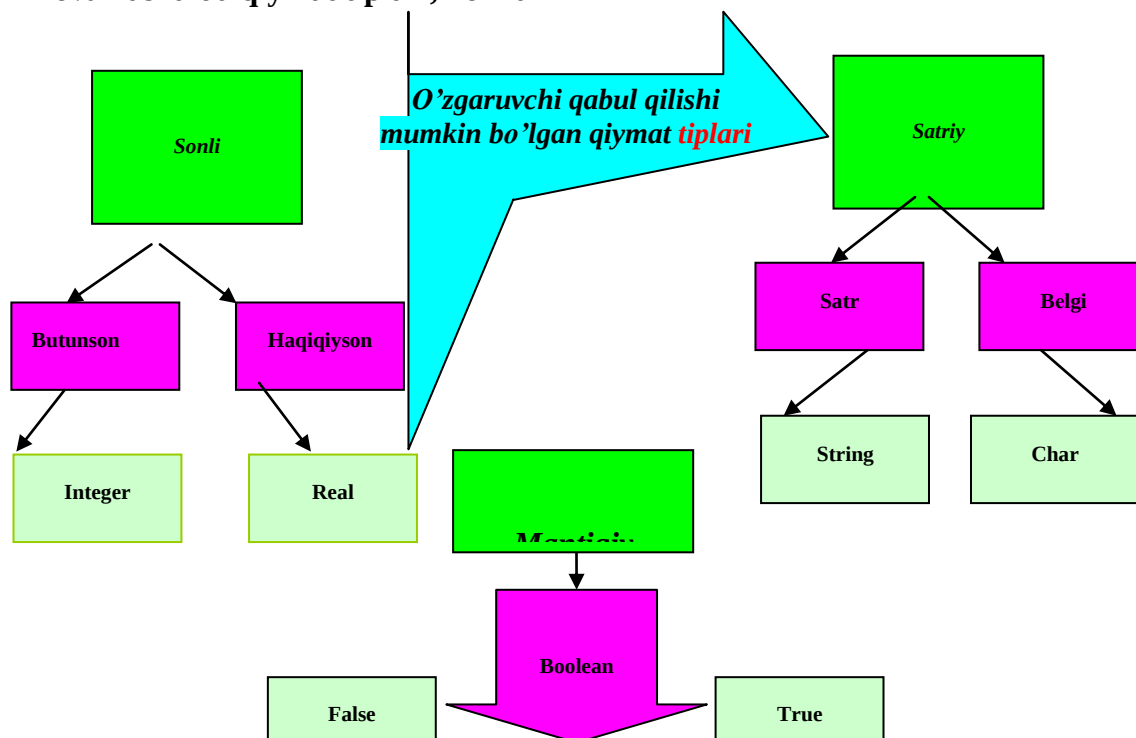
3-ilova.

Kattalik – qiymatning tipini, miqdorini ifodalovchi ko'rsatkichdir. U dasturda qo'llanilishiga ko'ra ikkiga bo'linadi

O'zgarmas kattaliklar – dasturning bajarilish jarayonida qiymatlari saqlanuvchi kattaliklar

O'zgaruvchilar - dastur bajarilishi jarayonida qiymatlari o'zgaradigan kattaliklar

4-ilova Paskalda qiymat tiplari, nomlari



Ma'lumki, ma'lumotlarning tahlili jarayonini ifodalovchi algoritm turli xil ob'ektlar (o'zgarmaslar, o'zgaruvchi miqdorlar, funksiyalar va hokazo) ustida ish olib boradi. Bu ob'ektlarga ularning vazifasi va qabul qiladigan qiymatlariga qarab maxsus nomlar beriladi. Shu nomlarni odatda, identifikatorlar deb ataladi. Identifikator deb harf yoki "_" belgisidan boshlanuvchi, harf, raqam va "_" belgisining ixtiyoriy ketma-ketligiga aytiladi.

Xizmatchi so'zlardan identifikator sifatida foydalanish mumkin emas. Odatda identifikator so'zining o'rniga qulayroq va qisqaroq qilib nom deyish mumkin. Dasturda qatnashuvchi ob'ektlarga nomlarni dastur tuzuvchi o'z ixtiyoriga ko'ra tanlab olishi mumkin. Bir xil nom bilan bir necha xil ob'ektlarni nomlash mutlaqo mumkin emas. Turbo Paskal muhitida nomda qatnashuvchi belgilar soni (nom uzunligi) 63 ta belgidan oshmasligi kerak.

Standart funksiyalar

Ba'zi matematik standart funksiyalarning Paskal tilida ifodalanishi quyida keltirilgan:

Ctandard funksiya	Pascal da yozilishi	Izoh
$ x $	abs(x)	Argumentning fasalyut qiymati
$arctg x$	arctan(x)	x ning arktangensi (rad)
$\cos x$	cos(x)	x ning kosinusi
$\sin x$	sin(x)	x ning sinusi
e^x	exp(x)	eksponensial funksiya
$\ln x$	ln(x)	natural logarifm ($x>0$)
$\sqrt{x}$	sqrt(x)	kvadrat ildiz ($x \geq 0$)
$[x]$	int(x)	x ning butunqismi
x^2	sqr (x)	x ning kvadrati

Paskal tilida identifikator tushunchasi majud bo'lib, dasturda ob'ektlarni nomlashda ishlatiladi. O'zgarmaslarni, o'zgaruvchilarni, belgi(metka), protsedura va funksiyalarni belgilashda ishlatilgan nom **identifikatorlar** deyiladi. Identifikatorlar lotin alfaviti harflaridan boshlanib qolgan harflari belgi yoki raqam ketma-ketligidan tashkil topgan bo'lishi mumkin. Masalan: xx, xx1, alfa&.

Komp'yuter foydalanuvchi tomonidan qo'yilgan masalani aniq va tushunarli ko'rsatmalar berilgandagina bajara oladi. Bu ko'rsatmalar ma'lum bir ma'noni anglatuvchi so'zlardan iborat bo'lib, komp'yuterga qanday operatsiyani bajarish lozimligini bildiradi va bu ko'rsatmalarga **operatorlar** deyiladi. Operatorlar dastur ishlaganda ketma-ket ravishda bajariladi. Paskal tilida bir satrga bir necha operatorlarni yozish mumkin.

Paskal tilida dastur matni bosh va asosiy bo'limdan tashkil topadi. Bosh bo'lim dastur nomi va o'zgaruvchilar, o'zgarmaslar, massivlar, belgilar(metkalar), protseduralar va funksiyalarni tavsiflashdan iborat bo'ladi. Asosiy bo'lim dastur tanasi deyilib, unda dasturda bajariladigan hamma operatorlar ketma-ketligi beriladi va u Begin (boshlamoq) so'zi bilan boshlanib End (tugash) so'zi bilan tugaydi. Umumiy holda dastur strukturasi quyidagi ko'rinishga ega:

Paskal dasturining umumiy strukturasi

Program <Dastur nomi>:

Uses <Foydalanadigan bibliotekalar (modullar) ro'yxati>;

Label<Ishlatiladigan belgilar(metkalar) ro'yxati>;

Const<Ishlatiladigan o'zgarmlarni aniqlash>;

Type<Yangi turlarni aniqlash>;

Begin

<Bajariladigan operatorlar ketma ketligi>

5 – ilova

Masalani EHMda yechish bosqichlari



Paskal dasturining sodda strukturasi

Program<dastur nomi>:

Tasnif qismi;

(bu qismda doimiylarni, nishonlarni, o'zgaruvchilarni, qism dasturlarni e'lon qilish);

begin

bajariluvchi operatorlar ketma-ketligi;

5 – ilova

Istalgan hayotiy yoki matematik, fizik va hokazo masala shartlarini ifoda qilish dastlabki ma'lumotlar va fikrlarni tasvirlashdan boshlanadi va ular qat'iy ta'riflangan matematik yoki fizik va hokazo tushunchalar tilida bayon qilinadi. So'ngra yechishning maqsadi, ya'ni masalani yechish natijasida ayni nimani yoki nimalarni aniqlash zarurligi ko'rsatiladi.

Masala shartining aniq ifodasi masalaning matematik (fizik va hokazo) qo'yilishi deb ham ataladi va istalgan masalani yechish eng avval uning qo'yilishidan boshlanadi. Masalani qo'yilishida boshlang'ich ma'lumotlar yoki argumentlar hamda qiymatlari aniqlanishi kerak bo'lgan kattaliklar, ya'ni natijalar ajratiladi. Masalani qo'yish uniEHMda yechishning **birinchi bosqichi** bo'ladi.

Masalani qo'yish uchun avval tekshirilayotgan ob'ektni matematik atamalarda tavsiflash, ya'ni iloji bo'lsa uning matematik modelini qurish lozim, qurilgan ifoda haqiqiy ob'ektni tekshirishni

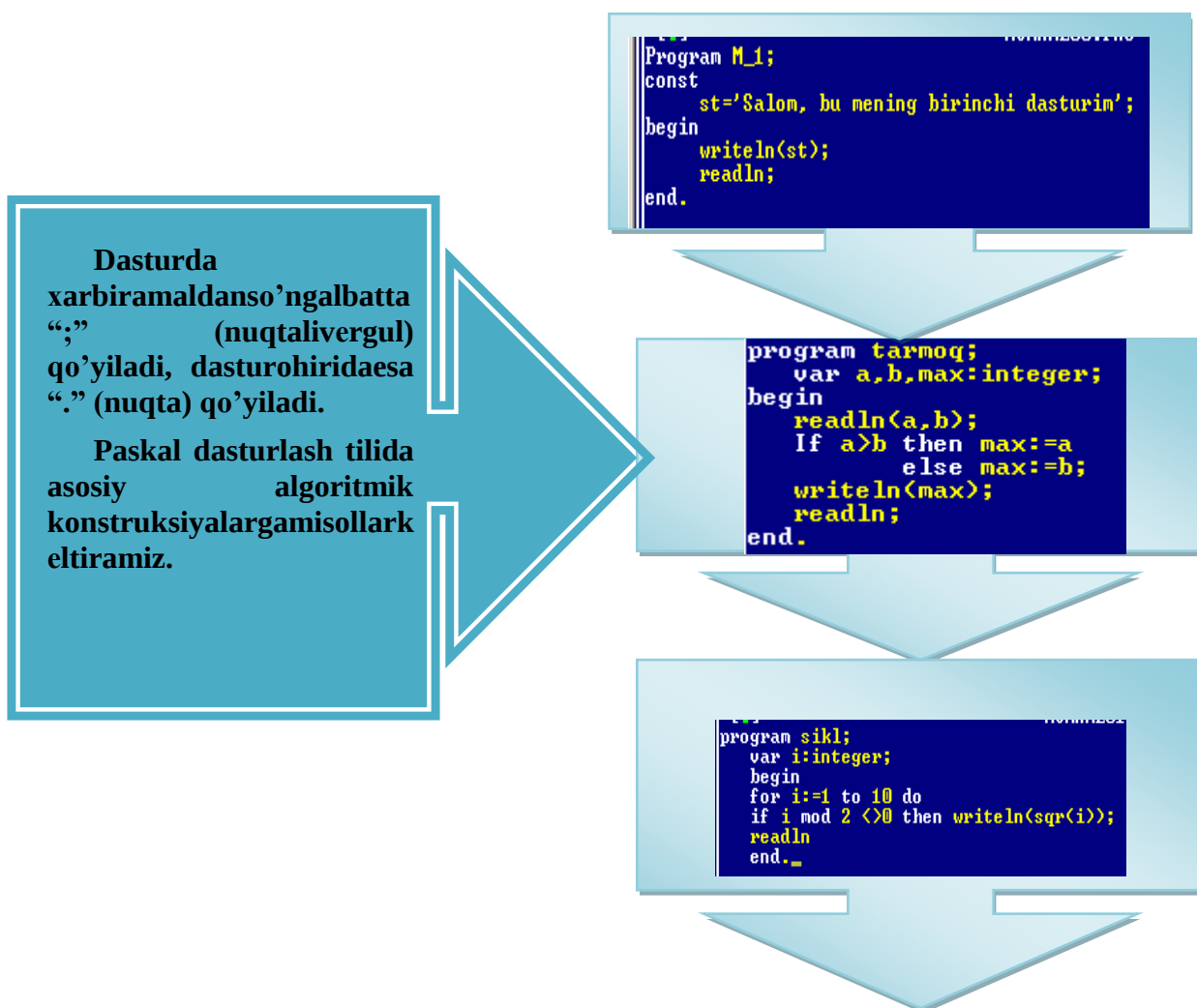
matematik masalani yechishga keltirish mikonini beradi. Bu bosqich masalani EHMda yechishning **ikkinci bosqichi** bo'ladi.

Navbatdagi bosqichda, ya'ni **uchinchi bosqichda**, masalani EHM dan foydalanib yechish uchun uning algoritmi tuziladi. Algoritmni turli – tuman ko'rinishda yozish mumkin. Informatika kursining asosiy vazifalaridan biri ham algoritm tuzish usullarini o'rganishdan iboratdir.

Algoritmning EHM da bajarilishi uchun bu algoritm biror dasturlash tilida yozilgan bo'lishi lozim. Masalani yechishning bu bosqichi **to'rtinchi bosqich** bo'lib, unda biron bir usulda yozilgan algoritm ma'lum bir dasturlash tiliga ko'chiriladi. Masalan, agar algoritm blok – sxema ko'rinishda tasvirlangan bo'lsa, uni Paskal dasturlash tiliga ko'chirish uchun har bir blokni tilning mos buyruqlar bilan almashtirish yetarli.

Bye sh in ch i b o s q i ch – dastur ko'rinishda yozilgan algoritmni EHM yordamida bajarish. Bu bosqichda natija olinib, olingan natijalarni tahlil qilish bilan tugallanadi.

Shunday qilib, masalalarni EHM da yechish bosqichlari bilan tanishib chiqdik. Shuni ta'kidlash kerakki, har doim ham bosqichlar bir – biridan yaqqol ajralgan holda bo'lmasdan, bir-biriga qo'shilib ketgan bo'lishi ham mumkin.



Faollashtiruvchisavollar:

1. Paskaldasturlau tiliningalifbosiganimalarkiradi?
2. Kattalikdebnimagaaytiladi?
3. Qandayqiymattiqlarinibilasiz?
4. Sonlitipniaytibbering.

6 – ilova

7-Ilova O'quv topshiriq:

FSMU texnologiyasiqoidasi.

Ushbu texnologiya munozarali masalalarni xal etishda xamda o'quv jarayonini baxs-munozarali o'tkazishda qo'llaniladi, chunki bu texnologiya talabalarni o'z fikrini ximoya qilishga, erkin fikrlash va o'z fikrini boshqalarga o'tkazishga, ochiq xolda baxslashishga xamda shu bilan birga baxslashish madaniyatini o'rgatadi. Tinglovchilarga tarqatilgan oddiy qog'ozga o'z fikrlarini aniq va qisqa xolatda ifoda etib, tasdiqlovchi dalillar yoki inkor etuvchi fikrlarni bayon etishga yordam beradi.

F – fikringizni bayon eting; S – fikringiz bayoniga sabab ko'rsating

FSMU texnologiyasi yordamida “Nima uchun dastur tuzishni har birimiz bilishimiz shart?”savolini yo'naltiruvchi uslubiy tavsiyalar asosida yechish:

Savol	Nima uchun dastur tuzishni har birimiz bilishimiz shart?
(F)Fikringizni bayon eting	
(S)Fikringiz bayoniga sabab ko'rsating	
(M)Ko'rsatgan sababingizni isbotlovchi dalil keltiring	
(U)Fikringizni umumlashtiring	

B/BX/B JADVALI

(8 - ilova)

Bilaman	Bilishni xohlayman	Bilib oldim

Nazorat ishlari

1. Masalani EHMda yechish bosqichlarini ketma-ketligini o'rnatig.

1. Komp'yuterga kiritish va natija olish
2. Algoritmini tuzish
3. Masalaning qo'yilishi
4. Dastur tuzish
5. Matematik modelini tuzish

2. Har bir bosqichiga izox bering.

№	Bosqich nomi	Izox
1	Masalaning qo'yilishi	
2	Matematik modelini tuzish	

3	Algoritmini tuzish	
4	Dastur tuzish	
5	Komp'yuterga kiritish va natija olish	

Mustaqil ish

1. “Paskalda standart funksiyalar” ga klaster tuzing.
2. Paskal dasturlash tilining operatorlaridan foydalanishga doir masalalar topish va dasturini tuzish.

№	Algoritm turi	Topshiriq
1.	Chiziqli algoritmgaga doir	2 ta masala
2.	Tarmoqlanuvchi algoritmgaga doir	2 ta masala
3.	Takrorlanuvchi algoritmgaga doir	2 ta masala

3. Uchta turdagi algoritmlarga mos «VENN» diagrammasini yaratish

8-MA'RUZA MASHG'ULOTI: PASCAL DASTURLASH TILIDA IFODALAR, OPERASIYALAR VA SODDA OPERATORLAR

Ma'ruza rejasi:

1. O'zlashtirish va ma'lumotlarni ekranga chiqarish operatori.
2. Ma'lumotlarni xotiraga muloqot usulida kiritish operatori.
3. Sodda dasturlar tuzish.

Paskal dasturlash tili, odatda, tavsiflangan o'zgaruvchilar uchun xotiradan joy ajratib, ularning turiga mos boshlang'ich qiymatlarni yozib qo'yadi.

O'zgaruvchi turi	Boshlang'ich qiymat	O'zgaruvchi turi	Boshlang'ich qiymat
Barcha butun sonli	0	Barcha haqiqiy sonli	0.0000000000e+00
char	' ' (probel)	boolean	FALSE
string	"" (bo'sh satr)	string[7]	"" (bo'sh satr)

O'zlashtirish operatori. O'zlashtirish operatori o'zgaruvchilarga qiymat berish uchun qo'llaniladi. U **:=** belgi orqali ifodalanadi. O'zlashtirish operatorining umumiy ko'rinishi quyidagicha:

o'zgaruvchi := ifoda;

Bu operator bajarilganda quyidagi ishlar bajariladi:

- 1) ifoda qiymati hisoblanadi;
- 2) ifodaning qiymati o'zgaruvchiga o'zlashtiriladi, ya'ni xotiraning o'zgaruvchi uchun ajratilgan qismiga (o'zgaruvchining «eski» qiymati o'chib ketadi) ifodaning qiymati yoziladi.

1- misol. Quyidagi dastur bajarilishi natijasida **a** nomli o'zgaruvchining qiymati **22** soniga teng bo'ladi.

2- misol. Quyidagi dastur bajarilishi natijasida **meva** nomli satrli o'zgaruvchining qiymati «olma» so'ziga teng bo'ladi.

```
var a: integer;
begin
  a := 22;
end.
```

```
var meva : string;
begin
  meva := 'olma';
end.
```

3- misol. Bu misolda **a** va **b** o'zgaruvchilarning qiymati qanday o'zgarishi yaqqol ko'rinadi.

```
var a,b,m: integer;
begin
  a := 8;      {a ning qiymati 8 ga teng bo'ldi}
  b := a*5;    {b ning qiymati a*5=8*5= 40 ga teng bo'ldi}
  b := b+10;   {endi b ning qiymati b+10= 40+10= 50 ga teng bo'ladi}
  m:=m*b;      {m ning boshlang'ich qiymati berilmagani uchun 0
               deb olinadi, demak, m ning qiymati 0*50=0 ga teng
               bo'ladi}
end.
```

Yuqoridagi misollarda o'zgaruvchilar turli qiymatlarni o'zlashtirdi. Ammo biz ularning natijasini ko'rmadik. Chunki ular kompyuter xotirasida qolib, ekranga chiqarilmaydi. Ma'lumotlarni kompyuter ekraniga chiqarish uchun **chiqarish operatoridan** foydalaniladi. Paskalda chiqarish operatori (protsedura) quyidagi ikki xil ko'rinishga ega:

Write(chiqarish ro'yxati) va **WriteLn(chiqarish ro'yxati)**

bu yerda **Write** (ing. – yozmoq) va **WriteLn** Paskalning xizmatchi so'zlari; chiqarish ro'yxati – o'zaro vergul bilan ajratilgan va ekranga chiqarilishi kerak bo'lgan ifoda, o'zgaruvchi yoki o'zgarmaslar ketma-ketligi. Chiqarish ro'yxatida ifoda qatnashsa, avval ifoda hisoblanib, hosil bo'lgan natija ekranga chiqariladi. Chiqarish ro'yxatidagi o'zgarmaslar belgili yoki satrli bo'lsa, albatta apostrof ichiga olinishi shart.

Write va **WriteLn** operatorlarining farqi shundaki, **Write** operatori yordamida ma'lumotlar ekranga chiqarilgach yurgich ekraning shu satrida qoladi va ekranga chiqariladigan keyingi ma'lumotlar shu satrga yurgich turgan joydan boshlab chiqariladi. **WriteLn** operatorida esa ma'lumotlar ekranga chiqarilgach yurgich keyingi satr boshiga o'tadi.

4- misol.
begin
 write('Yashna, '); write('gulla ');
 write('ona Vatanim!');
end.

Dastur bajarilgach, kompyuter ekranida

Yashna, gulla ona Vatanim!
yozuvi hosil bo'ladi.

5- misol.
begin
 writeln('Yashna, ');
 writeln('gulla ');
 write('ona Vatanim!');
end.

Dastur bajarilgach, kompyuter ekranida

**Yashna,
gulla
ona Vatanim!**
yozuvi hosil bo'ladi.

6- misol.
program o'zlashtirish_chiqarish;
var a,b:integer;
begin a:=23; b:=a+21;
 write('b ning qiymati ', b, ' ga teng');
end.

Dastur bajarilgach, kompyuter ekranida

b ning qiymati 44 ga teng
yozuvi hosil bo'ladi.

Ma'lumotlarni chiqarishda **chiqarish formatini** ko'rsatish mumkin. Chiqarish formati chiqarilayotgan ma'lumotlarning ko'rinishi (formati)ni belgilaydi. Buning uchun chiqarilayotgan o'zgaruvchidan keyin «:» (ikki nuqta) belgisi qo'yiladi. Masalan, a – haqiqiy turdagi o'zgaruvchi bo'lsa, chiqarish formatida ikkita parametr – chiqarilayotgan songa ajratilgan xonalar soni ko'rsatiladi. Masalan, **WriteLn(a:10:2);** operatori a ning qiymatini chiqarish uchun 10 ta xona ajratadi, shundan bitta xonasi nuqta va ikkita xonasi kasr qismi uchun ajratiladi. Agar son butun bo'lsa, chiqarish formatida bitta parametr – chiqarilayotgan songa ajratilgan xonalar soni ko'rsatiladi. Masalan, **WriteLn(b:6);**. Satrli va belgili o'zgaruvchilar uchun chiqarish formati ularning qiymatini chiqarish uchun ajratilgan joy (maydon) uzunligini aniqlaydi.

Chiqarilayotgan son yoki matn unga ajratilgan joyning o'ng chegarasi bo'yicha tekislanib chiqadi. Masalan, $a = 3.24$; bo'lsa, **WriteLn('a = ',a:6:2);** operatori ekranga $a = 3.24$ ko'rinishdagi yozuvni chiqaradi (= belgisidan keyin ikkita bo'sh joy (probel) qoladi).

Chiqarish formatida ajratilgan joy o'zgaruvchi qiymatining «uzunligi»dan kichik bo'lsa, chiqarish formati bekor qilinadi va o'zgaruvchining qiymati to'liqligicha ekranga chiqariladi. Faqatgina haqiqiy son kasr qismining formati bekor qilinmaydi. Haqiqiy sonni chiqarishda formati ko'rsatilmasa, u ekranga eksponensial ko'rinishda chiqariladi.

7- misol.

```
var a,b : real;
Begin    a:=3.24; b:=5;
        writeln('a=',a); writeln('b=',b);
end.
```

Kompyuter ekranida

```
a=3.2400000000E+00
b=5.0000000000E+00
```

8- misol.

```
var a,b : real;
Begin    a:=3.24; b:=5.3;
        writeln('a=', a:6:2);
        writeln('b=',b:1:0);
end.
```

Kompyuter ekranida

```
a= 3.24
b=5
```

Keltirilgan ikkala misolda ekranga chiqarilgan a va b o'zgaruvchilarning qiymatlari bir xil, ammo ularning ko'rinishida katta farq bor. 8-misoldagi ekranga chiqarilgan axborot albatta 7-misoldagiga nisbatan aniq va tushunarli ko'rinishga ega. Sonning kasr qismidagi kerakli raqamlar sonini aniq bil-magan hollarda ekranga noto'g'ri natija chiqarmaslik uchun chiqarish formatidan ehtiyotkorlik bilan foydalanish zarur.

Yodda tuting: ekrandagi natijani ko'rish uchun **ALT+F5** klavishlar juftligi bosiladi.

Paskalda o'zgaruvchilarga qiymat berishning o'zlashtirish operatoridan foydalanishdan boshqa usullari ham bor. Ulardan biri **ma'lumotlarni xotiraga muloqot usulida kiritish** deyiladi va **kiritish operatori** yordamida amalga oshiriladi. Kiritish operatoridan o'zgaruvchilarga dastur bajarilishi davomida kompyuter klaviaturasidan qiymat berish uchun qo'llaniladi. Kiritish operatori quyidagi ikki xil ko'rinishga ega:

Read (kiritish ro'yxati); va ReadLn(kiritish ro'yxati);

bu yerda **Read** (read (ing.) — o'qish) va **ReadLn** Paskalning xizmatchi so'zlari, **kiritish ro'yxati** esa bitta o'zgaruvchi yoki o'zaro vergul bilan ajratilgan bir nechta o'zgaruvchilar ketma-ketligi. Masalan: **Read(a); Read(alfa,betta); ReadLn(_name);**.

Kiritish operatori dastur ishlashini to'xtatadi va ro'yxatdagi o'zgaruvchilarga klaviatura orqali qiymat berilishini kutadi. Agar ro'yxatda bir nechta o'zgaruvchi bo'lsa, ularning qiymatlari o'zaro probel (bo'sh joy) bilan ajratib kiritilishi yoki **ENTER** klavishini bosib kiritilishi mumkin. Ikkala holda ham oxirgi o'zgaruvchi qiymati kiritilgach **ENTER** klavishini bosish shart.

Read va **ReadLn** operatorlarining farqlari quyidagicha. Bitta yoki bir nechta alohida yozilgan **Read** operatori yordamida kiritiladigan o'zgaruvchilar qiymati bir satrda probellar bilan ajratib kiritilishi mumkin. **ReadLn** operatori esa faqat o'zining ro'yxatida keltirilgan o'zgaruvchilarnigina qiymatini bir satrda probellar bilan ajratib kiritilishiga imkon beradi. Shuning uchun **ReadLn** operatoridagi ro'yxat tugagach keyingi kiritish operatorini ishlashi uchun albatta **ENTER** klavishini bosish shart.

1- misol.

```
Var a,b:Integer;
Begin
Read(a);
Read(b);
WriteLn('a+b=', a+b);
End.
```

2- misol.

```
Var
a,b:Integer;
Begin
Read(a, b);
WriteLn('a+b=', a+b);
End.
```

Ikkala misolda ham qiymat kiritish quyidagi ikki usuldan birida amalga oshirilishi mumkin.

1- usul: dastur ishga tushirilgach, ekranda satr boshida yurgich chiqadi va dastur **a** ning qiymati kiritilishini kutib turadi. Masalan, **a** ning qiymati sifatida 10 kiritiladi, so'ng probelni bosib **b** ning qiymati sifatida 11 deb kiritiladi. Endi **ENTER** klavishi bosilsa, ekranda quyidagilar aks etadi:

```
10 11
a+b=21
```

2- usul: dastur ishga tushirilgach, ekranda satr boshida yurgich chiqadi va dastur **a** ning qiymati kiritilishini kutib turadi. Masalan, **a** ning qiymati sifatida 10 kiritiladi, so'ng **ENTER** klavishini bosib **b** ning qiymati sifatida 11 deb kiritiladi. Endi **ENTER** klavishi bosilsa ekranda quyidagilar aks etadi:

```
10
11
a+b=21
```

3- misol.

```
Var a,b,g,m:Integer;
Begin
Read(a, b);
Read(g); m:=a+g+b;
WriteLn('Natija= ', m);
End.
```

4- misol.

```
Var a,b,g,m:Integer;
Begin
ReadLn(a, b);
Read(g); m:=a+g+b;
WriteLn('Natija= ', m);
End.
```

3-misolda ham qiymat kiritish 1–2-usuldan birida amalga oshirilishi mumkin.

4-misolda **a** va **b** o'zgaruvchilarning qiymati probel yoki **[ENTER]** klavishlari yordamida kiritilishi mumkin. O'zgaruvchi **g** ning qiymatini kiritish uchun **b** o'zgaruvchining qiymatini kiritilgach, **[ENTER]** klavishini bosish shart. Bu holda ekranda quyidagilardan biri bo'lishi mumkin.

10 11	10
12	11
Natija= 33	12
	Natija= 33

O'zgaruvchilarga qiymat berishda o'zlashtirish operatori qo'llanilsa, dastur bitta qiymat uchun bajarilaveradi, ya'ni o'zgaruvchining qiymati xotiraga bir marta kiritiladi. Biror o'zgaruvchi qiymatini almashtirish uchun esa har safar dasturga kirib o'zgartirish zarur bo'ladi.

Dasturda biror o'zgaruvchiga qiymat berishda kiritish operatori qo'llanilgan bo'lsa, u holda dasturni ishga tushirib o'zgaruvchining qiymatini klaviaturadan kiritilaveradi, ya'ni **muloqot usulida** o'zgartirilaveradi.

Muloqot usulida ozgina noqulaylik bor, ya'ni qaysi o'zgaruvchiga qiymat kiritayotganimizni yodda tutishimiz shart. Bu noqulaylikdan qutulish uchun **Write** yoki **WriteLn** operatoridan foydalanish mumkin. Masalan, **Write(a=); ReadLn (a); Write('b= '); ReadLn (b);** yozilsa, qaysi o'zgaruvchiga qiymat kiritilayotganini ekranda ko'rinib turadi.

ReadLn operatori beradigan yana bir imkoniyat bor. Ma'lumki, Paskal dasturlash tili amallarni juda tez bajargani uchun natijani ko'rishga ulgurib bo'lmaydi. Avval aytib o'tilganidek, ekrandagi natijani ko'rish uchun **ALT+F5** klavishlar juftligi bosiladi.

Barcha natijalar ekranga chiqarilganidan so'ng natijani ko'rib olgunitmizcha dasturni kuttirish uchun **end.** operatoridan avval ro'yxatsiz **readln** operatorini yozish kifoya. Bunday holda dastur ishini yakunlash uchun **ENTER** klavishi bosiladi. Albatta, bu operator kutilgan natijani berishi uchun undan oldin yozilgan oxirgi kiritish operatori ham **LN** qo'shimchasi bilan yozilgan bo'lishi shart.

Odatda, chiziqli algoritmlarning dastur shaklida yozilishi **chiziqli dastur** deb ataladi. Demak, chiziqli dasturdagi barcha operator ketma-ket kelish tartibida bajariladi va hech qanday shart tekshirilmaydi.

1- misol. Radiusi **R** bo'lgan aylananing uzunligini hisoblash dasturi tuzilsin va **R=9** birlik qiymat uchun bajarilsin.

Yechish. Aylananing uzunligini hisoblash formulasini esga olamiz: $L=2\pi R$. Paskal dasturlash tilida u $L := 2*pi*R$ ko'rinishda yoziladi. Dasturda bitta o'zgarmas **pi** va ikkita o'zgaruvchi **R** va **L** qatnashadi. Masala shartiga ko'ra **R=9**, ya'ni butun son. Shu sababli **R** o'zgaruvchi turi **Integer** deb olinadi. Aylananing uzunligi **L** esa ko'paytmada π qatnashganligi uchun, albatta haqiqiy (**Real**) turli bo'ladi. Aytilganlarni hisobga olib quyidagi dastur tuziladi:

```

Program aylana_uzunligi;
  Var R:Integer; L:Real;
  Begin
    r := 9; L := 2*pi*R; WriteLn('L= ',L,' birlik. '); readln;
  End.

```

Dastur ishga tushirilgach (**Ctrl+F9** klavishalar juftligi bosilgach yoki **Run** menyusidan **Run** bandi tanlangach), ekranda quyidagi natija hosil bo'ladi: **L= 5.6548667765E+01 birlik**.

Dasturda L – real turli bo'lgani uchun natija eksponensial ko'rinishda chiqdi. Agar chiqarish operatorida L:7:2 formatidan foydalanilsa radiusi 9 birlik bo'lgan aylananing uzunligi 56,54 birlikka tengligi ko'rinadi.

Bu dastur yordamida ixtiyoriy butun radiusli aylananing uzunligini hisoblash mumkin. Faqat har safar dasturdagi **R** ning qiymatini o'zgartirib turish lozim. Dasturga har safar o'zgartirish kiritmaslik uchun **R** ning qiymati kiritish operatori yordamida beriladi. Radiusning qiymati doim butun son bo'lavermasligini inobatga olib, uni **Real** turdagi o'zgaruvchi sifatida tavsiflanadi. Bularni hisobga olib quyidagi dastur tuziladi:

Dasturi	Ekrandagi natija
<pre> Program aylana_uzunligi; Var r, L : Real; Begin Write('Radiusni kiriting : '); ReadLn(r); L := 2*pi*r; WriteLn('L= ',L,'birlik. '); readln; End. </pre>	Radiusni kiriting : 9 L=5.6548667765E+01 birlik

Bu dastur ishga tushirilgach, ekranga «Radiusni kiriting : »yozuvi chiqadi va yurgich shu satrda qoladi. ReadLn operatori dastur bajarilishini to'xtatib, R o'zgaruvchiga qiymat berilishini kutadi. Klaviatura orqali radiusning sonli qiymati 9 ni kiritib **ENTER** klavishi bosilsa, R o'zgaruvchining qiymati 9 ga teng deb olinib, dastur bajarilishda davom etadi. Natijada kompyuter ekranida izlangan natija hosil bo'ladi. Dasturni qayta-qayta ishlatib, turli radiusli aylanalar uzunligini hisoblashni tashkil etish mumkin.

2- misol. Tomonlari mos ravishda a , b , c bo'lgan ixtiyoriy uchburchakning yuzini Geron formulasi orqali hisoblash dasturini tuzing va $a=3$, $b=4$, $c=5$ qiymatlarda hisoblang.

I usul	II usul
<pre> Program Uchburchak_yuzi; Var a,b,c:Integer; {uchburchak tomonlari} yp,s:Real; {yp-yarim perimetr, s-yuza} Begin a:=3; b:=4; c:=5; yp:=(a+b+c)/2; s:=sqrt(yp*(yp-a)*(yp-b)*(yp-c)); WriteLn('S= ',s,' kvadrat birlik'); Readln; End. </pre>	<pre> Program Uchburchak_yuzi; Var a,b,c:Integer; {uchburchak tomonlari} yp,s:Real; {yp-yarim perimetr, s-yuza} Begin Write('a,b,c ning qiymatlari kiritilsin '); ReadLn(a,b,c); yp:=(a+b+c)/2; s:=sqrt(yp*(yp-a)*(yp-b)*(yp-c)); WriteLn('S= ',s:2:2,' kvadrat birlik'); readln; End. </pre>
S = 6.0000000000E+00 kvadrat birlik	a,b,c ning qiymatlari kiritilsin 3 4 5 S = 6.00 kvadrat birlik

3- misol. Tomonlari R bo'lgan teng tomonli uchburchak, kvadrat va radiusi R ga teng doiraning yuzini hisoblash dasturini tuzing va $R=4$ da hisoblang.



Dasturi	Ekrandagi natija
<pre> Program Yuzalar_hisoblash; var r: Integer; s1,s2,s3:Real; begin Write('R ning qiymatini kiriting :');ReadLn(r); s1:=sqr(r)*sqrt(3)/4; s2:=sqr(r); s3:=pi*sqr(r); WriteLn('Uchburchak yuzi = ',s1); WriteLn('Kvadrat yuzi = ',s2); WriteLn('Doira yuzi = ',s3); readln; end. </pre>	R ning qiymati kiritilsin 4 Uchburchak yuzi = 6.9282032303E+00 Kvadrat yuzi = 1.6000000000E+01 Doira yuzi = 5.0265482457E+01

Ushbu dasturni o'zlashtirish operatori yordamida o'zgartirib natija olishni mustaqil ish sifatida qoldiriladi.

Nazorat savollari

1. O'zlashtirish operatori qanday vazifani bajaradi?
2. O'zlashtirish operatorining umumiy ko'rinishini misollar yordamida izohlang.
3. Ma'lumotlarni ekranga chiqarish operatorining umumiy ko'rinishi qanday?
4. Write va Writeln operatori farqini misollar yordamida izohlang.
5. Chiqarish formati nima va u nima uchun qo'llaniladi?

9-MA'RUZA: TARMOQLANUVCHI HISOBLASH JARAYONLARINI DASTURLASH

Ma'ruza rejası:

1. Mantiqiy amallar va ifodalar.
2. Shartli va shartsiz o'tish operatorlari.
3. Tanlash operatori.

Mantiqiy amallar:

AND («VA» – mantiqiy ko'paytirish amali)	OR («YOKI» – mantiqiy qo'shish amali)
NOT («EMAS» – mantiqiy inkor amali)	XOR (o'xshashlikni inkor etish amali)

Mantiqiy o'zgarmaslar True (rost) yoki False (yolg'on) mantiqiy qiymatlardan biridir.

Mantiqiy o'zgarmas qiymatlarni qabul qiladigan o'zgaruvchilar mantiqiy o'zgaruvchilar deb atalib, Paskalning **Boolean** maxsus so'zi orqali tavsiflanadi. Masalan:

```

var natija : Boolean;
    katta, kichik : Boolean;

```

Misol: a, b, c butun sonlari berilgan. Jumlanı rostlikka tekshiring: "a, b, c tomonli uchburchak yasash mumkin".

Program Boolean_33;

```

var a, b, c: integer;    l: boolean;
begin
  write('Uchta (a, b, c) butun sonlarni kiriting -> ');
  readln(a, b, c);
  l:=(a+b>c) and (a+c>b) and (b+c>a);
  writeln('a, b, c tomonli uchburchak yasash mumkin = ', l);
end.

```

Biz hozirgacha chiziqli, ya'ni buyruqlari ketma-ket bajariladigan dasturlar bilan tanishdik. Ba'zan berilgan masalani hal qilishda operatorlarning bajarilish tartibini buzishga, ya'ni boshqarishni dastur bo'yicha orqaga yoki oldinga o'tkazish zarur bo'ladi. Buning uchun dasturda boshqarish uzatilayotgan operatorga **nishon** qo'yiladi. Nishon o'zgaruvchining nomi kabi lotin harflari va raqamlar yordamida hosil qilinadi. Masalan, 7, N1, nishon2. Ularda ishlatiladigan son 0 dan 9999 gacha bo'la oladi. Birinchi kelgan nollar hisobga olinmaydi.

Dasturda qo'llaniladigan nishonlar dasturning tavsif qismida **Label** xizmatchi so'zi yordamida ko'rsatilishi shart. Nishonlardan dasturda **o'tish operatori** qo'llanilsagina foydalaniladi. O'tish operatori quyidagi ko'rinishga ega: **GOTO <nishon>;**, bu yerda **GOTO** (ing.- ga o'tilsin) operatori boshqarishni dasturning oldiga **nishon** qo'yilgan operatoriga uzatadi.

1- misol. Label N1; Var a,b,c:Integer; Begin a:=15; b:=13; c:=a+b; GoTo N1; {Boshqarish N1 nishonli operatorga uzatildi} c:=a-b; N1: WriteLn(c); End.	Bu dasturning ishlashi natijasida ekranda hosil bo'lgan c ning qiymati 28 ga tengdir. Chunki, boshqarish N1 nishonli chiqarish operatoriga uzatilgani uchun c:= a-b amalni bajar-masdan o'tkazib yuborildi.
--	--

O'tish operatorida hech qanday shart tekshirilmasdan boshqarish ko'rsatilgan nishonli operatorga uzatiladi. Lekin aksariyat masalalarni hal etishda biror shartning bajarilishiga qarab u yoki bu amallar ketma-ketligini bajarish kerak bo'ladi. Masalan, kvadrat tenglamani yechishda natijani hisoblash uchun diskriminantning ishorasiga qarab uch yo'nalishdan biri tanlanadi. Bu izohlar **tarmoqlanuvchi algoritmlar** mavzusini yodingizga solgan bo'lishi kerak. Bunday masalalarni hal qilish uchun Paskalda **tarmoqlanish operatori** qo'llaniladi.

Tarmoqlanish operatorining umumiy shakli quyidagicha:

If <shart> Then <operator yoki operatorlar> Else <operator yoki operatorlar>;

Bu yerda **if, then** va **else** Paskalning xizmatchi so'zlari bo'lib, ularning o'qilishi va ma'nosi quyidagicha: **If** (if) – «agar», **Then** (then) – «u holda», **Else** (else) – «aks holda». Odatda, <shart> rost yoki yolg'on qiymatlardan birini qabul qiluvchi mantiqiy ifoda; <operator yoki operatorlar> Paskalning ixtiyoriy operatori yoki operatorlari ketma-ketligidir. Shuni yodda tutingki, **Else** xizmatchi so'zidan avval yozilgan operatorlardan keyin «;» (nuqtali

vergul) qo'yilmaydi. Tarmoqlanish operatori quyidagicha ishlaydi: avval **shart** tekshiriladi, agar uning qiymati rost bo'lsa; **then** dan keyingi operator yoki operatorlar ketma-ketligi, aks holda **else** dan keyingi operator yoki operatorlar ketma-ketligi bajariladi.

2- misol. Kiritilgan son 25 dan katta bo'lsa, ekranga «katta», aks holda «katta emas» degan yozuvni chiqaruvchi dastur tuzing.

Yechish. Berilgan sonning turi ko'rsatilmagani uchun uni haqiqiy turda deb olinadi.

```
Program Taqqoslash;  
Var a:Real;  
Begin Write('Ixtiyoriy son kiriting: '); ReadLn(a);  
      If a>25 Then WriteLn('Katta')  
                Else WriteLn('Katta emas');  
End.
```

Tarmoqlanish operatorining qismlarini alohida satrlarda yozish ham mumkin.

Agar THEN yoki ELSE so'zlardan keyin bajarilishi lozim bo'lgan ikki yoki undan ortiq operatorlar ketma-ketligi yozilgan bo'lsa, bu operatorlar ketma-ketligi albatta **begin** xizmatchi so'zi bilan boshlanib, **end;** xizmatchi so'zi bilan tugallanishi lozim.

3- misol. Kiritilgan a sonning b songa ko'paytmasini va nisbatini hisoblovchi dastur tuzing.

Yechish. Berilgan sonlar turi ko'rsatilmagani uchun ular haqiqiy turda deb olinadi.

```
Program Nisbat;  
Label tamom;  
Var a, b: Real;  
Begin  
  Write('a sonni kiriting: '); ReadLn(a);  
  Write('b sonni kiriting: '); ReadLn(b);  
  WriteLn('Ko'paytma: ', a*b);  
  If b=0 Then begin WriteLn('Bo'lishni bajarish mumkin emas');  
goto tamom; end;  
  WriteLn(' Bo'linma: ', a/b);  
tamom: End.
```

Tarmoqlanish operatorining **Else** qismi zaruratga qarab ishlatiladi. Ya'ni, tarmoqlanish operatoridan quyidagi shaklda ham foydalanish mumkin:

If <shart> Then <operator yoki operatorlar>

Bu tarmoqlanish operatorining **qisqa** shakli deyiladi. Bu holda **shart**-ning qiymati rost bo'lsa **Then** dan keyingi operator yoki operatorlar ketma-ketligi bajariladi, aks holda boshqarish navbatdagi (tarmoqlanish operatoridan keyingi) operatorga o'tadi.

4- misol. Berilgan butun son manfiy bo'lsa, bu sonni kubi bilan almashtiruvchi dastur tuzing.

Yechish.

```
var a:Integer; {berilgan son}  
begin    Write('Ixtiyoriy butun son kiriting: '); ReadLn(a);  
    If a<0 Then a:= a*a*a; {son manfiy bo'lsa kubi bilan almashtiriladi}  
    WriteLn(a); readln; {readln operatori dastur natijasini ko'rib olish  
uchun yozildi}  
end.
```

Tarmoqlanish operatori tarkibida yana tarmoqlanish operatori qo'llanishi mumkin.

5- misol. Sonning ishorasini aniqlovchi dastur tuzing.

```
Var a:Integer; b:String; {Bitta satrda bir nechta o'zgaruvchini tavsiflash  
mumkin}
```

```
Begin  
    Write('Ixtiyoriy son kiriting: '); ReadLn(a);  
    If a<0 Then b:= 'manfiy' Else If a>0 Then b:= 'musbat' Else b:= 'nol';  
    WriteLn(b);  
End.
```

6- misol. Ikkita sondan kattasining topish (IKT) dasturini tuzing.

```
Var a,b,katta:Real;
```

```
Begin  
    Write('Birinchi sonni kiriting = '); ReadLn(a);  
    Write('Ikkinchi sonni kiriting = '); ReadLn(b);  
    If a>b Then katta:=a Else katta:=b; WriteLn('Katta son= ', katta);  
End.
```

Bu dasturda $a > b$ shart bajarilishi yoki bajarilmasligidan qat'i nazar WriteLn('Katta son= ', katta) operatori albatta bajariladi. Chunki, u dasturda tarmoqlanish operatori bilan bir satrda yozilgan bo'lsa ham uning tarkibiga kirmaydi. Nima uchun shundayligini o'ylab ko'ring!

Hozirgacha biz sodda shartlar asosida dastur tuzishni ko'rib chiqdik. Ammo tarmoqlanish operatorida murakkab shartlardan ham foydalanish mumkin. Murakkab shartlar oddiy shartlarga **NOT** – «emas», **AND** – «va», **OR** – «yoki» mantiqiy amallarini qo'llash natijasida hosil qilinadi.

Yodingizda bo'lsa **NOT** – mantiqiy inkor, **AND** – mantiqiy ko'paytirish va **OR** – mantiqiy qo'shish amallari deb yuritiladi. Bunday amallar bilan biz 8-sinfдан tanishmiz. **NOT** o'zidan keyin yozilgan shartning inkor etilgan qiymatini beradi. **AND** ikki yonida joylashgan shartlarning har ikkalasi rost bo'lgandagina rost qiymat beradi. **OR** ikki yonida joylashgan shartlardan hech bo'lmaganda bittasi rost bo'lgandagina rost qiymat beradi.

Mantiqiy ifodalarda birinchi navbatda **NOT** amali, ikkinchi navbatda **AND**, uchinchi navbatda **OR** amali bajariladi. Agar mantiqiy ifodalarda qavslar qatnashgan bo'lsa, ularning ichidagi ifoda birinchi bo'lib bajariladi. Teng kuchli amallar ketma-ket kelganda, amallar chapdan o'ngga qarab bajariladi. Mantiqiy amallar qo'llanganda shartlar qavs ichiga yoziladi. Masalan,

1) $x \in [a, b]$ (ya'ni, $a \leq x \leq b$) ni Paskalda **(A<=X) AND (X<=B)** shaklida yoziladi;

2) $\overline{t_1 = t_2}$ ni Paskalda **NOT(T1=T2)** shaklida yoziladi;

3) $y < -5$ yoki $y > 2$ ni Paskalda **(Y<-5) OR (Y>2)** shaklida yoziladi.

1- misol. x ning berilgan qiymatida quyidagi funksiyaning qiymatini hisoblash dasturini tuzing.

$$y = \begin{cases} x^2, & \text{agar } x \in (0, 1] \text{ bo'lsa} \\ x, & \text{agar } x \notin (0, 1] \text{ bo'lsa} \end{cases}$$

Yechish.

Var x, y: real;

Begin Write('x= '); ReadLn(x);

If (0<x) And (x<=1) Then y:=Sqr(x) Else y:=x;

WriteLn(y= ',y): ReadLn;

End.

Bu yerda $0 < x$ va $x \leq 1$ shartlarning har ikkalasi ham bajarilganda, ya'ni x ning qiymati $(0, 1]$ oraliqqa tegishli bo'lsa, $y := \text{Sqr}(x)$ o'zlashtirish operatori, aks holda, ya'ni x ning qiymati $(0, 1]$ oraliqqa tegishli bo'lmaganda, $y := x$ o'zlashtirish operatori bajariladi.

Quyidagi tarmoqlanishga oid misollar e'tiborga loyiqdir:

1) If (A>B) And (B>C) Then S:=B+7 ELSE S:=A*B-1;

Agar $A > B$ va $B > C$, ya'ni $A > B > C$ bo'lsa, u holda $S := B + 7$ operatori bajariladi, aks holda $S := A * B - 1$ operatori bajariladi.

2) If 5*B=M*M Then Goto 200 ELSE Goto 400;

Agar $5*B = M*M$ bo'lsa, u holda boshqaruv «200» nishonli operatorga, aks holda «400» nishonli operatorga o'tadi.

3) If R1<=R2 Then begin WriteLn(S); R:=R1+R2 end

Else begin WriteLn(S*R1); R1:=R2; R2:=0; end;

Agar $R1 \leq R2$ bo'lsa, u holda $\text{WriteLn}(S)$ va $R := R1 + R2$ operatorlari bajariladi, aks holda $\text{WriteLn}(S * R1)$; $R1 := R2$; va $R2 := 0$; operatorlari bajariladi.

4) If SR ='YASHIL' Then WriteLn('O'tish mumkin') Else WriteLn('O'tish mumkin emas');

Agar SR (svetofor rangi) ning qiymati «YASHIL» bo'lsa $\text{WriteLn}(\text{'O'tish mumkin'})$ operatori, aks holda $\text{WriteLn}(\text{'O'tish mumkin emas'})$ operatori bajariladi.

Ko'rinib turibdiki, agar berilgan shart o'rinli bo'lsa, u holda THEN xizmatchi so'zidan keyin yozilgan ko'rsatmalar bajariladi, aks holda ELSE xizmatchi so'zidan keyin yozilgan ko'rsatmalar bajariladi. Bu yerda uchinchi holat bo'lmasligini puxta tushunib olish lozim.

Endi tarmoqlanuvchi dasturlarga misollar ko'ramiz:

2- misol. Berilgan uchta a , b , c son ichidan kattasini (UKT) topish dasturini tuzing.	3- misol. Berilgan natural sonning toq yoki juftligini aniqlash dasturini tuzing.
<pre> Program UKT; Var a,b,c,max : Real; Begin Write('a,b,c sonlarning qiymatini kiriting: '); ReadLn(a,b,c); If a>b Then max:=a Else max:=b; If c>max Then max:=c; WriteLn('Berilgan uchta sondan kattasi= ',max); End. </pre>	<pre> Program Toq_juft; Var n : word; Begin Write('Natural son kiriting:'); ReadLn(n); If Odd(n) Then WriteLn('TOQ') Else WriteLn('JUFT '); End. </pre>

4- misol. $ax^2 + bx + c = 0$ kvadrat tenglamani yechish dasturini tuzing.

```

Program Kvadrat_tenglama;
Label Tamom;
Var a,b,c,d,x1,x2 : Real;
Begin

```

```

Write('a,b,c larning qiymatini kiriting: '); ReadLn(a,b,c);
d:=Sqr(b) - 4*a*c; {Diskriminant hisoblandi}
If d<0 Then begin WriteLn('Haqiqiy yechim yo`q'); Goto tamom;
end;
If d=0 Then begin WriteLn('Yechim bitta: '); WriteLn('x= ',
-b/(2*a)); Goto Tamom; end;
WriteLn('Yechim ikkita: ');
x1:=(-b-Sqrt(d))/(2*a); x2:=(-b+Sqrt(d))/(2*a);
WriteLn('x1= ',x1); WriteLn('x2= ',x2);
Tamom: ReadLn;
End.

```

Yuqoridagi masalarning yechimlaridan ko'rinadiki, tarmoqlanuvchi dasturlarda tarmoqlanishni tashkil etish qo'yilgan masalaning mohiyatidan kelib chiqar ekan.

Nazorat savollari

1. Tarmoqlanish operatorining qisqa va to'liq shakllari orasida qanday farq bor?
2. Pascalda qanday mantiqiy amallar qo'llaniladi?
3. Pascalda murakkab mantiqiy shartlarga misollar keltiring.
4. Mantiqiy ifodada amallarning bajarilish tartibini tushuntirib bering.
5. Mantiqiy ifodada qachon qavslar qo'llaniladi?

10-ma'ruza: Takrorlanuvchi hisoblash jarayonlarini dasturlash

Reja:

1. Parametrli takrorlash operatori
2. Repeat takrorlash operatori
3. While takrorlash operatori

1. Parametrli takrorlash operatori

Bir hil hisoblash jarayonlarini bir necha bor takrorlanishi **Takrorlash** deyiladi. Paskal programmalashtirish tilida takrorlash operatorining 3 xil turi mavjud.

- Parametrli takrorlash operatori
- Repeat takrorlash operatori
- While takrorlash operatori

Yechilayotgan masalaga qarab, dasturchi o'zi uchun qulay bo'lgan takrorlash operatoridan foydalanishi mumkin.

Parametrli takrorlash operatorining ikki xil ko'rinishi mavjud.

1. For *i* :=start **TO** finish **DO** <operator>;

For (uchun), TO (gacha), DO (bajarmoq) xizmatchi so'zlari.

i – takrorlash parametri;

start – takrorlash parametrining boshlang'ich qiymati;

finish - takrorlash parametrining oxirgi qiymati;

operator – takrorlash tanasi.

Takrorlash operatorining bu ko'rinishida **start** <= **finish** bo'lishi shart. Agar **start** > **finish** bo'lsa takrorlash tanasi biormarta ham bajarilmaydi.

Takrorlash parametrlariga (**i**, **start**, **finish**) butun qiymatlar berilishi shart. Bu parametrlar sifatida haqiqiy sonlar berilsa hatolik yuz beradi.

Operatorning ishlash tamoyili:

Takrorlash parametri (**i**), takrorlash parametri boshlang'ich qiymati (**start**) ni o'zlashtiradi va takrorlash tanasi (<**operator**>) bajariladi. Takrorlash parametri (**i**) birga oshirilib, takrorlash tanasi (<**operator**>) bajariladi. Bu jarayon toki takrorlash parametri (**i**), takrorlash parametrining oxirgi qiymati (**finish**) ga tenglashgunga qadar davom etadi.

Takrorlash tanasi sifatida faqat bitta operatoridan foydalanish mumkin. Agar takrorlash tanasida bir nechta operatoridan foydalanmoqchi bo'lsak bu operatorlarni **begin end** xizmatchi so'zlari orasiga olishimiz kerak.

Break – prosedurasini har qanday takrorlash operatoriga qo'llash mumkin. Bu prosedura takrorlash tugatilishini taminlaydi. Yani boshqarilishni takrorlash operatoridan keyingi operatorga uzatadi.

Continue – prosedurasini har qanday takrorlash operatoriga qo'llash mumkin. Bu prosedura takrorlash parametrining keyingi qiymatni qabul qilishini taminlaydi. Boshqacha so'z bilan aytganda takrorlash tanasi tugatiladi. Bunda takrorlashning o'zi tugatilmaydi.

Misol: N natural soni berilgan. Birdan N gacha bo'lgan sonlar yig'indisini hisoblovchi programma tuzilsin.

program takrorlash;

```

var i,n,s; integer;
begin
  write('N='); readln(n);
  s:=0;
  For i:=1 TO n DO
    s:=s+i;
  writeln('Yig`indi=',s);
  readln;
end.

```

Ayrim paytlarda takrorlash parametri o`sb borish emas, kamayib borish tartibida o`zgartirish mumkin. Bu takrorlash operatorining ikkinchi xil ko`rinishidir:

2. For i :=start **DOWNTO** finish <operator>;

Bu erda **DOWNTO** (gacha kamayib) – tilning xizmatchi so`zi.

Bu operatorida takrorlash parametri (i), takrorlash parametri boshlang`ich qiymati (**start**) ni o`zlashtiradi va takrorlash tanasi (<**operator**>) bajariladi. Takrorlash parametri (i) birga kamaytirilib, takrorlash tanasi (<operator>) bajariladi. Bu jarayon toki takrorlash parametri (i), takrorlash parametrining oxirgi qiymati (**finish**) ga tenglashgunga qadar davom etadi.

Yuqorida keltirilgan misol uchun takrorlash operatori quyidagicha yoziladi.

For i:=n **DOWNTO** 1 **DO**

s:=s+i;

Parametrli takrorlash operatori takrorlanishlar soni oldindan ma`lum bo`lgan hollarda qo`llaniladi. Takrorlashning takrorlanishlar sonini quyidagi formuladan aniqlash mumkin.

$$N = \left[\frac{x_k - x_h}{h} \right] + 1$$

Bu erda x_k oxirgi qiymat, x_h boshlangich qiymat, h qadam;

Quyidagi yig`indini hisoblovchi programma tuzilsin.

$$s = 1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{50}$$

```

program takrorlash2;
var i: integer;
    s: real;
begin
  s:=0;
  for i:=50 downto 1 do
    s:=s+1/i;
  writeln('Yig`insi=',s:2:2);
  readln;
end.

```



Nazorat savollari.

1. Takrorlash operatorlari nima uchun qo'llaniladi?
2. Parametrli takrorlash operatorining necha xil turlari mavjud ?
3. Parametrli takrorlash operatorini tushuntirib bering.
4. Parametrli takrorlash operatorida takrorlash tanasi deb nimaga aytiadi ?
5. **Break** va **Continue** proseduralari qo'llanilishini tushuntiring ?
6. Takrorlash parametrlari qanday qiymat qabul qiladi ?
7. Takrorlashning takrorlanishlar soni qanday aniqlanadi?

Mustaqil echish uchun masalalar:

1. n natural soni berilgan. Quyidagilarni hisoblovchi programma tuzilsin.
 - a) 2^n
 - b) $n!$; $n!=1*2*3*...*n$; yani 1 dan n gacha bo'lgan sonlar ko'paytmasi.

c) $\left(1 + \frac{1}{1^2}\right)\left(1 + \frac{1}{2^2}\right)....\left(1 + \frac{1}{n^2}\right)$

d) $\sqrt{2 + \sqrt{2 + ... + \sqrt{2}}} ; (n \text{ ta ildiz.})$

e) $\sqrt{3 + \sqrt{6 + ... + \sqrt{3(n-1) + \sqrt{3n}}}}$

f) $\frac{1}{\sin 1} + \frac{1}{\sin 1 + \sin 2} + ... + \frac{1}{\sin 1 + ... \sin n}$

g) $\frac{\cos 1}{\sin 1} * \frac{\cos 1 + \cos 2}{\sin 1 + \sin 2} + ... + \frac{\cos 1 + ... + \cos n}{\sin 1 + ... + \sin n}$

8. Birdan 100 gacha bo'lgan juft sonlarni chiqaruvchi programma tuzilsin.
9. a haqiqiy soni va, n natural soni berilgan quyidagilarni hisoblovchi programma tuzilsin.

b) a^n ;

c) $a(a+1)...(a+n-1)$;

d) $\frac{1}{a} + \frac{1}{a(a+1)} + ... + \frac{1}{a(a+1)...(a+n)}$;

e) $a(a-n)(a-2n)...(a-n^2)$;

4. Hisoblansin: $(1+\sin 0.1)(1+\sin 0.2)...(1+\sin 10)$;

5. X, A haqiqiy, n natural soni berilgan. Quyidagi ifodani hisoblovchi programma tuzilsin.

6. $((...(\mathbf{x+a})^2+\mathbf{a})^2+...+\mathbf{a})^2+\mathbf{a}$; bu erda n ta qavs qatnashgan.

7. n natural soni berilga. Quyidagi ifodani hisoblovchi programma tuzilsin.

$$1*2+2*3*4+...+n*(n+1)...*2n;$$

8. n, k butun sonlari berilgan. ($n \geq k \geq 0$).Quyidagi ifodani hisoblovchi programma tuzilsin.

$$\frac{n(n-1)...(n-k+1)}{k!}$$

9. Quyidagilarni hisoblovchi programma tuzilsin.

$$a) \sum_{i=1}^{100} \frac{1}{i^2};$$

$$b) \sum_{i=1}^{50} \frac{1}{i^3};$$

$$c) \sum_{i=1}^{10} \frac{1}{i!};$$

$$d) \sum_{i=1}^{128} \frac{1}{(2i)^2};$$

$$e) \prod_{i=2}^{100} \frac{i+1}{i+2};$$

$$f) \prod_{i=2}^{10} \left(1 + \frac{1}{i!}\right)^2;$$

2. Repeat takrorlash operatori

Agar takrorlashning takrorlanishlar soni oldindan ma`lum bo`lmasa, yani hisoblash jarayonida qaysidir o`zgaruvchining qiymatiga bo`g'liq bo`lsa u holda **repeat** yoki **while** operatoridan foydalaniladi.

Repeat operatori quyidagicha ishlatiladi:

repeat

<operator 1>

<operator 2>

...

<operator N >

Until <shart>;

Bu erda **repeat** (takrorlamoq), **until** (gacha) xizmatchi so`zlar;

<operator 1>; <operator 2>; ... <operator N> takrorlash tanasi;

<**shart**> takrorlash tanasi bajarilgandan so`ng, takrorlashdan chiqish uchun tekshiriladigan shart. (mantiqiy ifoda).

Operatorning ishlash tartibi:

Repeat xizmatchi so`zidan keyingi operatorlar bajariladi, keyin shart tekshiriladi. Agar shart yolg`on (false) natija bersa **repeat** xizmatchi so`zidan keyingi operatorlar qayta bajariladi. Shart qayta tekshiriladi, bu jarayon shart rost (true) natija berguncha takrorlanadi.

Misol. Quyidagi yig`indini hisoblovchi programma tuzilsin.

$$s = 1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{50}$$

Bu programma parametrli takrorlash operatoridan foydalangan holda oldingi darsda tuzilgan edi. Endi repeat takrorlash operatori orqali programma tuzamiz va takrorlash operatorlarini farqini ko`rib olamiz.

program Misol;

var i: Byte;

s: real;

begin

s:=0; i:=1;

REPEAT

s:=s+1/i;

i:=i+1;

UNTIL i>50;

```
writeln('S=',s:2:2);
readln;
```

end.



Nazorat savollari.

1. **Repeat** operatori takrorlash tanasida qanday operatorlar bo`lishi mumkin?
2. **Repeat** operatori necha marta takrorlanadi?
3. **Repeat** operatoridan qanday chiqiladi?
4. **Repeat** operatori ishlashini tushintirib bering.

Mustaqil echish uchun masalalar:

1. Oldingi mavzudagi 1–8 misollarni repeat operatoridan foydalangan holda yechib chiqing.
2. N natural soni berilgan. Quyidagilarni hisoblovchi programma tuzilsin. (Repeat takrorlash operatori orqali)

a) $\sum_{k=1}^n \frac{1}{k};$

b) $\sum_{k=1}^n \frac{1}{k^5};$

c) $\sum_{k=1}^n \frac{1}{(2k+1)^2};$

d) $\sum_{k=1}^n \frac{(-1)^k}{(2k+1)k};$

1. n natural soni va x haqiqiy soni berilgan. Quyidagilarni hisoblovchi programma tuzilsin. (Repeat takrorlash operatori orqali)

a) $\sum_{i=1}^n \frac{x^i}{i!};$

b) $\sum_{i=1}^n (\frac{1}{i!} + \sqrt{|x|});$

c) $\sum_{i=1}^n \frac{x + \cos(ix)}{2^i};$

d) $\prod_{i=1}^n (1 + \frac{\sin(ix)}{i!});$

2. n natural soni berilgan. Quyidagi qonuniyat asosida a_n ni hisoblovchi programma tuzilsin. (Repeat takrorlash operatori orqali)

$$a_0=1; \quad a_k=ka_{k-1}+1/k; \quad k=1, 2, \dots$$

3. n natural soni berilgan. ($n \geq 4$). Quyidagi qonuniyat asosida v_n ni hisoblovchi programma tuzilsin. (Repeat takrorlash operatori orqali)

$$v_1=v_2=0; \quad v_3=1.5; \quad v_i = \frac{i+1}{i^2+1}v_{i-1} - v_{i-2}v_{i-3} \quad i=4, 5, \dots$$

4. q, r, b, c, d haqiqiy sonlari va n natural soni berilgan. ($n \geq 2$). Quyidagi qonuniyat asosida x_n ni hisoblovchi programma tuzilsin. (Repeat takrorlash operatori orqali)

$$x_0=c; \quad x_1=d; \quad x_k=qx_{k-1}+Rx_{k-2}+b;$$

$$k=2, 3, \dots$$

5. $a_0=a_1=1; \quad a_i=a_{i-2}+\frac{a_{i-1}}{2^{i-1}}; \quad i=2, 3, \dots;$ bo`lsin. Quyidagi ko`paytmani hisoblovchi programma tuzilsin. $a_0*a_1*...*a_{14};$ (Repeat takrorlash operatori orqali)

6. Quyidagi ifodani hisoblovchi programma tuzilsin.

$$\begin{array}{c}
\frac{1}{1 + \frac{1}{3 + \frac{1}{5 + \frac{1}{\dots}}}} \\
101 + \frac{1}{103}
\end{array}$$

7. Noldan farqli bo'lmagan x haqiqiyni berilgan. Quyidagi ifodani hisoblovchi programma tuzilsin.

$$\begin{array}{c}
\frac{x}{x^2 + \frac{2}{x^2 + \frac{4}{x^2 + \frac{8}{\dots}}}} \\
x^2 + \frac{256}{x^2}
\end{array}$$

3. While takrorlash operatori

Repeat operatorida takrorlashning tanasi kamida bir marta takrorlanadi. Shu bir marta hisoblash ham yechilayotgan masalani mohiyatini buzib yuborishi mumkin. Bunday hollarda **While** takrorlash operatoridan foydalangan maqsadga muvofiq.

While operatorining umumiy ko'rinishi.

While <shart> **DO**

Begin

<takrorlash tanasi>;

End;

Bu yerda **While** (hozircha), **DO** (bajarmoq) xizmatchi so'zlari;

<shart> takrorlash tanasi bajarilishini ifodalovchi mantiqiy ifoda;

<takrorlash tanasi> ixtiyoriy operator yoki operatorlar majmuasi.

Operatorning ishlash tartibi.

Agar <shart> rost (true) qiymatga ega bo'lsa, <takrorlash tanasi> bajariladi. Qachonki shart yolg'on (false) qiymatga ega bo'lsa takrorlash tugatiladi.

Agar <shart> rost (true) qiymatga ega bo'lmasa, <takrorlash tanasi> biror marta ham bajarilmaydi.

Misol. N natural soni berilgan. Quyidagi yig'indini hisoblovchi programma tuzilsin.

$$s = 1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{N}$$

Program namuna;

```

Var n, i: integer;
s: real;
begin
  write (`N`); Readln (n);
  s: = 0, i: = 0;
  while i<n do
    begin
      i: = i+1;
      s: = s+1/i;
    end;
  WriteLn (`S=`, s:2:2);
  ReadLn;
end.

```



Nazorat savollari.

1. While operatori takrorlash tanasida qanday operatorlar bo`lishi mumkin?
2. While operatoridan qanday chiqiladi?
3. While operatori ishlashini tushintirib bering.

11-ma'ruza: Massivlar va ular bilan ishlash

Reja:

1. Massiv turlarini ifodalash
2. Massiv elementlarini tartiblash usullari
3. Ikki o'lchamli massivlar

Biz shu paytgacha asosan oddiy toifalar bilan ishladik. Ma'lumotlar toifasi Pascalda ikkiga bo'linadi.

Ma'lumotlar toifasi	
Oddiy toifalar	Murakkab toifalar
1. Standart	1. Massivlar
2. Sanoqli	2. To'plamlar
3. Cheklangan	3. Yozuvlar
	4. Fayllar

Massiv - bu bir xil toifali, chekli qiymatlarning tartiblangan to'plamidir. Massivlarga misol qilib matematika kursidan ma'lum bo'lgan vektorlar, matrisalarni ko'rsatish mumkin.

Massiv bir o'lchamli deyiladi, agar uning elementiga bir indeks orqali murojaat qilish mumkin bo'lsa.

Bir o'lchamli massivni e'lon qilish quyidagicha bo'ladi:

VAR massiv_nomi: ARRAY [boshlang'ich_indeks..oxirgi_indeks] OF ma'lumot toifasi;

Misol uchun:

VAR mas: ARRAY [1..25] OF INTEGER;

Yuqorida biz boshlang'ich indeksni va oxirgi indeksni 25 bo'lgan bir o'lchamli massivni e'lon qildik.

Massivlarni boshqacharoq yo'l bilan ham e'lon qilish mumkin. Misol uchun:

TYPE tab = array [1..25] **OF** integer;

VAR mas: tab;

Quyidagi massivlarni e'lon qilishga bir necha misollar keltirilgan:

VAR

mas1: **ARRAY** [1..10] **OF** REAL;

mas2: **ARRAY** [5..15] **OF** INTEGER;

mas3: **ARRAY** [1..10] **OF** BOOLEAN;

mas4: **ARRAY** [BYTE] **OF** INTEGER;

mas5: **ARRAY** [1..10] **OF** 2..5;

mas 1 elementlari haqiqiy sonlardan iborat bo'lgan 10 ta elementdan tashkil topgan massiv. Indekslari esa 1 dan 10 gacha bo'lgan sonlar (1..10).

mas2 elementlar butun sonlardan iborat bo'lgan 11 ta elementdan tashkil topgan massiv. Indekslari esa 5 dan 15 gacha bo'lgan sonlar ([5..15]).

mas3 elementlari mantiqiy qiymatlardan (TRUE, FALSE) iborat bo'lgan sonlar ([1..10]).

mas4 elementlari butun sonlardan iborat bo'lgan 257 ta elementdan tashkil topgan massiv. Indekslari esa 0 dan 256 gacha bo'lgan sonlar.

mas5 elementlari 2, 3, 4, 5 sonlaridan iborat bo'lishi mumkin bo'lgan 10 ta elementdan tashkil topgan massiv. Indekslari esa 1 dan 10 gacha bo'lgan sonlar.

Massiv elementlariga murojaat qilish oddiy o'zgaruvchilarga murojaat qilishdan biroz farq qiladi. Massiv elementiga murojaat qilish uning indeksi orqali bo'ladi.

mas 1 [5]:=10; mas 1 massivining 5- elementi 10 qiymat o'zlashtirilsin;

readln(mas 1[1]); mas 1 massivining 1- elementi kiritilsin;

writeln(mas 1[1]; mas 1 massivining 2-elementi ekranga chiqarilsin;

Misol1: Elementlari butun sonlardan iborat bo'lgan, 10 elementdan tashkil topgan massiv elementlarni kirituvchi va ekranda teskari tartiblash talab qilinsin.

Program mas;

var i: integer;

mas1: **ARRAY** [1..10] **OF** INTEGER;

{massivni e'lon qilish}

begin

writeln ('Massiv elementlarini kiriting');

for i: = 1 **to** 10 **do**

readln (mas1 [i];

writeln ('massiv elementlarini teskari tartibda chiqarish');

for i: = 10 **downto** 1 **do**

write (mas1 [i], ' ');

readln;

end.

Misol2. Yuqorida keltirilgan misoldagi massiv elementlari yig'indisini chiqaruvchi dastur tuzilsin.

```
Program massiv_sum;  
var i, s: integer;  
masl: ARRAY [1..10] OF INTEGER;  
    {massivni e'lon qilish}  
begin  
  writeln ('Massiv elementlarini kiriting');  
  for i: = 1 to 10 do  
    readln (masl [i]);  
    s:= 0;          {boshlang'ich qiymat 0}  
  for i:=1 to 10 do  
    s: = s + masl[i];  
    {massiv elementlari yig'indisini olish}  
  writeln ('Massiv elementlari yig'indisi=', s);  
  readln;  
end.
```

2. Massiv elementlarini tartiblash usullari.

Massivni tartiblashtirishning bir necha usullari (algoritmлари) mavjud. Ulardan quyidagi usullarni qarab chiqamiz:

- tanlash usuli;
- almashtirish usuli.

Tanlash usuli yordamida massivni o'sish bo'yicha tartiblashtirish algoritmi quyidagicha:

1. Massivning birinchi elementidan boshlab qarab chiqilib eng kichik element topiladi.
2. Birinchi element bilan eng kichik element joylari almashtiriladi.
3. Ikkinchi elementidan boshlab qarab chiqilib eng kichik element topiladi.
4. Ikkinchi element bilan eng kichik element joylari almashtiriladi.
5. Bu protsess bitta oxirgi elementgacha takrorlanadi.

Bu algoritm dasturi quyidagicha bo'ladi:

```
Program Sort;  
Const Size=5;  
Var i,j,min,k,buf: Integer;  
a: Array[1..Size] of Integer;  
Begin  
  Writeln ('Massivni tartiblashtirish');  
  Write (Size:3, 'ta massiv elementini kiriting');  
  For k:=1 to Size Do Read(a[k]);  
  Writeln ('Tartiblashtirish');  
  For i:=1 to Size-1 Do  
    Begin
```

```

{ kichik elementni topish }
min:=i;
For j:=i+1 to Size Do
  Begin
    If a[j]<a[min] then min:=j;
    buf:=a[i]; a[i]:=a[min]; a[min]:=buf;
    For k:=1 to Size Do Write (a[k], ' ');
    Writeln;
  End;
End;
Writeln('Massiv tartiblashtirildi.');
```

Dastur natijasi:

Massivni tartiblashtirish 5 ta massiv elementini kiriting

12 -3 56 47 10

Tartiblatirish

-3 12 56 47 10

-3 10 56 47 12

-3 10 12 47 56

-3 10 12 47 56

Massiv tartiblashtirildi.

Almashtirish usuli yordamida massiv elementlarini o'sib borishda tartiblashtirish algoritmi quyidagicha:

1. Massivning birinchi elementidan boshlab ketma-ket hamma qo'shni elementlar bir-biri bilan solishtirilib, agar birinchisi ikkinchisidan kichik bo'lsa ular joyi almashtirilib boriladi.

2. Bu protsess davomida kichik qiymatli elementlar massiv boshiga katta elementlar esa oxiriga siljutilib boriladi. Shu sabab bu usul «puzirka» usuli ham deyiladi.

3. Bu protsess massiv elementlar sonidan bitta kam marta takrorlanadi.

Masalan:

3 2 4 5 1 bunda 3 bilan 2 va 5 bilan 1 almashtiriladi.

2 3 4 1 5 bunda 4 bilan 1 almashtiriladi.

2 3 1 4 5 bunda 3 bilan 1 almashtiriladi.

2 1 3 4 5 bunda 2 bilan 1 almashtiriladi.

1 2 3 4 5

Bu algoritm dastursi quyidagicha bo'ladi:

Program Sort;

Const Size=5;

Var i,j,min,k,buf: Integer;

a: Array[1..Size] of Integer;

Begin

Writeln ('Massivni puzirek(kupikcha) usulida tartiblashtirish');

```

Write (Size:3,'ta massiv elementini kiriting');
For k:=1 to Size Do Read(a[k]);
Writeln ('Tartiblatirish');
For i:=1 to Size-1 Do
    Begin
        For k:=1 to Size-1 Do
            Begin
                If a[k]>a[k+1] then
                    Begin
                        buf:=a[k]; a[k]:=a[k+1]; a[k+1]:=buf;
                    End;
            End;
        For k:=1 to Size Do Write (a[k],' ');
        Writeln;
    End;
Writeln('Massiv tartiblashtirildi.');
```

Dastur natijasi:

Massivni puzirek usulida tartiblashtirish 5 ta massiv elementini kiriting

3 2 4 1 5

Tartiblashtirish

2 3 4 1 5

2 3 1 4 5

2 1 3 4 5

1 2 3 4 5

Massiv tartiblashtirildi.

Massivda eng kichik yoki eng katta elementni izlash algoritmi ma'lumki birinchi element eng kichik (katta) deb olinib keyin boshqa elementlar bilan ketma-ket solishtirilib chiqiladi. Solishtirish oxirgi elementgacha bajariladi.

Quyida bu algoritm dasturi keltirilgan:

Program MinMax;

Var i,min: Integer;

a: Array[1..10] of Integer;

Begin

Writeln ('Massivdan eng kichik elementni izlash');

Write (' 10-ta massiv elementini kiriting');

For i:=1 to 10 Do Read(a[i]);

min:=1;

For i:=2 to 10 Do

If a[i]<a[min] Then min:=i;

Writeln('Izlanayotgan eng kichik element:',a[min]);

Writeln('Element nomeri',min);

End.

3. Ikki o'lchamli massivlar

Bir o'lchamli massivlar uchun ishlatilgan o'zgaruvchilar, bir xil jinsdagi berilganlarni xotirada saqlash uchun foydalaniladi. Ikki o'lchamli massivlarda esa, satr va ustunlar orqali bir xil jinsdagi qiymatlarni ikki o'lchamli o'zgaruvchilar ichida saqlash uchun foydalaniladi.

Ikki o'lchamli massivlarda o'zgaruvchilarni e'lon qilish.

TYPE

Massiv_ismi=ARRAY [eng kichik index...eng kata index, eng kichik index... eng kata index] OF element_toifasi;

VAR

Massiv_o'zgaruvchisi: massiv_tur_ismi;

Yoki

Massiv_o'zgaruvchisi:array [engkichikindex...engkattaindex
Engkichikindex... ..engkattaindex] OF element_toifasi;

Misol;

Type

Ism=array [1...10,1...5] of string [5];

Var

Talaba_ismi: ism;

Yoki

Talaba_ismi: array [1...10,1...5] ofstring [5];

Ikki o'lchamli massivlarning e'lon qilinishida, bir o'lchamlidan yagona farqi, array so'zidan keyin qirrali qavs ichida yozilgan qiymatlar ikki marta yozilganligidadir.

Bulardan birinchisi, satr nomeri, ikkinchisi esa ustun nomerini beradi. Ikki o'lchamli massivlar matematika kursidan ma'lum bo'lgan matrisalarni eslatadi.

Var a: array [1...3, 1...3] of integer;

b: array [1...3, 1...4] of integer;

A matrisa	B matrisa
a ₁₁ a ₁₂ a ₁₃	b ₁₁ b ₁₂ b ₁₃ b ₁₄
a ₂₁ a ₂₂ a ₂₃	b ₂₁ b ₂₂ b ₂₃ b ₁₄
a ₃₁ a ₃₂ a ₃₃	

A matrisa 3 ta satr, 3 ta ustunga ega;

B matrisa 2 ta satr, 4 ta ustunga ega;

a₁₁- a biru bir element deb o'qiladi. A o'nbir deyilmaydi.

M ta satr va n ta ustunga ega bolgan massivga (mxn) o'lchamli massiv deyiladi. Agar m=n bo'lsa kvadrat massiv deyiladi.

Misol: A(3x3) matrisa berilgan. Shu matrsa elementlarini kirituvchi va ekranga jadval ko'rinishida chiqaruvchi programma tuzilsin.

Program Matrisa;

Var a : Array [1..3, 1..3] of integer;

i : integer; {Satrlar indeksi}

```

        j : integer; {Ustunlar indeksi}
Begin
    Writeln ( 'Massiv elementlarni kiritish' );
    For i := 1 to 3 do
        For j := 1 to 3 do
            begin
                Writeln ( 'a[ , i , j, ' ] =' );
                Readln ( a[ i , j ] );
            End;
        Writeln ( 'Massiv elementlarini Ekranga chiqarish ' );
    For i:=1 to 3 do
        Begin
            For j := 1 to 3 do
                Write ( a [ i, j ] : 2 );
            End;
        Readln;
    End.

```

12-ma'ruza: Satr tipidagi ma'lumotlar

Reja:

1. Belgili (simvolli) massivlar. Satrlar
2. Satrlar bilan ishlashda qo'llaniladigan prosedura va funksiyalar.
3. Satrlarni ishlatilishiga misol.

Belgili (simvolli) massivlar. Satrlar

Satrlar bilan ishlashning eng oddiy usullaridan biri bu CHAR toifasidagi chiziqli massiv hosil qilishlikdir.

Var ms: ARRAY [1..50] OF CHAR;

Shu o'rinda massivning barcha qoidalaridan foydalanish mumkin, lekin satrlar bilan ishlashning bu usuli ancha noqulayliklarni olib keladi. Shuning uchun Pascalda satrlar qatori bilan ishlash uchun maxsus **STRING** toifasi kiritilgan.

Dasturlash davomida bir necha bor satrlarga duch keldik. Misol uchun:

Writeln (qo'shtirnoq ichidagi ifoda bu – satr `) simvol nomeri 32 bo'lgan belgi satrlar bilan ishlashda qo'llaniladi. Agar bu belgini satr ishida ishlatilishiga to'g'ri kelib qolsa qo'shimcha yana birmarta yoziladi.

Writeln (` Qo'shtirnoqli (``) satr `) ; satrlarni e'lon qilish uchun STRING xizmatchisi so'zi ishlatiladi.

Var S1: string [50] ;
Yoki

Type Satr: STRING [50]

Var S1: satr;

S1 o'zgaruvchisi Char toifasidagi ixtiyoriy 50 tagacha belgini olish mumkin.

Satr o'zgaruvchiga qiymatni o'zlashtirish operatori yoki kiritish operatori orqali berish mumkin.

```
S1:= `Satrlarga misol`;  
Readln (s1) ;  
Satrli o'zgaruvchilarni qo'shish (yig'ish)  
Var ism, fam: string [15]  
FIO: string [30]:  
begin  
ism:= `Abror`;  
fam:= `Inatov` :  
Fio:= fam+ism{Fio:= fam+` `+ism;}  
writeln (fio)  
readln:  
end.
```

Ekranga quyidagicha natija chiqadi: Inatov Abror

Familiya va ismini ajratish uchun shaklli qavsda keltirilgan holda yozish lozim. Pascal da satrli o'zgaruvchining eng uzun qiymati 255 ta belgidan oshmaydi. Agar satrli o'zgaruvchilarning hajmi keltirilmagan bo'lsa 255 ta belgi deb qabul qilinadi.

Quyida ikkita bir xil uzunlikdagi o'zgaruvchilar e'lon qilingan.

```
Var S1: STRING;  
S2: STRING [255]  
Satrlar bilan ishlovchi funksiyalar va proseduralar  
Concat (s1, [s2 ..., sn] ; string ) : string  
Bir necha satrlarni yig'ish  
Length (S; string) ; integer;  
Satrning joy uzunligini aniqlash
```

```
Var s ; string  
begin  
writeln (` satr kiriting`)  
readln (s) ;  
writeln (` bu satr `Length (s) , ` ta belgidan iborat` ) :  
readln:  
end.
```

```
Pos ( P,S: string ) ; Byte;
```

P satri, S satridan izlash. Natija s o'zgaruvchidan uchragan birinchi belgining o'rnini, agar P o'zgaruvchisidagi satr s o'zgaruvchisidan topilmasa nol qiymat qaytaradi.

```
Copy (S: string: 1 ; integer ; C: integer):  
Satrning ma'lum qismidan nusxa olish:  
S - Joriy satr;
```

I - nusxa olish boshlanadigan index (Agar satr uzunligidan katta son kiritilsa bo'sh satr qaytariladi).

C - nusxa olinadigan belgilar soni (Agar satr uzunligi kam bo'lsa, borini qaytaradi)

Copy (S: string; I:integer; C:integer);	Delete (var S: string; I: integer; C:integer
<pre> Var s1, s2, s3: string [20] begin s1:= `GULOY` ; writeln (S1); {GULOY} s2:= Copy (s1,1, 3); writeln(s2) ; {GUL} s3:= Copy(s1, 4, 2) writeln(s3) ; {OY} writeln(s3+s2); { OYGUL} readln; end </pre>	<pre> Var s,c: string [20] Begin C:= ` HON` S:= ` GULTOJI` Writeln (s): {GULTOJI} Delete(s,1, 3); Writeln (s); {TOJI} Insert (c, s, 5); Writeln (s); {TOJIHON} Readln; End. </pre>

Delete (var S: string; I: integer; C: integer);

Satrning ma'lum qismini o'chirish

S – Joriy satr;

I – O'chirish boshlanadigan index (Agar satr uzunligi katta son kiritilsa o'chirilmaydi)

C – o'chiriladigan belgilar soni

Insert (P: string; Var s: string; I: integer);

Satr orasiga qo'shish

S – Joriy satr; P – Qo'shilishi lozim bo'lgan satr;

I – Qo'shish boshlanadigan index;

Str (X [: Width [: Decimals]]; var S);

X o'zgaruvchisidagi (butun yoki haqiqiy) sonli qiymatni, S satrli o'zgaruvchisiga o'zlashtiriladi. Width X haqiqiy sonining butun qismi uchun, Decimals kasr qismidan nechta son olish uchun ajratilgan.

Val(S; var V; var Code: Integer);

S stridagi sonlarni, V o'zgaruvchisiga o'zlashtirish. Code agar satrda raqamlardan boshqa belgilar qatnashmagan bo'lsa, nolga teng bo'ladi. Aks holda satrdagi raqamlardan tashqari uchragan birinchi belgining o'rnini bildiradi.

Misol: x:=123.565; Str(x:2:2,s); { s=123.56 }

Val(s,x,code); { x=123.56, code=0}

S:= '123G456'; Val(s,x,code); { x:=123, code=4}

13-ma'ruza: Qism dasturlardan foydalanish

Reja:

1. Qism dastur. Parametrsiz protseduralar.
2. Qism dastur. Parametrli protseduralar
3. Funktsiyalar
4. Rekursiya. O'z - o'zini chaqiruvchi qism dasturlar.

1. Qism dastur. Parametrsiz protseduralar.

Dasturlash mobaynida bir xil ifodalarni, hisoblash jarayonini qayta-qayta hisoblashga to'g'ri keladi. Dasturlash tillarida, ko'mpyuter hotirasini va dasturchaning voqtini tejash maqsadida, bunday takrorlanuvchi jarayonlarni dasturda ajratib yozib, unga asosiy dasturdan murojaat qilish imkoniyatlarini keltirilgan.

Dasturning asosiy qismidan murojaat qilib, bir nechabor ishlatish mumkin bo'lgan bunday operatorlar guruhiga **qisim dastur** deyiladi.

Qisim dastur protsedurali yoki funksiyali qisim dastur bo'lishligi mumkin. Qisim dastur sifatida olingan natijada yagona qiymat qabul qilsa funksiyasidan, olinadigan natijalar bir nechta bo'lsa protseduradan foydalanish maqsadiga muvofiqdir.

Protseduralarning ikki xil turi mavjud.

1. parametrli protseduralar
2. parametrsiz protseduralar

Bu mavzuda parametrsiz protseduralarga to'xtalib o'tamiz.

Parametrsiz protseduralarning umumiy ko'rinishi

Procedure	<protsedura nomi>;
Uses	<modullar nomi>;
Label	<metkalar nomi>;
Const	<o'zgarmlar nomi>;
Type	<foydalaniladigan toyifalar nomi>;
Var	<o'zgaruvchilar nomi>;
	<protseduralarni e'lon qilish bo'limi>;
Begin	
	<operatorlar>;

End;

Parametrsiz protseduralarning **pascal** dasturidan farqi.

Parametrsiz protseduralarning pascal dasturidan yagona farqi sarlovhasida bo'lib, qism dasturda **program** xizmatchi so'zidan emas, **procedure** xizmatchi so'zidan foydalaniladi. Bundan tashqari protsedura sarlovhasini yozish shart. Yana shuni takitlash lozimki, har qanday qism dastur **End** xizmatchi so'zidan kegin ";" bilan tugatiladi.

Parametrsiz protseduraga murojat qilishda dastur tanasida protsedura nomi yoziladi. Dasturda protsedura nomi operatorlar kabi ishlatiladi. Bu operatorlar bajarilganda protsedura to'liq bajariladi. Parametrsiz protsedurada asosiy dasturning barcha (global) o'zgaruvchilardan foydalanish mumkin.

Global o'zgaruvchilar

Ham asosiy programmada, ham protsedurada ishlatiladigan o'zgaruvchilar, asosiy programmaning e'lon qilish blokida e'lon qilinadi. Bu o'zgaruvchilar asosiy programmadan protseduraga uztib, uyerda ishlatilgandan kegin, natijaviy qiymatni yana asosiy programmaga uzatilishi mumkin.

Local o'zgaruvchilar

Faqatgina protsedurada ishlatiladigan o'zgaruvchilardir. Ular protsedurada e'lon qilinadi va protseduradan olgan yangi qiymatni asosiy programmaga uzatib bo'lmaydi.

Qism dasturda yana bir nechta ichki qism dasturlardan foydalanish mumkin.

Misol : $\frac{\sqrt{|x-1|}}{1 + \frac{x^2}{2} + \frac{y^2}{4}}$ ifoda ($x=2, y=2$), ($x=2, y=1$), ($x=5, y=4$) bo'lgan hollar

uchun hisoblash talab qilinsin. Bu misolni ikki xil usulda ishlab solishtirib ko'ramiz.

Oddiy usul	Qism dastur
Program usul_1; var x, y, s: real ; Begin x:=2; y:=2; s:=sqrt(abs(x-1))/(1+sqr(x)/2+sqr(y)/4); writeln (s:3:2); x:=2; y:=1; s:=sqrt(abs(x-1))/(1+sqr(x)/2+sqr(y)/4); writeln (s:3:2); x:=5; y:=4; s:=sqrt(abs(x-1))/(1+sqr(x)/2+sqr(y)/4); writeln (s:3:2); readln ; end.	Program usul_2; var x, y, s: real ; procedure hisobla; begin s:=sqrt(abs(x-1))/(1+sqr(x)/2+sqr(y)/4); writeln (s:3:2); end Begin x:=2; y:=2; hisobla x:=2; y:=1; hisobla x:=5; y:=4; hisobla readln ; end.

Birinchi usulda bir xil ifoda takror yozilgan. Ikkinchi usulda esa protsedura orqali ixcham va sodda ko'rinishda berilgan. Bu oddiy hisoblash jarayoni takrorlovchi holat uchun. Endi o'zingiz murakkab hisoblash jarayonlari uchun protseduradan foydalanish qanchalik qulay ekanligini o'ylab ko'ring.

2. Qism dastur. Parametrli protseduralar

Pascal da barcha operatorlar protsedure yoki funksiyalardir. Misol uchun readln(c), writeln(s:2:2) va hakazo. Dasturchi tomonidan kiritilgan protseduralar ham parametrli protseduralar bo'lishim mumkin. Asosiy programma va protsedura orasidagi qiymatlar almashinuvi bir yoqlama va ikki yoqlama bo'lishi mumkin.

1) Bir yoqlama protseduralar:

Bir yoqlama protseduralarda qiymatlar faqat bir tomondan ikkinchi tomonga uzatiladi. Yani asosiy programmada protseduraga yoki protseduradan asosiy programmaga uzatiladi.

Agar asosiy programmadagi biror qiymat protsedurada ishlatilishi lozim bo'lsa, shu qiymat, qiymat parametri yordamida protseduraga uzatiladi. Ammo olingan yangi qiymatni asosiy programmaga uzatish mumkin emas.

Bir yoqlama protseduraning umumiy ko'rinishi

```
PROCEDURE Nomi(qiymat_parametrlar:Toifasi);
```

```
    E'lon qilish bo'limi;
```

```
BEGIN
```

```
    Qism dastur tanasi;
```

```
END;
```

yoki

```
PROCEDURE Nomi(VAR o'zgaruvchan_parametrlar:Toifasi);
```

```
    E'lon qilish bo'limi;
```

```
BEGIN
```

```
    Qism dastur tanasi;
```

```
END;
```

Qiymat parametrlar – asosiy dasturdan protseduraga uzatilgan qiymatlarni qabul qilib oluvchi parametrlar.

O'zgaruvchan parametrlar – protsedura bajarilishi natijasida olingan qiymatlarni asosiy dasturga uzatuvchi parametrlar. O'zgaruvchan parametrlardan oldin **VAR** xizmatchi so'zini yozish shart.

Eslatma: Qiymat parametrlari va o'zgaruvchan parametrlar toifasi, asosiy dasturdagi qiymati uzatilayotgan yoki natijani qabul qilayotgan o'zgaruvchilar toifasi bilan bir xil bo'lishi lozim.

Misol: **procedure** kattasi (a,b:integer);

Var

```
    c: integer;
```

Begin

```
    if a>b then c:=a
```

```
        else c:=b;
```

```
    Writeln('Natija=',c);
```

End;

Yoki

procedure kichigi (**Var** c:integer);

Var

```
    a,b:integer;
```

Begin

```
    Read(a);
```

```

Read(b);
if a>b then c:=b
      else c:=a;

```

End;

Bu protseduralar bir yoqlama protseduralardir. “**kattasi**” protsedurasida “a” va “b” qiymat parametrlari asosiy dasturdan mos ravishda qiymatlarni oladi. Bu qiymatlar protsedura ishida ishlatiladi. “c” o`zgaruvchisini lokal e`lon qilinganligi uchun asosiy programmaga qiymatini uzatish mumkin emas.

“**kichigi**” protsedurasida “a” va “b” lokal o`zgaruvchilariga protsedura ichida qiymat berilib, natija “c” o`zgaruvchi vositasida asosiy programmaga uzatiladi.

2) Ikki yoqlama protseduralar:

Ikki yoqlama protseduralarda ham asosiy programmadan, ham protseduradan asosiy programmaga qiymat uzatish mumkin. Yani protsedura parametrlari sifatida ham qiymat parametrlari, ham o`zgaruvchan parametr ishlatiladi.

Ikki yoqlama protseduraning umumiy lo`rinishi:

```

PROCEDURE Nomi (VAR O`zgaruvchan_parametr: Toifasi;
                  Qiymat_parametr: Toifasi);

```

```

    E`lon qilish bo`limi;

```

```

BEGIN

```

```

    Qism dastur tanasi;

```

```

END;

```

Misol: **procedure kattasi** (**Var** c:integer; a,b:integer);

```

Begin

```

```

    if a>b then c:=a

```

```

        else c:=b;

```

```

End;

```

“a” va “b” qiymat parametrlari asosiy dasturdan qiymat qabul qilib oladi. Protsedura bajarilganidan keyin, natija “c” o`zgaruvchan parametr orqali asosiy dasturga uzatiladi.

Misol : $f(x, y) = \frac{\sqrt{|x-1|}}{1 + \frac{x^2}{2} + \frac{y^2}{4}}$ ifodani f(2,2), f(2,1), f(5,4) bo`lgan hollar

uchun hisoblash talab qilinsin. Bu misolni ikki xil usulda ishlab solishtirib ko`ramiz.

Bir yoqlama protsedura	Ikki yoqlama protsedura
program bir_yoqlama; procedure hisobla(x,y:real); var f: real; begin f:=sqrt(abs(x-1))/(1+sqr(x)/2+sqr(y)/4); write('x=',x:2:2); write(' y=',y:2:2);	program ikki_yoqlama; Var ff:real; procedure hisobla(var f: real; x,y:real); begin f:=sqrt(abs(x-1))/(1+sqr(x)/2+sqr(y)/4); write('x=',x:2:2); write(' y=',y:2:2);

<pre>writeln(' f=',f:3:2); end; Begin hisobla(2,2); hisobla(2,1); hisobla(5,4); readln; end.</pre>	<pre>end; Begin hisobla(ff,2,2); writeln(' f=',ff:3:2); hisobla(ff,2,1); writeln(' f=',ff:3:2); hisobla(ff,5,4); writeln(' f=',ff:3:2); readln; end.</pre>
---	---



Nazorat savollari.

1. Parametrli protsedura deb qanday protseduralarga yatiladi?
2. Parametrli protseduralarning nechta xil turi mavjud?
3. Bir o'qqlama protseduralarni tushuntirib bering.
4. Ikki o'qqlama protseduralarni tushuntirib bering.
5. Qanday parametrlar qiymat parametrlar deyiladi?
6. Qanday parametrlar o'zgaruvchan parametrlar deyiladi?

3. Funktsiyalar

Foydalanuvchi tomonidan e'lon qilingan funktsiyalar ham standart funktsiyalar kabi bitta qiymatni hisoblash uchun ishlatiladi.

Funktsiyalar ishlatilishiga ko'ra protseduralardan bir oz farq qiladi. Misol uchun protseduradan olinadigan qiymatni to'g'ridan to'g'ri ifodalarni hisoblashda ishlatib bo'lmaydi. Lekin bu o'rinda funktsiyalardan foydalanish mumkin. Misol uchun oldingi mavzudagi programmada ekranga chiqarishni

Writeln(hisobla(ff,2,2)); ko'rinishida yozsak hatolik sodir bo'ladi.

Funktsiyaning protseduradan asosiy farqi, asosiy programmadan birdan ortiq qiymat olishidan qat'i nazar, faqat bitta bitta qiymatni qaytarishidir.

Funktsiyani e'lon qilish:

Umumiy ko'rinishi	Namuna
FUNCTION Nomi(Parametr_e'loni): Natija_toifasi; E'lon qilish bo'limi; BEGIN Funksiya tanasi; END;	Function Yuza(a,h:real):real; Begin Yuza:=a*h/2; End;

Funktsiyalar parametrli yoki parametrsiz ishlatiladi. Parametrsiz ishlatilganda, qavs ichidagi yozilgan parametr e'loni tashlab ketiladi. Funktsiyani e'lon qilishda **natija toifasini** e'lon qilish shart.

Misol: $f(x, y) = \frac{\sqrt{|x-1|}}{1 + \frac{x^2}{2} + \frac{y^2}{4}}$ ifodani $f(2,2)$, $f(2,1)$, $f(5,4)$ bo'lgan hollar uchun

hisoblash talab qilinsin. Bu misolni funksiya qism dasturdan foydalangan holda ishlab chiqamiz.

Funksiya qism dastur	Protsedura qism dastur
Program funksiya; var f: real; Function hisobla(x,y:real):real; begin hisobla:=sqrt(abs(x-1))/ (1+sqrt(x)/2+sqrt(y)/4); end ; Begin {funksiya qiymati biror o'zgaruvchiga o'zlashtiriladi} f:= hisobla (2,2); writeln(' f=',f:3:2); {yoki quyidagicha ishlatish mumkin} writeln(' f=', hisobla (2,1):3:2); writeln(' f=', hisobla (5,4):3:2); {funksiyani ifodada ishlatish mumkin} f:=hisobla(2,2)+hisobla(2,1)+hisobla(5,4); writeln(' f(2,2)+f(2,1)+f(5,4)=' ,f:3:2); readln; end .	program ikki_yoqlama; Var ff:real; procedure hisobla(var f: real; x,y:real); begin f:=sqrt(abs(x-1))/(1+sqrt(x)/2+sqrt(y)/4); write('x=',x:2:2); write(' y=',y:2:2); end ; Begin hisobla (ff,2,2); writeln(' f=',ff:3:2); hisobla (ff,2,1); writeln(' f=',ff:3:2); hisobla (ff,5,4); writeln(' f=',ff:3:2); readln ; end .

4. Rekursiya. O'z - o'zini chaqiruvchi qism dasturlar.

Pascal dasturlash tilida prosedura yoki funksiyalar o'z – o'zini chaqirish imkoniyatiga ega. Bunday qism dasturlar rekursiyali (o'z – o'zini chaqiruvchi) qism dastur deyiladi.

Rekursiyali qism dasturlarning uch hil turi mavjud.

1) Rekursiyali funksiya murojaat qilinishidan oldin bajariluvchi

Function Rekursiya;

Begin

Funksiya tanasi;

IF shart **THEN** Rekursiya;

end;

2) Rekursiyali funksiya murojaatdan keyin bajariluvchi

Function Rekursiya;

Begin

IF shart **THEN** Rekursiya;

Funksiya tanasi;

end;

3) Rekursiyali funksiya murojaatdan keyin va oldin bajariluvchi

```

Function Rekursiya;
Begin
    Funksiya tanasi S1;
    IF shart THEN Rekursiya;
    Funksiya tanasi S2;
end;

```

Rekursiyali qism dasturlarga qo`yiladigan asosiy talab, qandaydir qiymatda rekursiya yolg`on qiymat qabul qilishi kerak. Shundagina chaqirilgan funksiyalar qaytadi. Aks holda funksiya o`z – o`zini davomli ravishda chaqiradi va hatolik yuz beradi.

Tushinarli roq bo`lishi uchun  $n!$  faktorialni hisoblochi dasturni ko`rib chiqamiz. $n! = 1 * 2 * \dots * (n-1) * n$;

```

program rekursiya;
var N: integer;
function fact(i: integer):longint;
begin
    if i=0 then begin
        fact:=1; exit;
    end;
    fact:=i*fact(i-1);
end;

```

```

begin
    write('N='); readln(N);
    writeln(n, '!=', fact(n));
    readln;
end.

```

Dasturni $n=5$ uchun tahlil qilamiz.

	N=5 kiritiladi fact(5)	Rekursiyaning qaytishi $5!=120$;	
↓	$i=5$; $\text{fact}(5) := 5 * \text{fact}(4)$;	$\text{fact}(5) := 5 * 24$; (120)	↑
↓	$i=4$; $\text{fact}(4) := 4 * \text{fact}(3)$;	$\text{fact}(4) := 4 * 6$; (24)	↑
↓	$i=3$; $\text{fact}(3) := 3 * \text{fact}(2)$;	$\text{fact}(3) := 3 * 2$; (6)	↑
↓	$i=2$; $\text{fact}(2) := 2 * \text{fact}(1)$;	$\text{fact}(2) := 2 * 1$; (2)	↑
↓	$i=1$; $\text{fact}(1) := 1 * \text{fact}(0)$;	$\text{fact}(1) := 1 * 1$; (1)	↑
↓	$i=0$; $\text{fact}(0) := 1$;		↑

Fibonachchi ketma ketligining n – hadini rekursiya qism dastur orqali hisoblovchi dastur.

```

program Fibonachchi;

```

```

var N: integer;
Function fib(i: integer):longint;
begin
  if (i=1) or (i=2) then fib:=1
    else fib:=fib(i-1)+fib(i-2);
end;
begin
  write('N='); readln(N);
  writeln(Fib(n));
  readln;
end.

```



Nazorat savollari.

1. Funksiyaning protseduradan asosiy farqlari?
2. Funksiya qanday e`lon qilinadi?
3. Rekursiya nima?
4. Rekursiyali qism dasturga qo`yiladigan talablar?

Mustaqil yechish uchun masalalar:

1. 16 mavzudagi masalalarni funksiya qism dasturdan foydalangan holda eching.
2. Birinchi elementi bir bo`lgan, qolgan ixtiyoriy elementi o`zidan oldingi elementlar yig`indisiga teng bo`lgan ketma – ketlikning n – hadi qiymatini rekursiya qism dastur orqali aniqlovchi programma tuzilsin.
3. Birinchi elementi bir bo`lgan, qolgan ixtiyoriy elementi o`zidan oldingi elementlar qvadratlarining yig`indisiga teng bo`lgan ketma – ketlikning n – hadi qiymatini rekursiya qism dastur orqali aniqlovchi programma tuzilsin.
4. a haqiqiy va n butun sonlari berilgan. ($n \geq 0$). Shu a sonini n - darajaga ko`taruvchi rekursiyali qism dastur tuzing.

14-ma`ruza: Foydalanuvchining prosedura va funksiyalari

Reja:

1. Asosiy tushunchalar
2. Turbo Pascal tilidagi protseduralar
3. Turbo Pascal tilidagi funksiyalar
4. Formal va faktik parametrlar
5. Qismdasturlar buyruqlari
6. Masalalar yechish

1. Asosiy tushunchalar

Katta va murakkab dasturlarni ishlab chiqishda dasturlashga strukturali yondashuv va murakkab dastur har biri dastlabki dasturdan soddaroq ko'proq (yoki kamroq) funksional – tugallangan qismlarga ajratiladigan dasturlarni quyidan loyihalashdan foydalaniladi. Bunday dasturlarni oson sozlash va foydalanish

mumkin, chunki dastur kodining ayrim fragmentlarini yozish va ularni asosiy dasturdan alohida sozlash imkoniyati mavjud. Qismdasturlardan foydalanish bitta qism dastur mumkin turli parametrlar bilan dasturda bir necha marta foydalanish mumkin bo'lgan hollarda samarali bo'ladi. Bu kompyuter xotirasini tejashga imkon beradi.

Qismdasturlar o'z navbatida pastroq saviyadagi qismdasturlar ko'rinishida amalga oshiriladigan mayda qismlarga ajratish mumkin. Qismdasturlar ishlatiladigan dasturlar strukturalashgan deb hisoblanadi.

Ta'rif

Qismdastur – dasturning turli joylaridan ixtiyoriy sonda bajarish uchun chaqirib olish mumkin bo'lgan tilning nomlangan, mantiqiy tugallangan operatorlar guruhi.

Qismdastur tuzilishiga ko'ra dasturga o'xshash, lekin dasturdan farqi albatta sarlavhasida ko'rsatiladigan original nomga ega. Qismdasturlar ulardan foydalanuvchi dasturlarning tavsiflash bo'limida tavsiflanadi. Qism dasturga uning nomi bo'yicha murojaat qismdasturni chaqirish deb ataladi. Qismdastur chaqirilgandan so'ng darhol unga kiruvchi operatorlar bajarila boshlaydi; ularning oxirgisi bajarilgandan so'ng boshqaruv orqaga asosiy dasturga o'tadi. Qismdastur ichida lekin boshqa qismdasturlarga yoki o'z-o'ziga (rekursiv chaqiruv) murojaatlarni o'z ichiga olishi mumkin.

Dasturda foydalaniladigan ba'zi o'zgaruvchilar yordamchi bo'lib, qismdastur ichida lokallashtirilishi mumkin. Har bir qismdastur dasturning o'zi kabi tavsiflar bo'lishi va operatorlar bo'limidan iborat. Tavsiflar bo'limida qismdasturda foydalanuvchi komponentalar, o'zgaruvchilar, tiplar, past darajadagi ichma-ich qo'yilgan qismdasturlar va h.k.lar tavsiflanadi. Lekin qismdastur ichida barcha tavsiflangan komponentalar, o'zgaruvchilar va h.k.larga asosiy dasturdan kirib bo'lmaydi. Shunga o'xshash, agar yuqori darajali qismdastur past darajali qismdastur tavsifini o'z ichiga olsa, past darajali qismdasturda tavsiflangan konstanta va o'zgaruvchilarga yuqoriroq darajali qismdasturdan kirib bo'lmaydi.

Shunday qilib, nomlarni lokallashtirish prinsipi amalga oshadi. Qismdastur ichida tavsiflangan barcha obyektlar lokal hisoblanadi. Shu bilan birga ixtiyoriy qismdastur yuqori darajali qismdasturlar va asosiy dasturda tavsiflangan konstanta o'zgaruvchilardan foydalanilishi mumkin. Bunday obyektlar berilgan qismdasturga nisbatan global deb ataladi. Bunday tavsiflash qismdastur tavsifidan oldin kelishi lozim.

Lokal o'zgaruvchi nomi global o'zgaruvchi nomi bilan ustma-ust tushishi mumkin. Lekin bunda quyidagi qoidani e'tiborga olish lozim: qismdastur ichida lokal o'zgaruvchi (bu qismdastur ichida tavsiflangan) global o'zgaruvchini (asosiy dastur yoki yuqori darajadagi qismdasturda tavsiflangan) yopadi, ya'ni qismdastur ichida shu nom bilan lokal o'zgaruvchi bog'lanadi, o'sha nomdagi global o'zgaruvchiga kirib bo'lmaydi.

Turbo Pascal tilida ikkita tipdagi qismdasturlardan foydalaniladi: protseduralar (**Procedure**) va funksiyalar (**Function**).

Protseduralarning funksiyalardan asosiy farqi oxirgisining natijasi bitta yagona qiymatdan iborat. Turbo Pascal ning barcha protsedura va funksiyalarini ikki guruhga ajratish mumkin:

- Turbo Pascal tilining standart protseduralari va funksiyalari;
- foydalanuvchi aniqlagan protsedura va funksiyalar.

Birinchilari til tarkibiga kiradi va qat'iy tayinlangan nom bo'yicha bajarish uchun chaqiriladi. Ularning matni dasturchidan yashirilgan va u ularga nisbatan tayyor qismdasturlar foydalanuvchisi sifatida ish ko'radi. Ikkinchilarini foydalanuvchining o'zi ishlab chiqadi va nomlaydi. Barcha standart vositalar maxsuslashgan kutubxona modullari (SYSTEM, CRT, DOS, GRAPH, OVERLAY, PRINTER va h.k.) da joylashgan bo'lib, ular sistema nomlariga ega. SYSTEM moduli jimlik bo'yicha ishga tushadi, barcha qolganlari zaruriy paytda zahiralangan USES so'zi yordamida ulanishi mumkin:

uses Crt, Dos, Printer;

2. Turbo Pascal tilidagi protseduralar

Protsedura – bu qandaydir tugallangan amallar ketma-ketligini bajarish uchun nom bo'yicha chaqirish mumkin bo'lgan bog'liqmas nomlangan dastur qismidan iborat.

Turbo Pascal tilida protsedura tuzilishi quyidagi ko'rinishga ega:

procedure <protsedura nomi> (<formal parametrlar tavsifi>); <buyruqlar>;
{lokal belgilar, konstantalar, tiplar, o'zgaruvchilar, past darajadagi protsedura va funksiyalarni tavsiflash bloki}

begin

{protsedura tanasi}

end;

Protsedura va dastur tuzilishi juda o'xshash <protsedura nomi> – Turbo Pascal tili identifikatori protsedurani chaqirganda formal parametrlar faktli parametrlar bilan almashtiriladi. Formal parametrlar faktik parametrlar uchun joy zahiralaydi. Bu parametrlar orqali protseduraning asosiy dasturi ma'lumot almashishi yuz beradi. Formal parametrlar nomlari ixtiyoriy. Asosiy dasturdagi protseduraga murojaat qilishda faktli parametrlar soni va kelish tartibi protseduradagi formal parametrlar soni va kelish tartibiga aniq mos kelishi lozim. Aks holda protseduraga murojaat xatoga olib kelishi mumkin. Xususan, xato formal va faktik parametrlar tiplari ustma-ust tushmasligi tufayli paydo bo'lishi mumkin.

3. Turbo Pascal tilidagi funksiyalar

Funksiya protseduradan uning natijasi yagona qiymat bo'lishi bilan farq qiladi, shuning uchun funksiyaga murojaatni o'zgaruvchi va konstantalar bilan birga ifodalarda foydalanish mumkin.

Turbo Pascal tilida funksiya tuzilishi quyidagi ko'rinishga ega:

function <funksiya nomi> (<formal parametrlar tavsifi>); <natija tipi>;
<buyruqlar>;

{lokal belgilar, konstantalar, tiplar, o'zgaruvchilar, past darajadagi protsedura va funksiyalarni tavsiflash bloki}

begin

{operatorlar bo'limi}

<funksiya nomi> := <natija>;

end;

Operatorlar bo'limida hech bo'lmaganda bitta funksiya ishi natijasi sifatida qaytariladigan qiymatni uning nomiga beradigan bitta operator joylashishi lozim. Agar bunday qiymat berishlar bir nechta bo'lsa, funksiya bajarish natijasi oxirgi berish operator qiymatidan iborat bo'ladi. Agar bunday operator bo'lmasa yoki bajarilmagan bo'lsa, u holda funksiya qaytaruvchi qiymat aniqlangan bo'lib qoladi.

Funksiya protsedura kabi uning nomi bo'yicha chaqiriladi:

<funksiya nomi> (<faktli parametrlar>)

α burchak tangensini hisoblash funksiyasi berilgan bo'lsin:

function Tangent (alpha:real):real;

begin

Tangent := sin(alpha)/cos(alpha)

{alpha – burchakning radianlardagi qiymati!!! 1 rad. = $180^\circ/\pi \approx 57^\circ$ }

end;

Funksiyani quyidagicha chaqirish mumkin:

Tangent (Pi);

y := Tangent (Pi);

funksivani chaqirish

Bu yerda Pi – funksiyaning faktik parametri, u o'z navbatida π sonini qaytaruvchi Turbo Pascal tilining standart funksiyasi hisoblanadi. Shunday qilib, funksiyadan quyidagi qiymatga ega bo'lamiz:

alpha = 3.1415926535897932385.

Yana ta'kidlaymizki, protseduradan farqli funksiyaning chaqirish alohida operator ko'rinishida rasmiylashtirilmaydi (bu man qilinmasada). Funksiyaga murojaat funksiya ko'rsatgichini biror ifodada operand sifatida foydalanish yo'li bilan amalga oshiriladi. Agar ular funksiya tavsifida ko'zda tutilgan bo'lsa, funksiya ko'rsatgichi argumentlar ro'yxati bilan birgalikda funksiya nomi – faktik parametrlardan iborat. Funksiya uchun ham, protsedura uchun ham uning nomidan keluvchi formal parametrlar ro'yxati bo'lishi shart emas.

4. Formal va faktik parametrlar

Qismdasturning formal parametrlar bu qismdasturga qanday parametrlar bilan murojaat qilish lozimligini ko'rsatadi (parametrlar soni, ularning ketma-ketligi, tiplari). Yuqorida ko'rsatilganidek, formal parametrlar nuqtali vergullar bilan ajratilgan guruhlariga bo'lingan parametrlar ro'yxati ko'rinishida qismdastur sarlavhasida beriladi. Formal parametrlar guruhiga bir kadegoriyadagi bir tipdagi parametrlar kiritiladi.

Barcha formal parametrlar to'rtta kategoriyaga ajratish mumkin:

- parametr – qiymatlar (bu parametrlar asosiy dasturda qismdastur bilan o'zgartirilmaydi);
- parametr – o'zgaruvchilar (bu parametrlarni qismdastur asosiy dasturda o'zgartirishi mumkin);
- parametr – konstantalar;
- parametr – protseduralar va parametr – funksiyalar (ya'ni protsedura tipidagi funksiyalar).

Har bir formal parametr uchun nom va odatda tip, parametr–o'zgaruvchi yoki parametr–konstanta holida uning kategoriyasini ko'rsatish lozim.

Formal parametr tipi amalda ixtiyoriy bo'lishi mumkin, lekin qismdastur sarlavhasida yangi tipni kiritishi mumkin emas. Masalan:

function Max(A: array[1..100] of real) : real;

deb yozish mumkin emas.

Bu sarlavhani to'g'ri yozish uchun asosiy dasturda tip-massivni kiritish, keyin uni sarlavhada foydalanish lozim:

type tArr = **array**[1..100] of real;

function Max (A: tArr) :real;

Qismdasturga murojaat qilganda formal parametrlar mos chaqiriluvchi dastur yoki qismdastur faktik parametrlar bilan almashinadi.

4.1. Parametr-qiymatlar

Tavsifi:

<parametr nomi> : <parametr tipi>

Protseduraga faktik parametr nusxasi uzatiladi, protsedura bajarilishida faqat bu nusxa o'zgaradi. Uning o'zgarishi faktik parametr qiymatiga hech qanday ta'sir etmaydi. Bu parametrni qiymati bo'yicha uzatish deb ataladi. Protsedurani chaqirishda parametr-qiymatga faktik parametr – ko'rsatilgan tipdagi ifoda mos kelishi lozim. Parametr-qiymat tipi fayl tipidan tashqari ixtiyoriy bo'lishi mumkin.

Bir tipdagi parametr-qiymatlar bir nechta bo'lsa, ularni bir guruhga birlashtirish, ularning nomlarini vergullar bilan ajratib sanash, keyin umumiy tipni ko'rsatish mumkin. Parametrlar ayrim guruhlari bir-biridan nuqtali vergul bilan ajratiladi, masalan:

procedure Inp(max, min:real; n:wor);

function Mult(x, y:integer) :real;

Faktik parametr sifatida qismdasturni chaqirganda parametr-qiymat o'rniga fayl komponentni o'z ichiga olmagan qiymat berish uchun muvofiq tipdagi ixtiyoriy ifoda qo'yilishi mumkin, masalan:

Inp(abs(z), -abs(t), 2*k);

M:= Mult(x+y, x-y);

M:= Mult(7, a **div** 3);

4.2. Parametr-o'zgaruvchilar

Faktik parametr qiymati protsedurani bajarish jarayonida o'zgarish uchun uning tavsifi oldidan amal qilishi eng yaqin nuqtali vergulgacha, ya'ni bir guruh chegarasigacha mumkin bo'lgan xizmat so'zi var ishlatilishi lozim. Bunday parametr o'zgaruvchi deb qaraladi.

Tavsifi:

var <parametr nomi>: <parametr tipi>

Protseduraga faktik parametr manzili uzatiladi, shuning uchun mos protsedura ichida formal parametr o'zgarishi faktik parametrni ham o'zgaradi. Bu murojaat bo'yicha parametr-o'zgarishi deb ataladi (faktik parametr manziliga murojaat nazarda tutiladi). Parametr-o'zgaruvchiga protsedura chaqirilganda faktik parametr-ko'rsatilgan tipdagi o'zgaruvchi mos kelishi lozim. Parametr-o'zgaruvchilar, shu jumladan fayl ixtiyoriy bo'lishi mumkin.

4.3. Parametr-konstantalar

Tavsifi:

const <parametr nomi>: <parametr tipi>

Protseduraga faktik parametr manzili uzatiladi, lekin bunday parametrga protsedurani bajarish jarayonida yangi qiymat berish mumkin emas (bu kompilyator tomonidan to'sib qo'yiladi). Parametr-konstantaga protsedurani chaqirganda faktik parametr – ko'rsatilgan tipdagi ifoda mos kelishi lozim.

Asosiy dasturga protsedura ishi natijalari uzatiladigan protsedura barcha chiqish parametrlari parametr-o'zgaruvchilar kabi tavsiflanishi lozim. Ikkinchi tomondan, parametr qiymatlar yoki konstantalardan foydalanish faktik parametrlar – ifodalar bilan ishlashga imkon beradi, hamda protsedura ishlash paytida faktik parametr qiymatini tasodifiy o'zgartirish mumkin emasligini kafolatlaydi, bu esa dasturning ishonchliligini oshiradi. Parametr-qiymatlardan foydalanish xotirani qo'shimcha sarf etishni talab etadi (chunki har bir parametr-qiymat uchun xotirada qo'shimcha o'rin ajratadi).

Eslatma

Parametr-konstantalardan foydalanish faqat Turbo Pascalning 7.0 talqinida mumkin.

Bir tipdagi vazifalarni parametrlar orqali sanab o'tish mumkin. Turli tipdagi va turlicha vazifalarni parametrlar tavsifi vergul bilan ajratiladi, masalan:

procedure elephant (const c123: integer;

i, j: integer;

a, b, c: real;

var f1: real;

Bu yerda c123 – parametr konstanta, i va j – butun tipdagi parametr – qiymatlar, a, b va c – haqiqiy tipdagi parametr – qiymatlar, f1 – haqiqiy tipdagi parametr-o'zgaruvchi.

4.4. Tipsiz parametrlar

Turbo Pascal da tipini ko'rsatmasdan parametr-o'zgaruvchi va parametr konstantalardan foydalanish mumkin. Bu holda faktik parametr ixtiyoriy tipdagi o'zgaruvchi bo'lishi mumkin, u yoki bu parametrning to'g'ri qo'llanishiga ma'suliyat dasturchi zimmasiga tushadi, masalan:

procedure Shark (var par1, par2; len: word) :real;

Bu yerda par1, par2 – tipsiz parametr – o'zgaruvchilar (ular o'rniga masalan, oddiy tipdagi, tip-massiv, tip-yozuv va h.k.lar ixtiyoriy o'zgaruvchilarni foydalanish mumkin); len – parametr-qiymat.

Tipsiz parametr dastur ichida tipga ega emas va undan foydalanish oldidan konkret tipga mos tipdagi identifikatorlarni qo'llab almashtirish lozim, bunda olingan natija ixtiyoriy o'lchovda bo'lishi mumkin.

Eslatma

Qator hollarda bir tipdagi o'zgaruvchini boshqa tipdagi o'zgaruvchiga almashtirish talab e'tiladi. Turbo Pascalda bunday almashtirish nisbatan oddiy amalga oshiriladi. Buning uchun tipning identifikatorini, so'ngra uning orqasidan doira qavslarda berilgan tipdagi o'zgaruvchini ko'rsatish lozim.

4.5. Massivlar va ochiq tipdagi satrlar

Turbo Pascal 7.0 da parametr-o'zgaruvchilar sifatida o'lchovlari berilmagan bir o'lchovli massivlar va ochiq tipdagi satrlardan foydalanish mumkin. Faktik parametr sifatida bu holda massiv yoki ixtiyoriy o'lchovdagi satrdan foydalanish mumkin, lekin massiv ochiq massiv komponentlari kabi komponentlaridan iborat bo'lishi lozim. Bunday parametrlar qismdastur massiv yoki ixtiyoriy o'lchamdagi satrni qayta ishlay olishi uchun ajratilgan. Faktik parametr bu holda High (1-ilova) funksiyasi yordamida aniqlanishi mumkin.

Massivda maksimal elementni hisoblash funksiyasini qaraymiz. Uzatiluvchi parametr sifatida ochiq tipdagi massivdan foydalanamiz:

procedure Max (var Mas: **array of integer**):integer;

var N:integer;

 i:byte;

begin

 N:=Mas[0];

 for i:=1 to High(Mas) do

 if N < Mas[i] then N:=Mas[i]

 Max:=N

end;

Bu misolda qismdasturga faqat bitta parametr uzatiladi va u butun sonlarning bir o'lchovli ixtiyoriy massivi bilan ishlay oladi. Lekin ochiq massiv uchun qismdastur ishlayotganda stekda uning nusxasi yaratiladi, bu esa stekning to'lishiga olib kelishi mumkin.

Ochiq massiv boshqacha ko'rinishi – ochiq satr bo'lib, OpenString standart tipi yoki String tipi yordamida va {\$S+} kompilyator kalitidan foydalanib berilishi mumkin. Masalan, satrni qandaydir simvol bilan to'ldiruvchi protsedura sarlavhasi

procedure FillChar (var Str: OpenString, Ch:Char);
yoki
{\$S+}
procedure FillChar (var Str: string, Ch:Char);
ko'rinishga ega bo'lishi mumkin.

4.6. Parametr – protseduralar va parametr – funksiyalar

Uzatiluvchi parametr parametr–protsedura yoki parametr–funksiya, ya'ni protsedura tipidagi parametr bo'lishi mumkin. Bu parametr parametr–qiymat hisoblanadi, ya'ni **var** xizmat so'zisiz yoziladi.

Bu holda faktik parametr sifatida talab qilingan tipdagi parametrlarning zarur soniga ega bo'lgan mos protsedura yoki funksiyadan foydalaniladi.

Parametr–protsedura va parametr–funksiyalar uchun boshqa protsedura tipidagi o'zgaruvchilar uchun bo'lgani kabi qoidalar mavjud: qismdasturlar {\$F+} kalit bilan kompilyatsiyalanishi lozim yoki far buyrug'iga ega bo'lishi lozim, Turbo Pascal tilining standart qismdasturlar ichida e'lon qilmasligi lozim, inline yoki interrupt buyruqlarga ega bo'lishi lozim.

5. Qismdasturlar buyruqlari

Qismdastur sarlavhasidan keyin buyruqlar kelishi mumkin. Buyruqlar qismdasturni joylashtirish haqida qo'shimcha axborotni translayatorga beradi. Turbo Pascal tilining standart buyruqlari **assembler**, **external**, **far**, **near**, **forward**, **inline**, **interrupt** bo'lib hisoblanadi. Masalan, **assembler** buyruqdan foydalanish qismdastur Turbo Pascal til kompilyator ichiga qurilgan assembler buyruqlar ketma-ketligidan iborat ekanligini ko'rsatadi.

far buyrug'i xotiraning uzoq modelini aniqlaydi, ya'ni bu buyruq bilan qo'shib olib beriluvchi dasturning ixtiyoriy qismidagi protsedura va funksiyalarga murojaat qilish imkoniyatini ta'minlaydi. Bu buyruqning ko'rsatish zarur, masalan, prosedura tiplaridan foydalanganda. Aksincha, **near** buyrug'i xotiraning uzoq modelini aniqlaydi.

Eslatma

ShK mikroprotsessor tuzilishiga ko'ra dasturlarda xotiraning ikkita modeli: yaqin va uzoq modellari foydalanilishi mumkin. Xotira modeli dasturning turli qismlaridan qismdasturni chaqirish imkoniyatini aniqlaydi: agar yaqin model foydalanilayotgan bo'lsa, chaqiruv faqat 64 Kbayt chegarasida (asosiy dastur va unda foydalanilayotgan har bir modelga ajratilayotgan kodning bitta segmenti chegarasida); uzoq modelda ixtiyoriy segmentdan chaqiruv mumkin. Yaqin model bir bayt va qismdasturni chaqirganda bir necha mikrosekund tejaydi, shuning uchun kompilyatsiyaning standart rejimi bu xotira modelini ko'zda tutadi.

forward buyrug'i qismdasturni oldinlab ketuvchi tavsifida ishlatiladi, masalan, qismdasturlardan biri ikkinchi qismdasturga murojaatni ikkinchisi o'z navbatida birinchisiga murojaatni o'z ichiga olganda shu hol sodir bo'ladi. Har bir

qismdastur unga murojaat qilunguncha tavsiflanishi lozim bo'lgani uchun ikkinchi qismdastur sarlavhasini **forward** buyrug'i bilan birinchi qismdastur tavsifini berishdan oldin e'lon qilish zarur.

Masalan, A nomli protsedura B nomli protsedurani chaqiradi, o'z navbatida B protsedura A protsedurani chaqiradi.

{A protseduraning ildamlovchi tavsifi}

procedure A(y:<o'zgaruvchi tipi>); **forward**;

{B protsedura sarlavhasi}

procedure B(x:<o'zgaruvchi tipi>);

{B protseduraning tavsiflar bo'limi}

begin

A(p); {B dan A protsedurani chaqirish}

end;

procedure A; {A protseduraning asosiy tavsifi}

{A protseduraning tavsiflar bo'limi}

begin

B(g); {A dan B protsedurani chaqirish}

end;

forward buyrug'i protsedura tavsifi dastur matnidan keyin joylashtiriladi, lekin unga hozirdanoq murojaat qilish mumkin (ikkita protsedura tavsifidan oldin keluvchi birinchi protsedura tavsifidan).

6. Masalalar yechish

1.1 – Misol.

Dasturda $\sin(x)$ funksiya qiymatlarini berilgan aniqlikda darajali qator yig'indisi orqali hisoblovchi protsedura va funksiyalardan foydalanishni tavsiflaymiz.

$\sin(x)$ trigonometrik funksiya yoyilmasidan iborat

$$x - \frac{x^3}{3!} + \frac{x^5}{5!} - \dots + (-1)^N \cdot \frac{x^{2 \cdot N + 1}}{(2 \cdot N + 1)!} + \dots$$

darajali qator xadlarini yig'indisini hisoblash

dasturini yozish talab etiladi. Hisoblashlarni alohida protsedura va funksiya ko'rinishida tashkil etish zarur. Natijalarni Turbo Pascal tilining standart $\sin(x)$ funksiyasi yordamida olingan qiymat bilan taqqoslash lozim.

Kiruvchi ma'lumotlar: Ikkita haqiqiy epsilon sonlar (hisoblash aniqligi) va x (argument qiymati).

Chiqish ma'lumotlari: Foydalanuvchi yozgan protsedura va funksiya va Turbo Pascal tilining $\sin(x)$ standart funksiyasi yordamida olingan darajasini qator hadlari yig'indilari.

Kiruvchi va chiquvchi ma'lumotlarga misollar:

Kiruvchi ma'lumotlar	Chiquvchi ma'lumotlar
$\epsilon = 0.01$ $x=0$	0.00000000 – protsedura ishi natijasi 0.00000000 – funksiya ishi natijasi 0.00000000 – standar funksiya ishi natijasi

$\epsilon = 0.01$ $x=2$	0.90934744 – protsedura ishi natijasi 0.90934744 – funksiya ishi natijasi 0.90929742 – standar funksiya ishi natijasi
----------------------------	---

Yechim

Bizning misolda asosiy dasturda quyidagi o'zgaruvchilarni tavsiflash yetarli: ϵ – hisoblashning berilgan aniqligi; x – funksiya argumenti; $\sin$ – funksiya hisoblash natijasi. Bu o'zgaruvchilar $\sin$ funksiyaning hisoblash protsedurasiga nisbatan global bo'ladi. Yordamchi o'zgaruvchilar lokal bo'ladi (protsedura ichida lokallashtirilgan). $\sin$ qator yig'indisini topish protsedurasida quyidagi formal parametrlar foydalaniladi: ϵ_1 – hisoblashlarning berilgan aniqligi; x_1 – funksiya argumenti. Bu parametrlar protsedurani bajarishda o'zgaruvchi shuning uchun qiymat bo'yicha uzatiladi (parametr-qiymatlar). s_1 – hisoblash natijasi; u asosiy dasturga uzatilishi lozim va shuning uchun parametr-o'zgaruvchi kabi tavsiflanadi. Prinsip jihatidan formal parametrlar uchun ϵ , x , $\sin$ nomlaridan foydalanish mumkin, chunki bunday holda bu tushunmovchilikka olib kelmasligi lozim. Protseura ishi natijasi bo'lib yagona s_1 qiymat hisoblanadi. Shuning uchun bu holda funksiya foydalanish mumkin. Bunday funksiya dasturda keyinroq $\sin_2$ nom bilan tavsiflangan. Quyidagi formal parametrlardan foydalaniladi: ϵ_2 – hisoblashning berilgan aniqligi; x_2 – funksiya argumenti. Funksiya qiymati extended tipga ega (o'zgarish oralig'i $3,4 \cdot 10^{-4932}$ dan $1,1 \cdot 10^{-4932}$).

Eslatma

single, double, extended, com tiplardan dasturda samarali foydalanish faqat $\{ \$N+ \}$ qo'yilgan buyruqda 8087 hamprotsessori mavjud bo'lganda samarali bo'ladi. U jimlik bo'yicha $\{ \$N- \}$ o'chirilgan holatda bo'ladi.

Dasturning berilgan matni 1.1-listingda keltirilgan.

1.1-Listing. Darajali qator hadlari yig'indisini hisoblash dasturi

```

{ $N+ }
program P1_1;
Uses Crt;
{ Global o'zgaruvchilar tavsifi }
var epsilon, x, sinus: extended;
    procedure sin1(e1, x1:extended; var s1:extended);
        { Lokal o'zgaruvchilar tavsifi }
        var s, sqrx:extended;
            n:byte;
    begin
        { sin1 protsedurasi operatorlar bo'limi }
        s:= x1; { qatorning birinchi hadi }
        n:= 1; { qator hadi darajasi }
        s1:= 0; { qator yig'indisi }

```

```

    sqr x := sqr(x1);

while abs(s) > e1 do
    begin
        s1:= s1 + s;
        n:= n + 2;
        s:= -s*sqr x/((n-1)*n);
    end
end;
Function sin2(e2, x2:extended):extended;
{Lokal o'zgaruvchilar tavsifi}
var s, sqr x, s2:extended;
        n:byte;
begin
    {sin2 protsedurasi operatorlar bo'limi}
    s:= x2; {qatorning birinchi hadi}
    n:= 1; {qator hadi darajasi}
    s2:= 0; {qator yig'indisi}
    sqr x := sqr(x2);
while abs(s) > e2 do
    begin
        s2:= s2 + s;
        n:= n + 2;
        s:= -s*sqr x/((n-1)*n);
    end;
    {sin2 funksiyaning s2 hisoblangan qiymatini berish}
    sin2:=s2
end;
begin
    clrscr; {ekranni tozalash}
    {clrscr – CRT modulda tavsiflangan Turbo Pascal standart protsedurasi}

    writeln('SIN(X) funksiyani hisoblash:');
    write('Berilgan aniqlikni kiriting: epsilon=');
    readln(epsilon);
    write('funksiya argumentini kiriting: x=');
    readln(x);
    {sin1 protsedurani chaqirish}
    sin1(epsilon, x, sinus);
    {sin1 protseduraga murojaat qilganda e1, x1, s1 formal parametrlar epsilon,
        x, sinus formal parametrlar bilan almashadi}

    write(sinus:20:18, ' – ');
    writeln('protsedura ishi natijasi');

```

```

write(sin2(epsilon, x):20:18, ' - ');
{Bu yerda funksiyaga murojaat writeln operatori ichida ro'y beradi, bu
  mumkin, chunki bu operator argumenti ifoda hisoblanadi}
writeln(' funksiya ishi natijasi');
write(sin(x):20:18, ' - ');
writeln(' standart funksiya ishi natijasi');
readkey; {bosilgan tugma kodini o'qish}
{readkey:char – CRT modulda tavsiflangan Turbo Pascal til standart
funksiyasi}
end.

```

write, writeln, read, readln operatorlar haqiqatda Turbo Pascal tilining kirish–chiqish standart protseduralarini chaqirishni amalga oshiradi. Biz foydalanadigan Crt(clrscr va readkey) modul standart qismdasturlari kabi bu protseduralar dasturchi yaratayotgan protsedura va funksiyalardan farqli oldindan tavsiflashga muhtoj emas. Kirish–chiqish standart protseduralari System moduli tarkibiga kiradi. System moduli Crt modulidan farqli dasturga avtomatik ulanadi, shuning uchun uning Uses bo'limida ko'rsatilmaydi.

1.2 – Misol.

Funksiyalardan foydalanish.

Length() standart funksiyaga o'xshash satr o'zgaruvchi uzunligini aniqlash funksiyasidan foydalanib dastur yozish va unga LengthMy() deb nom berish talab etiladi. Natija to'g'riligini nazorat qilish uchun Length() standart funksiyasidan foydalanib satr uzunligini aniqlang.

Kiritish ma'lumotlari: Str, String tipi.

Chiqish ma'lumotlari: L1 va L2, Integer tipi. L1 – LengthMy() funksiya ishi natijasi, L2 – Length() funksiya ishi natijasi.

Kirish va chiqish ma'lumotlariga misollar:

Kiruvchi ma'lumotlar	Chiquvchi ma'lumotlar
Rossiya	6 6
Ona_ramkali_novda	14 14

Yechish

Berilgan masalani yechish uchun satrda eng birinchi bayt (String tipidagi o'zgaruvchi) 0 indeksga egaligi va satrning joriy uzunligini o'z ichiga olganligini hisobga olamiz. Shunday qilib Ord(Satr[0]) qiymat joriy satr uzunligini aniqlaydi.

Dasturning berilgan matni 1.2-listingda keltirilgan.

1.2 – Listing. Length() funksiyaga o'xshash funksiyadan foydalanib, satr o'zgaruvchi uzunligini aniqlash.

```

program p1_2;
var
  Satr:string;

```

```

    k, k1 :byte;

function length1(s:string):byte;
begin
    length1:= Ord(Satr[0]);
end;
procedure readdate; {formal parametrlar yo'q}
begin
    readln(Satr);
end;
procedure work; {formal parametrlar yo'q}
begin
    k:=length(Satr); k1:=length1(Satr);
end;
procedure outdate; {formal parametrlar yo'q}
begin
    writeln(k, ' ', k1);
end;
begin
    {asosiy dstur}
    indate;
    work;
    outdate;
end.

```

Mustaqil o'rganish uchun mavzular

1. Dasturlarda protsedura tipidagi o'zgaruvchilardan foydalanish.
2. Notipik parametr–o'zgaruvchilar. Qismdasturga o'zgaruvchi uzunlikka ega bir o'lchovli massivlarni uzatish.

Nazorat savollari

1. Qismdasturlar nima uchun ishlatiladi? Qismdasturlar katta dasturlarni yozishda qanday rol o'ynaydi? Qismdasturni ta'rifini bering.
2. Funksiya protseduradan nima bilan farq qiladi? Turbo Pascalda protsedura va funksiya tuzilmasini tavsiflang. Qanday hollarda protseduradan, qaysi hollarda funkseyadan foydalanish mumkin? Protsedura matematik yoki mantiqiy ifoda operandi bo'lishi mumkinmi?
3. Qismdasturning qaysi parametrlari formal, qaysilari faktik deb ataladi? Formal parametrlar nomlari faktik parametrlar nomi bilan ustma-ust tushishi mumkinmi? Formal parametrni ta'riflashning qanday usullari sizga ma'lum?
4. Dasturda qaysi obyektlar lokal, qaysilari global bo'ladi? Lokal o'zgaruvchi nomi global o'zgaruvchi nomi bilan ustma-ust tushishi mumkinmi?

5. Xotiraning qaysi modellari dasturlarda ishlatilishi mumkin? Xotiraning qanday modeli asosiy dasturning kodi ixtiyoriy segmentidan qismdasturni chaqirishga imkon beradi?
6. Qaysi buyruqlar Turbo Pascalda qismdasturlarni yozishda foydalanishi mumkin? Qaysi hollarda u yoki bu buyruq foydalanishi mumkin?

Topshiriqlar

1. $m \times n$ ($4 \leq m, n \leq 20$) o'lchovli to'g'ri burchakli matritsa har bir satri maksimal elementlari ko'paytmasini hisoblovchi funksiyani yozing. Berilgan funksiya yordamida asosiy dasturda uchta massiv uchun ko'rsatilgan qiymatlarni toping va ularni kamayish tartibida chiquvchi *output.txt* faylga chiqaring. Berilgan ma'lumotlar *input.txt* faylidan kiritiladi.
2. Matnli axborotni o'z ichiga oluvchi faylni shifrlash uchun mo'ljallangan dasturni yozing. Shifrlash uchun mo'ljallangan fayl nomi ishga tushirishda dasturga parametr sifatida uzatilishi mumkin. Shifrlashni ochish algoritmini aniqlang.

15-ma'ruza: Paskalda ma'lumotlarning faylli turlari

Reja:

1. Toifalangan fayllar
2. Matnli fayllar
3. Toifalanmagan fayllar

1. Toifalangan fayllar

Programma ishlashi natijasida olingan ma'lumotlarni, saqlab qo'yish uchun, egiluvchan disklar, qattiq disklar va boshqa har xil tashqi qurilmalardan foydalaniladi. Ma'lumotlarni saqlab qo'yish uchun tashqi qurilmalardan foydalanish qulay va ishonchlidir.

Ma'lumotlarni saqlab qo'yish uchun, tashqi xotiraning nomlangan qismiga fayl deyiladi. Bunday fayllar fizik fayllar deyiladi.

Fizik fayllar bilan ishlash uchun, programmashtirish tillarida maxsus strukturalashgan, toifalangan fayllar kiritilgan. Bunday fayllar mantiqiy (логический) fayllar deyiladi. Mantiqiy fayllar, hech qanday fizik xotirani band qilmasdan ma'lumotlarning mantiqiy modelini o'zida saqlaydi.

Toifalangan fayl, bir xil toifadagi ma'lumotlarning cheklanmagan miqdoridir. Fayl bir nechta elementdan tashkil topgan bo'lganligi uchun, faqat fayl ko'rsatkichi ko'rsatayotgan elementga murojaat qilish mumkin. Toifalangan faylni e'lon qilish uchun **file** xizmatchi so'zidan foydalaniladi.

Var a: File of Byte; {a fayli baytlar ketma ketligidan tashkil topgan }

b: File of Integer; {b fayli butun sonlar ketma ketligidan tashkil topgan }

c: File of String; {c fayli satrlar ketma ketligidan tashkil topgan }

d: File of Char; {d fayli belgilar ketma ketligidan tashkil topgan }

Fayl uch xil xolatda bo'lishi mumkin.

1. Yopiq xolatda.

2. Yozish uchun ochilgan xolatda.
 3. O`qish uchun ochilgan xolatda.
- Fayllar bilan ishlovchi protseduralar:

1. **Assign**(f; name: string);

assign protsedurasi faylli o`zgaruvchini ( mantiqiy faylni), diskdagi fayl (fizik fayl) bilan bog`lash uchun xizmat qiladi.

f – faylli o`zgaruvchi (identifikator);

name – fizik fayl saqlangan yo`lni ko`rsatuvchi satr;

D:\TP\tiko.dat

Agar faqat fayl nomi berilgan bo`lsa, fayl joriy katalogdan qidiriladi.

2. **Rewrite**(f) – protsedurasi yangi fayl xosil qiladi, yoki bor faylni qayta yozish uchun ochadi va fayl ko`rsatkichi fayl boshiga o`rnatadi. Agar fayl bo`lsa, uning oldingi ma`lumotlari o`chiriladi. Agar fayl yo`q bo`lsa xosil qilinadi. Bu protsedura bajarilishidan oldin faylli o`zgaruvchi **assign** protsedurasi orqali tashqi fizik fayl bilan bog`langan bo`lishi shart. Bu protsedura bajarilgandan so`ng fayl yozish uchun ochiq, o`qish uchun yopiq holatda bo`ladi.
3. **Reset** (f) - protsedurasi faylni o`qish uchun ochib, fayl ko`rsatkichini birinchi elementga o`rnatadi. Bu protsedura bajarilishidan oldin faylli o`zgaruvchi **assign** protsedurasi orqali tashqi fizik fayl bilan bog`langan bo`lishi va fayl albatta mavjud bo`lishi lozim. Bu protsedura bajarilgandan so`ng fayl yozish uchun yopiq, o`qish uchun ochiq holatda bo`ladi.
4. **Close** (f) – ochiq faylni yopish. Bu protsedura bajarilgandan so`ng fayl o`qish uchun xam, yozish uchun xam yopiq xolatga o`tadi. Dastur oxirida fayllarni yopish lozim. Aks holda fatolik sodir bo`lishi mumkin.
5. **Eof** (f): **Boolean**; - funksiyasi fayl oxirini aniqlash uchun ishlatiladi. Bu funksiya, fayl ko`rsatkichi oxirgi elementga o`rnatilganda **true** qiymat qaytaradi, aks holda **false**.
6. **Read**(f, s1, [s2, s3,...s_n]); bu protsedura fayl ko`rsatkichi o`rnatilgan elementning qiymatini o`zgaruvchiga o`qish uchun xizmat qiladi. s1 faylning toifasi bilan bir xil toifadagi o`zgaruvchi. Bu protsedura bajarilgandan so`ng, fayl ko`rsatkichi keyingi elementga o`rnatiladi.
7. **Write**(f, s1, [s2, s3,...s_n]); bu protsedura o`zgaruvchilar qiymatini faylga yozish uchun xizmat qiladi. s1 faylning toifasi bilan bir xil toifadagi o`zgaruvchi. Bu protsedura bajarilganda, o`zgaruvchi qiymati fayl ko`rsatkichi o`rnatilgan joyga yozadi va fayl korsatkichi bir pozitsiya suriladi.

Toifalangan fayllar bilan ishlovchi protsedura va funksiyalar

Toifalangan fayllar elementlari noldan boshlab indexlanadi. Bu esa faylning istalgan elementiga, hohlagan vaqtda murojaat qilish imkoniyatini beradi.

1. **Seek**(f, n); protsedurasi fayl ko`rsatkichini n – elementga o`rnatadi.
2. **FileSize**(f):**LongInt**; - funksiyasi fayldagi elementlar sonini aniqlaydi.

3. **FilePos(f):Integer**; - funksiyasi joriy elementning fayldagi o`rnini aniqlab beradi. (fayl ko`rsatkichi qaysi elementda turganligini ko`rsatadi)
4. **Truncate(f)**; - fayl ko`rsatkichidan keyin fayl yakuni belgisini qo`yadi. Bunda fayl ko`rsatkichidan keyingi elementlar o`chirib tashlanadi.

Misol: n natural soni berilgan. Elementlari n ta butun sondan iborat bo`lgan fayl elementlarini kirituvchi va ekranga chiqaruvchi programma tuzilsin.

```

program fayl_namuna;
Var f: File Of Integer;
  {butun sonlardan tashkil topgan faylni e`lon qilish}
  i,n,k: integer;
begin
  assign(f, 'namuna.dat');
  Rewrite(f);
  { faylni yozish uchun ochish}
  write('n='); readln(n);
  writeln('Fayl elementlarini kiriting');
  for i:=1 to n do
    begin
      readln(k); { klaviatura orqali son kiritish}
      write(f,k); { va uni `f` fayliga yozish}
    end;
    close(f); {faylni yopish}
    Reset(f); {faylni o`qish uchun ochish}
    while not eof(f) do
      begin
        read(f,k); {faylning keyingi elementini o`qish}
        write(k, ' '); { va uni ekranga chiqarish}
      end;
      close(f); {faylni yopish}
      readln;
    end.

```



Nazorat savollari.

1. Fizik fayl deb qanday fayllarga aytiladi?
2. Mantiqiy fayl deb qanday fayllarga aytiladi?
3. Fizik va mantiqiy fayllar qanday bog`laniladi?
4. Fayllar nima uchun va qanday ochiladi?
5. Fayl qanday yopiladi?
6. Fayl ko`rsatkichi deb nimaga aytiladi?

Mustaqil yechish uchun masalalar:

1. Elementlari butun sonlardan iborat bo'lgan f fayl berilgan. Quyidagilarni hisoblovchi programma tuzilsin. Faylning elementlari soni n berilgan.
 - a) fayl elementlari yig'indisini;
 - b) fayl elementlari ko'paytmasini;
 - c) fayl elementlari kvadratlarining yig'indisini;
 - d) faylning oxirgi elementi aniqlansin;
2. Faylning elementlari soni n berilgan. Elementlari butun sonlardan iborat bo'lgan f fayl hosil qiling. Quyidagilarni aniqlovchi programma tuzilsin.
 - a) fayl elementlarining kattasi;
 - b) fayl elementlarining juft o'rindagilarining kichigi;
 - c) faylning eng katta va eng kichik elementlari yig'indisi;
 - d) faylning birinchi va oxirgi elementlari farqi;
3. \$ belgisi bilan tugaydigan simvolli (belgili) fayl f berilgan. f fayl elementlarini g fayliga teskari tartibda joylashtiruvchi programma tuzilsin.
4. f fayl berilgan. (m21_2.dat faylidan foydalaning). f fayl elementlarining juft qiymatlilaridan g faylini, toq qiymatlilaridan h haylini hosil qiling. f, g, h fayllarini ekranga chiqaring.
5. n ta elementdan iborat bo'lgan simvolli fayl berilgan. Shu faylda eng ko'p uchraydigan belgini va shu belgi sonini aniqlovchi programma tuzilsin.
6. n natural soni berilgan. Elementlari butun sonlardan iborat bo'lgan f fayl hosil qiling. f fayl elementlaridan tub bo'lganlaridan g faylini hosil qiling. f, g fayllarini ekranga chiqaring.

2. Matnli fayllar

Pascalda toifalangan fayllardan tashqari, matnlar bilan ishlash uchun matnli fayl toifasi kiritilgan. Matnli fayllarni e'lon qilishda **Text** xizmatchi so'zidan foydalaniladi.

Var t: Text;

Matnli fayllar bilan ishlash toifalangan fayllar bilan ishlashdan bir oz farq qiladi. Matnli fayllarda ma'lumotlar satrlarda saqlanadi. Matnli fayl elementlari har xil uzunlikdagi satrlardir. Bu satrlar bir biridan satr oxiri belgisi bilan ajratiladi. Matnli fayl elementlari indekslanmagan bo'lganligi uchun, faylning istalgan elementiga bevosita murojaat qilib bo'lmaydi.

Matnli fayllar bilan ishlovchi protsedura va funksiyalar:

1. **Append(f)** – mavjud matnli faylning oxiriga ma'lumot qo'shish uchun ochadi.
2. **Readln(f, s)** – fayldan satrlar qatorini o'qish protsedurasi. Bu protsedura bajarilganda fayl ko'rsatkichi o'rnatilgan qator, qator oxiri belgisi yordamida ajratilib, **string** toifasidagi o'zgaruvchilarga o'zlashtiriladi.
3. **Writeln(f, s)** – bu protsedura faylga ma'lumotni yozib satr oxiri belgisini o'rnatadi.
4. **Eoln (f)** – Bu funksiya satr oxirini aniqlash uchun ishlatiladi. Agar fayl ko'rsatkishi satr oxirida bo'lsa, **true** qiymat qaytaradi. Aks holda **false** qiymat qaytaradi.

5. **SeekEoln**(f) – funksiyasi satr oxirini aniqlash uchun ishlatiladi. Bu funksiya bajarilganda fayl ko`rsatkishi satrning keyingi elementiga o`rtaniladi. Agar satr oxiri bo`lsa **true**, aks holda **false** qiymat qaytaradi.
6. **SeekEof**(f) – funksiyasi fayl oxirini aniqlash uchun ishlatiladi. Bu funksiya bajarilganda fayl ko`rsatkishi keyingi elementga o`rtaniladi. Agar fayl oxiri bo`lsa **true**, aks holda **false** qiymat qaytaradi.

3. Toifalanmagan fayllar

Pascalda toifalanmagan fayllarni hosil qilish mumkin. Yani fayl elementi sifatida turli toifalardan foydalanish mumkin. Toifalanmaga fayllarni e`lon qilish quyidagicha bo`ladi:

Var <Fayl_o`zgaruvchisi> : **File**;

Toifalanmagan fayllarda quyidagi funksiya va protseduralardan foydalaniladi:

1. **Reset**(F, S); - protsedurasi faylni o`qish uchun ochadi. Bu erda F – fayl o`zgaruvchisi, S – har bir blok uchun belgilangan xotira xajmi.
2. **BlokRead**(F,V, N); - protsedurasi bajarilganda F va V da joylashgan S uzunlikdagi N bloklar o`zlashtiriladi. N o`qilishi lozim bo`lgan bloklar soni, V o`qilgan bloklar joylashtirilgan birinchi adres raqami.
3. **Rewrite**(F, S); - protsedurasi f fayliga S uzunlikdagi yozuvlarni yozish uchun ochadi.
4. **BlokWrite**(F, V, N); - protsedurasi F fayliga tezkor xotiraning V adresli joyidagi N ta yozuvni joylashtiradi.

Misol: n ta satrdan iborat bo`lgan matnli fayl hosil qiluvchi va ekranga chiqaruvchi programma tuzilsin.

```

program fayl;
Var f: text;
    s: string;
    i, n: integer;
begin
    assign(f, 'matn.txt'); rewrite(f);
    write('n='); readln(n);
    writeln('Fayl elementlarini kiriting');
    for i:=1 to n do
    begin
        readln(s); writeln(f,s);
    end; close(f); Reset(f);
    writeln('Fayl elementlari');
    for i:=1 to n do
    begin
        readln(f,s); writeln(s);
    end; readln;
end.
```



Nazorat savollari.

1. Matnli fayllar qanday e`lon qilinadi?
2. Matnli fayllarni toifali fayllardan qanday farqi bor?
3. **Append** va **Rewrite** protseduralari farqi?
4. Matnli fayllar bilan ishlovchi protsedura va funksiyalarni tushuntirib bering?
5. Toifalanmagan fayllar bilan ishlovchi protsedura va funksiyalar?

Mustaqil yechish uchun masalalar:

1. n ta satrdan iborat bo`lgan matnli fayl hosil qiling. Shu fayldagi eng uzun va eng qisqa satrni aniqlovchi programma tuzilsin.
2. n ta satrdan iborat bo`lgan matnli fayl hosil qiling. Shu fayldagi kichik harflarni kattasiga almashtiruvchi programma tuzilsin.
3. n ta satrdan iborat bo`lgan matnli fayl hosil qiling. Har bir satrdagi oshiqcha probellarni olib tashlovchi programma tuzilsin.
4. n ta satrdan iborat bo`lgan matnli fayl hosil qiling. Har bir satrda sonlar probel bilan ajratilgan. Shu fayldagi sonlar yi`g`indisini chiqaruvchi programma tuzilsin.
5. n ta satrdan iborat bo`lgan matnli fayl hosil qiling. Shu faylda foydalanuvchi tomonidan kiritilgan so`z borligini aniqlovchi programma tuzilsin.
6. n ta satrdan iborat bo`lgan matnli fayl hosil qiling. Fayldagi foydalanuvchi tomonidan kiritilgan s1 so`zini s2 so`zi bilan almashtiruvchi programma tuzilsin.
7. n ta satrdan iborat bo`lgan matnli (f) fayl hosil qiling. Faylning har bir satrida inglizcha va o`zbekcha so`zlar “-” belgisi bilan ajratilgan. Inglizcha so`zlarni E fayliga, o`zbekcha so`zlarni U fayliga o`tqazuvchi programma tuzilsin.

16-ma’ruza: Turbo Paskalda modullar

Reja:

1. **Modullar. Modulli programmalash**
2. **Crt moduli. Klaviatura va ekran bilan ishlash**

1. Modullar. Modulli programmalash

Har qanday malakali dasturchi o`z programmasini protsedura va funkdiyalaridan foydalanib tuzadi. Lekin bu programma mantlarini har bir programma uchun qayta yozish yoki ko`chirib o`tqazish lozim. Yuqori darajali dasturlash tillarida bu masalani hal qilish uchun modullar ishlab chiqish kiritilgan.

Modul – resurslaridan boshqa programmalarda ham foydalanish mumkin bo`lgan programmadir.

Biz programma tuzish mobaynida bir necha bor modullarga duch keldik. Yani programmada **CRT.TPU**, **GRAPH.TPU** modullarini ishlatdik. Bular standart modullardir. Quyida pascal da ishlatiladigan ba`zi standart modullar keltirilgan. Bu modullardan foydalanish uchun modullarni e`lon qilish bo`limida (**Uses**) e`lon qilish lozim.

System	Standart protsedura va funksiyalarni o`z ishiga oladi. Bu modulni e`lon qilmasa ham bo`ladi. Chunki avtomatik tarzda barcha
---------------	---

	programmalar uchun qo'llaniladi.
Dos	MS Dos operatsion tizimi bilan ishlovchi protsedura va funksiyalarni o'z ichiga oladi.
Crt	Ekran, klaviatura va ovozni nazorat qiluvchi protsedura va funksiyalarni o'z ichiga oladi.
Graph	Grafikaga oid protsedura va funksiyalarni o'z ichiga oladi.
Printer	Printer nazorati protseduralarni o'z ichiga oladi.

Standart modullardan tashqari, foydalanuvchi tomonidan ishlab chiqilgan modullarga “foydalanuvchi modullari” deyiladi. Foydalanuvchi modullaridan foydalanish uchun modullarni e'lon qilish (**Uses**) bo'limida e'lon qilinadi va programmaning istalgan qismida xuddi pascal operatorlari kabi ishlatiladi.

Foydalanuvchi modullarini tashkil qilish.

Modul quyidagi qismlardan tashkil topadi:

1. Modul sarlavhasi.
2. Interfeys bo'limi.
3. Realizatsion bo'limi.
4. Initsializatsiya¹ bo'limi.

UNIT modul_sarlavhasi;

interface

{Global e'lon qilishlar}

implementation

{ Lokal e'lon qilishlar

Global protsedura va Funktsiyalar }

Begin

{Initsializatsiya bo'limi}

End.

Modul sarlavhasi

Modul sarlavhasi **unit** xizmatchi so'zidan keyin yoziladigan identifikator bo'lib, modul saqlanadigan fayl nomi bilan bir hil bo'lishi lozim. Modul sarlavhasini tashlab ketish mumkin emas. Modul yozilgan programma kompilyatsiya qilinganda **.exe** kengaytmali fayl emas, **.tpu** kengaytmali fayl hosil bo'ladi.

Interfeys bo'limi

Interfeys bo'limida tashqi programmada ishlatish mumkin bo'lgan barcha o'zgaraslar, o'zgaruvchilar, ma'lumotlar toifalari, funksiyalar va protseduralar keltiriladi. Interfeys bo'limin **interface** xizmatchi so'zidan boshlanadi. Modulda ham, boshqa programmalar kabi boshqa modul resurslaridan foydalanish mumkin. Buning uchun **interface** xizmatchi so'zidan keyin **uses** xizmatchi so'zi orqali modullar e'lon qilinadi. Agar modullardan foydalanilmasa **uses** bo'limi tushirib qoldiriladi. Bulardan so'ng modul xam boshqa programmalar kabi quyidagi bo'limlarni o'z ichiga olishi mumkin:

- o'zgaraslarni e'lon qilish bo'limi;
- toifalarni e'lon qilish bo'limi;

- o`zgaruvchilarni e`lon qilish bo`limi;
- protsedura va funksiyalarni e`lon qilish bo`limi;

Protsedura va funksiyalarni e`lon qilish bo`limida faqat sarlavhalar keltiriladi. Protsedura va funksiyalar esa realizatsion bo`limda keltiriladi.

Realizatsion bo`limi

Realizatsion bo`limi, modulning qism dasturlarini o`z ichiga oladi. Bundan tashqari bu bo`limda lokal metkalar, o`zgarmaslar, toifalar va o`zgaruvchilar bo`lishi mumkin.

Realizatsion bo`limi **implementation** xizmatchi so`zidan boshlanadi. So`ngra **uses** xizmatchi so`zidan keyin qism dasturlarda ishlatilgan modullar ro`yhati keltiriladi. Zarur modullar **interface** bo`limida keltirilgan bo`lsa, bu erda qayta yozish shart emas. Undan so`ng quyidagilar bo`lishi mumkin:

- metkalarni e`lon qilish bo`limi;
- lokal o`zgarmaslarni e`lon qilish bo`limi;
- lokal toifalarni e`lon qilish bo`limi;
- lokal o`zgaruvchilarni e`lon qilish bo`limi;
- protsedura va funksiyalarni e`lon qilish bo`limi;

Realizatsion bo`lim “ishchi bo`lim” deb ham yuritiladi. **Interface** bo`limida e`lon qilingan protsedura va funksiya sarlavhalari **implementation** bo`limi sarlavhalari bilan mos tushishi lozim.

Initsializatsiya bo`limi

Initsializatsiya bo`limi **Begin** va **End** xizmatchi so`zlari orasida keltiriladi. Agar **Begin** so`zi tushirib qoldirilgan bo`lsa, u holda initsializatsiya bo`limi ham yo`q hisoblanadi. Ba`zan modullarga murojaat qilishdan oldin uni initsializatsiya qilishga to`g`ri keladi. Bu har qanday amal bo`lishi mumkin. Masalan ekranni tozalash, grafik rejimni o`rnatish, mantiqiy fayllarni fizik fayllar bilan bog`lash va hakazo...

Biron – bir modulni ishlatayotgan programma bajarilganda bu modulning initsializatsiya bo`limi (agar u mavjud bo`lsa) programmaning asosiy tanasi bajarilishi oldidan bir marta bajariladi. Bir nechta modullar ishlatilgan bo`lsa ularning initsializatsiya bo`limi **uses** operatorida ko`rsatilgan tartibda bajariladi.

Misol: Uch burchak yuzini hisoblovchi namuna nomli foydalanuvchi moduli hosil qilinsin. Buniy uchun quyidagi amallar ketma – ketligi bajariladi:

1. Quyidagi programma pascal redaktorida kiritilib, **namuna** deb nom qo`yoladi.

Unit Namuna;

Interface

uses crt;

procedure yuza(var s: real; a,h:real);

implementation

procedure yuza(var s: real; a,h:real);

begin

s:=a*h/2;

end;

```

Begin
  clrscr;
  {Programma ishga tushganida ekranni tozalash}
  WriteLn('Namuna modulidan foydalamish');
End.

```

2. **ALT + F9** tugmalari orqali kompilyatsiya qilinadi. Kompilyatsiyadan so`ng **namuna.tpu** fayli hosil bo`ladi.
3. Quyidagi programma o`rqli namuna modulidan foydalanish mumkin.

```

Uses namuna;
Var a,h,s:real;
Begin
  Write(' a='); ReadLn(a);
  Write(' h='); ReadLn(h);
  yuza(s,2,3);
  WriteLn(' s=',s:2:2);
  ReadLn;
End.

```



Nazorat savollari.

1. Modul nima ?
2. Modullar nima uchun ishlatiladi ?
3. Standart modullarni sanab bering.
4. Foydalanuvchi modullari qanday hosil qilinadi ?
5. Modulning programmadan farqi nimada ?

Mustaqil yechish uchun masalalar:

1. n natural soni berilgan. n faktorialni hisoblovchi **fact** nomli foydalanuvchi moduli tuzilsin va bu moduldan foydalanilgan programma tuzilsin.
2. Kvadrat, doira, uch burchakni ikkita tomoni va ular orasidagi burchagi orqali yuzini hisoblovchi **yuza** nomli foydalanuvchi moduli hosil qilinsin. Yuza modulidan foydalanilgan programma ishlab chiqilsin.
3. n natural soni va n ta butun sonli elementdan iborat chisiqli massiv berilgan. Chiziqli massiv elementlarini kirituvchi va eng katta qiymatini topuvchi **masmax** foydalanuvchi modulini hosilqiling.
4. Kvadrat massiv o`lchami n berilgan. Massiv elementlari kirituvchi va chiqaruvchi **massiv** nomli foydalanuvchi moduli tuzilsin.

2. Crt moduli. Klaviatura va ekran bilan ishlash

Modul – resurslaridan boshqa programmalarda ham foydalanish mumkin bo`lgan programmadir. Modullarga keyinroq batafsil tohtalamiz. Hozircha standart modullardan bo`lgan **Crt** moduli imkoniyatlari bilan tanishib chiqamiz.

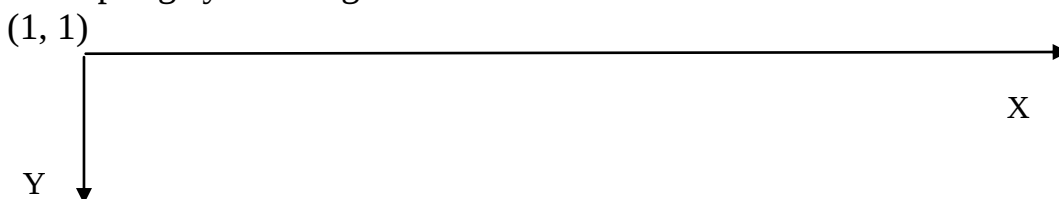
Crt (Cathode ray tube – electron nurli trubka) moduli ekran, klaviatura va ovozni nazorat qiluvchi protsedura va funksiyalarni o`z ichiga oladi. Crt moduli bilan ishlaganda toliq ekran alohida satrlarga, satrlar bittadan belgi sig`adigan kataklarga bo`linadi. Bu kataklarning har biriga alohida fon rangi, katakka

yoziladigan belgiga alohida rang berish mumkin. Bundan tashqari belgini yonib – o`chib turadigan qilish mumkin. Programmada Crt modulidan foydalanish uchun programma sarlavhasidan keyin **Uses** xizmatchi so`zi orqali **Crt** moduli e`lon qilinadi.

Matnli rejim odatda 25 ta qator, 80 ta belgi sig`dirish imkoniyatiga ega. Ekranda hosil qilish mumkin bo`lgan ranglar soni esa 16 ta.

Koordinatalar sistemasi

Kompyuter ekrani koordinatalar sistemasida sanoq boshi qilib ekranning yuqori chap burchagi kiritilgan. OX o`qi chapdan o`ngga, OY o`qi yuqori chap burchakdan pastga yo`naltirilgan.



Mantli rejimda (modul Crt) koordinata boshi (1,1), grafik rejimdan (modul Graph) koordinata boshi qilib (0,0) qabul qilingan.

Rangni boshqaruvchi protseduralar

HighVideo – protsedurasi ekranga chiqarilayotgan belgilar uchun yuqori yorug`lik o`rnatadi.

LowVideo – protsedurasi ekranga chiqarilayotgan belgilar uchun past yorug`lik o`rnatadi.

NormVideo – protsedurasi ekranga chiqarilayotgan belgilar uchun o`rtacha yorug`lik o`rnatadi.

TextBackGround(color) – protsedurasi ekranga chiqariladigan belgi uchun ekran foni rangini o`rnatish. Color faqat 0 – 7 oraliqdagi qiymatlarni qabul qilishi mumkin.

TextColor(color) - protsedurasi ekranga chiqariladigan matn rangini o`rnatadi.

Ekran bilan ishlovchi protseduralar

ClrEol – protsedurasi kursor turgan joydan boshlab satr oxirigacha ekranni tozalaydi. Bu protsedura bajarilgandan so`ng, ekranning shu qismiga, joriy fon rangi o`rnatiladi.

ClrScr – protsedurasi ekranni joriy fon bilan tozalaydi. Agar fon o`rnatilmagan bo`lsa qora rang bilan tozalanadi. Kursor koordinatasi (1,1) bo`lgan katakka o`rnatiladi.

DelLine – protsedurasi ekranning kursor turgan satrini o`chiradi.

GotoXY(x,y) – protsedurasi kursorni ko`rsatilgan koordinataga o`rnatadi. Agar birorta koordinata no`to`g`ri berilgan bo`lsa protsedura bajarilmaydi.

InsLine – protsedurasi kursor turgan o`ringa, joriy fon rangida bosh satr qoshadi.

Window(x1,y1,x2,y2) – protsedurasi ekranda matnli yona hosil qiladi. kursorni (1,1) koordinataga o`rnatadi. Agar birorta koordinata no`to`g`ri berilgan bo`lsa, protsedura bajarilmaydi.

Ekran bilan ishlovchi funksiyalar

WhereX: Byte – funksiyasi kursorning joriy X koordinatasini aniqlaydi.

WhereY: Byte – funksiyasi kursorning joriy Y koordinatasini aniqlaydi.

Klaviatura bilan ishlovchi funksiyalar

KeyPressed: Boolean – funksiyasi klaviaturadan biror tugma bosilsa **True**, aks holda **False** qiymat qaytaradi. (Shift, Alt, NumLock – kabi yordamchi tugmalaridan tashqari)

ReadKey: Char – funksiyasi klaviaturadan belgini o`qiydi.

Ovozni boshqaruvchi protseduralar

Sound(n) – protsedurasi n chastotali tovush chiqishini ta`minlaydi.

NoSound – protsedurasi tovush dinamigini o`chiradi.

Delay(n) – protsedurasi programmani n millisekund davomida toxtashini ta`minlaydi.

Crt modulida quyidagi o`zgarmaslar orqali ranglarni o`rnatish mumkin.

kodi	Color (nomi)	rang
0	Black	Qora
1	Blue	Ko`k
2	Green	Yashil
3	Cyan	Havorang
4	Red	Qizil
5	Magenta	Malinarang
6	Brown	Jigarrang
7	LightGray	Och kulrang
8	DarkGray	To`q kulrang
9	LightBlue	Och havorang
10	LightGreen	Och zangori
11	LightCyan	Feruzarang
12	LightRed	Och qizil
13	LightMagenta	Och malinarang
14	Yellow	Sariq
15	White	Oq
Color+128	Blink	Yonib – o`chish

Eslatma:

Biror rangni o`rnatishda uning kodini yoki nomini ishlatish mumkin.

Namuna: Quyidagi misollarni kiriting va Crt moduli protsedura va funksiyalarining ishlatilishiga e`tibor bering.

```
Program rangli_suzlar1;
```

```
Uses crt;
```

```
Var ism, fam: string;
```

```
Begin
```

```
  ClrScr; {ekranni tozalash}
```

```
  WriteLn('Familiyangizni kiriting');
```

```
  ReadLn(fam);
```

```
  WriteLn('Ismingizni kiriting');
```

```
Program rangli_suzlar2;
```

```
Uses crt;
```

```
Var ism: string;
```

```
    i: integer;
```

```
Begin
```

```
  ClrScr; {ekranni tozalash}
```

```
  WriteLn('Ismingizni kiriting');
```

```
  ReadLn(ism);
```

ReadLn(ism); TextColor (Red); {matn rangini o`rnatish} <b>GotoXY</b> (20,10); {kursorni o`rnatish} WriteLn(Fam); TextBackground (green); {Fon rangini o`rnatish} <b>GotoXY</b> (20,11); {kursorni o`rnatish } WriteLn(Ism); ReadLn; End.	for i:=0 to 15 do Begin TextColor (i); {matn rangini o`rnatish} <b>GotoXY</b> (20,10+i); {kursorni o`rnatish } WriteLn(Ism); End; for i:=0 to 15 do Begin TextBackground (i+1); {fon rangini o`rnatish} <b>TextColor</b> (i); {matn rangini o`rnatish} GotoXY (40,10+i); {kursorni o`rnatish } WriteLn(Ism); End; ReadLn; End.
---	---



Nazorat savollari.

1. Crt moduli nima maqsadda qo`llaniladi?
2. Crt moduli funksiya va protseduralaridan foydalanish uchun nima qilish lozim?
3. Kompyuter ekrani koordinatalar sistemasini tushuntirib bering.
4. Ekran foniga necha hil rang tanlash mumkin?
5. Mant rangiga necha hil rang tanlash mumkin?
6. Crt moduli funksiya va protseduralarini sanab bering.

Mustaqil yechish uchun masalalar:

1. Kiritilgan ismni ekran o`rtalariga, sariq fonga qizil rang bilan chiqaruvchi programma tuzilsin. Programma ishga tushganida ekran tozalansin.
2. "Referat" tituliga o`xshash, har hil rangli yozuvlarni chiqaruvchi programma tuzilsin.
3. "Yuruvchi satr". Qora ekran o`rtalarida o`ngdan chapga qarab yuruvchi satr programmasini tuzing.
4. Kiritilgan satrni xar bir harfini xil rangda chiqaruvchi programma tuzilsin.

Muhim ko`rsatmalar.

Klaviatura orqali kiritilgan belgini ASCII kodini aniqlovchi programma tuzilsin.

Klaviatura tugmalarini ikki guruhga bo`lish mumkin. Birinchi guruh oddiy tugmalar. Ikkinchi guruh maxsus tugmalar (klavishalar). Ikkinchi guruh tugmalari ReadKey funksiyasi orqali o`qilganda bir bosishda birdaniga ikkita kod qaytaradi. Bulardan birinchisi albatta #0 belgisi bo`ladi. Klaviatura orqali kiritilgan belgini ASCII kodini aniqlash uchun quyidagi programmadan foydalanish mumkin.

Uses Crt;

```

Var c: char;
Begin
  ClrScr;
  Repeat
    c:=ReadKey;    {belgini o`qish}
    if c=#0 then    {agar maxsus belgi bo`lsa}
      Begin
        c:=ReadKey;
        WriteLn('Maxsus klavisha+',c,Ord(c));
      End
    else WriteLn(c,'->',Ord(c));
  Until c=#13; { enter tugmasini bosguncha takrorlash}
End.

```

KeyPressed: Boolean;

KeyPressed – funksiyasi klaviaturadan biror tugma bosilganda programmani tugatish uchun juda qulay.

Repeat Until KeyPressed;

Yoki quyidagicha foydalanish ham mumkin.

While not KeyPressed do <sikl tanasi>;

Bu jarayon klaviatura orqali biror tugma bosilguncha bajariladi.

Uses Crt;

```

Var c: char;
chiqish : boolean;
Begin
  ClrScr;
  WriteLn('F1 -> Tovush chiqarish');
  WriteLn('F2 -> Tovushni to`xtatish');
  WriteLn('F4 -> Programmani tugatish');
  chiqish:=False;
  Repeat
    c:=ReadKey;
    if c=#0 then
      Begin
        c:=ReadKey;

        Case c Of
          #59: Sound(5000); {F1 tugmasi}
          #60: NoSound;    {F2 tugmasi}
          #62: chiqish:=true; {F4 tugmasi}
        End;
      End;
    Until chiqish;
  End.

```

1. a va b haqiqiy sonlari berilgan. Quyidagi menyularni chiqaruvchi programma tuzilsin.

F1 – Yi`g`indi; **F2** – Ayirma;
F3 – Ko`paytma;                      **F4** – Bo`linma;
F10 – Chiqish

2. X haqiqiy sonlari berilgan. Quyidagi menyularni chiqaruvchi programma tuzilsin.

F5 – Kvadrat; **F6** – Ildiz;
F7 – sin(x); **F8** – cos(x);
F9 – tg(x); **F10** – Chiqish

3. Rang kodini bildiruvchi son kirituvchi. Quyidagi menyularni o`zida mujassamlashtirgan programma tuzilsin.

F5 – Fon rangini o`rnatish;
F6 – Matn rangini o`rnatish;
F7 – Ixtiyoriy fon rangi o`rnatish;
F8 – Ixtiyoriy matn rangi o`rnatish;
F9 – Rang kodini kiritish;
F10 – Chiqish.

17-ma'ruza: Graph moduli

Reja:

1. Grafik rejim. Koordinatalar sistemasi
2. Grafik imkoniyatlar. Shakllarni bo'yash
3. Koordinata ko'rsatkichini o'rnatish
4. Bo'yalgan shakllarni chizish

1. Grafik rejim. Koordinatalar sistemasi

Barcha dasturlash tillari kabi Paskal dasturlash tilida ham grafik ma'lumotlar bilan ishlash mumkin. Buning uchun **Graph** modulidan foydalaniladi. **Graph** modulini e'lon qilish uchun **Uses** xizmatchi so'zidan keyin **Graph** yoziladi.

Uses Graph;

Grafik rejimni o'rnatish

Biz shu vaqtgacha matnli rejimda ishladik. Shuning uchun ekranda faqat so'zlar va simvollarni ko'rganmiz. Ekranga aylana, uchburchak va boshqa shakllarni chiqarish uchun grafik rejimni o'rnatish lozim. Grafik rejimni o'rnatish uchun **InitGraph** prosedurasidan foydalaniladi.

InitGraph(Gd, Gm:Integer, Path: string); - bu erda **Gd** - drayver raqami, **Gm** – rejim raqami, **Path** – kerakli drayver joylashgan fayl yo'li. Drayverlar ***.bgi** kengaytmali fayllarda saqlanadi. Agar **Path** bo'sh satrdan iborat bo'lsa, drayver joriy katalogdan qidiriladi.

Turli xil drayverlarni o'rnatish va rejimlarni tanlash uchun quyidagi jadvaldan foydalanish mumkin.

nuqta

Drayver raqami	Rejim raqami
Detect=0	CGAC0=0 {320X200 ta nuqta 4 xil rang}

CGA=1	CGAC1=1 {320X200 ta nuqta 4 xil rang}
MCCA=2	CGAC2=2 {320X200 ta nuqta 4 xil rang}
EGA=3	EGALO=0 {640X200 ta nuqta 16 xil rang}
EGA64=4	EGAHi=1 {640X350 ta nuqta 16 xil rang}
EGAMONO=5	EGA64Lo=0 {640X200 ta nuqta 16 xil rang}
IBM8514=6	EGA64Hi=1 {640X350 ta nuqta 16 xil rang}
HERCMONO=7	VGALo=0 {640X350 ta nuqta 16 xil rang}
ATT400=8	VGALoMed=1 {640X350 ta nuqta 16 xil rang}
VGA=9	VGAHi=2 {640X480 ta nuqta 16 xil rang}

Quyidagi kichik programmada grafik rejimni o‘rnatishni ko‘rib chiqamiz.

Program Grafik_rejim;

Uses Graph;

Var gd, gm, xato: Integer;

Begin

gd:=**Detect**; {Drayverni avtomatik aniqlash}

InitGraph(gd,gm,'D:\Bp\bgi'); {Grafik rejimni o‘rnatish}

xato := **GraphResult**;

if xato<>0 then WriteLn(xato,'-raqamli xatolik')

else Line(100,100,300,100);

{To‘g‘ri chiziq chizish}

ReadLn;

CloseGraph; {Grafik rejimni yopish}

End.

Agar **InitGraph** prosedurasi ishga tushganida **Gd** = 0 (yoki **Detect**) bo‘lsa, zarur drayver va bu drayverga mos eng yaxshi rejim avtomatik tarzda aniqlanadi.

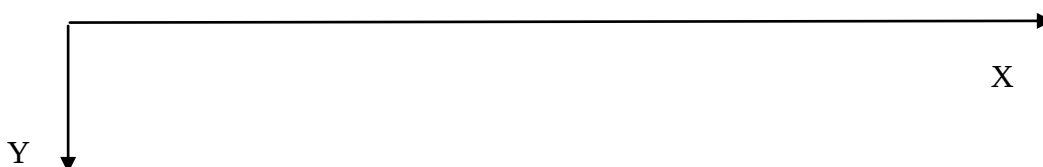
GraphResult; - funksiyasi grafik rejimga o‘tishda sodir bo‘lgan xatoliklarni aniqlash uchun ishlatiladi. Bu funksiya quyidagi qiymatlarni qaytarishi mumkin.

- 0 – qiymati xatolik sodir bo‘lmasa qaytariladi;
- 2 – grafik plata o‘rnatilmagan;
- 3 – drayver fayli topilmadi. Bu xatolik drayver fayli yo‘li xato ko‘rsatilganida sodir bo‘lishi mumkin.
- 4 – noto‘uri drayver o‘rnatilgan.
- 5 – grafik rejimni o‘rnatish uchun tezkor xotira etishmayapti.

CloseGraph; - grafik rejimni yopish.

Grafik rejimda koordinatalar sistemasi (0,0) dan boshlanadi.

(0,0)



Grafika bilan ishlovchi ba’zi prosedura va funksiyalar

Line(x1,y1,x2,y2:Integer); - protsedurasi (x1,y1) va (x2,y2) nuqtalarni tutashtiruvchi kesma chizadi.

Rectangle(x1,y1,x2,y2:Integer); – protsedurasi joriy rang bilan yuqori chap burchagi (x1,y1) va quyi o‘ng burchagi (x2,y2) nuqtalar bo‘lgan to‘g‘ri to‘rtburchak chizadi.

Circle(X,Y:Integer; R: Word); – protsedurasi markazi (X,Y) nuqtada bo‘lgan R – radiusli aylana chizadi.

Arc(x,y:Integer; StAng, EndAng, Radius: Word); - aylana yoyini chizish;

x,y – aylana markazi koordinatalari; **Radius** – aylana radiusi;

StAng, EndAng – yoyning boshlang‘ich va oxirgi burchaklari;

Ellipse(x,y:Integer; StAng, EndAng, XR, YR: Word); - elips yoyini chizish (x,y) – elips markazi; StAng, EndAng – yoyning boshlang‘ich va oxirgi burchaklari; XR, YR – elips balandligi va eni.

PutPixel(x,y, color); - protsedurasi (x,y) koordinatasiga color rangli nuqta qo‘yadi.

GetPixel(x,y); - funksiyasi (x,y) koordinatali nuqtaga qoyilgan rangni aniqlaydi.

GetColor: word; - funksiyasi joriy rangni aniqlaydi.

SetBkColor(color:word); - protsedurasi fon rangini o‘rnatadi.

GetBkColor: word; - funksiyasi fon rangni aniqlaydi.

ClearDevice; protsedurasi grafik ekranni tozalab kursorni (0,0) nuqtaga o‘rnatadil.

SetColor(Color:word); - protsedurasi chiziq rangini o‘rnatadi. Color sifatida rang nomi yoki kodini keltirish mumkin. Quyidagi jadvalda keltirilgan ranglardan foydalanish mumkin.

kodi	Color (nomi)	rang	Graph ga doir misol
0	Black	Qora	Program uy; Uses Graph ; Var gd,gm: integer; Begin clrscr ; gd := Detect ; InitGraph (Gd,Gm,'d:\bp\BGI\'); Rectangle (100,200,500,450); Line (100,200,300,100); Line (300,100,500,200); Rectangle (150,250,220,320); Rectangle (280,350,320,450); ReadLn ; CloseGraph ; End.
1	Blue	Ko‘k	
2	Green	Yashil	
3	Cyan	Havorang	
4	Red	Qizil	
5	Magenta	Malinarang	
6	Brown	Jigarrang	
7	LightGray	Och kulrang	
8	DarkGray	To‘q kulrang	
9	LightBlue	Och havorang	
10	LightGreen	Och zangori	
11	LightCyan	Feruzarang	
12	LightRed	Och qizil	
13	LightMagenta	Och malinarang	
14	Yellow	Sariq	
15	White	Oq	

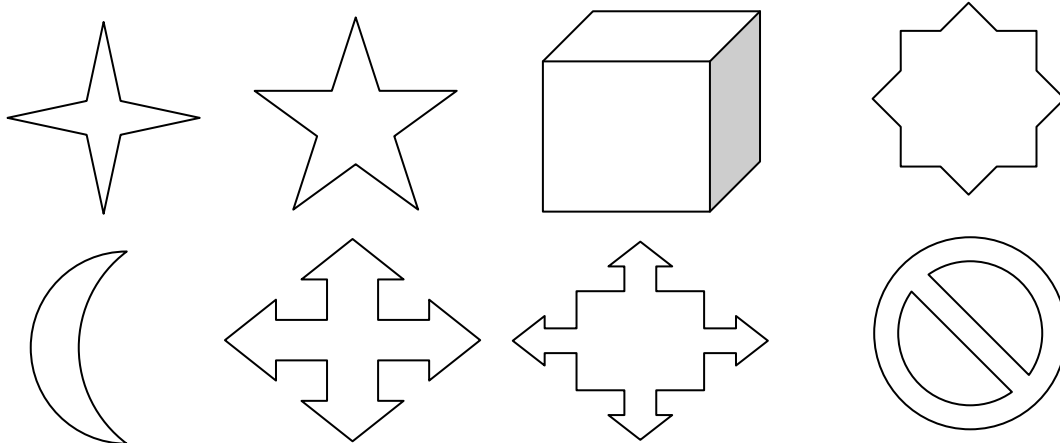


Nazorat savollari.

1. Grafik rejim qanday o'rnatiladi?
2. Grafika bilan ishlovchi fayl drayverlari qanday kengaytmali bo'ladi?
3. Siz ishlayotgan kompyuterda drayverlar qaysi katalogda saqlangan?
4. Qachon drayver joriy katalogdan izlaniladi?
5. GraphResult funksiyasini tushuntirib bering?
6. Grafika bilan ishlovchi protsedura va funksiyalarni sanab bering?

Mustaqil yechish uchun masalalar:

1. **Graph** modulidan foydalangan holda, ixtiyoriy mavzuda rasm chizing.
2. Uchburchak, to'rtburchak, aylana, elips va besh yulduz shakllarini xar hil rangda chizuvchi programma tuzilsin.
3. "Mashina" rasmini chizuvchi programma tuzilsin.
4. "Robot" rasmini chizuvchi programma tuzilsin.
5. Quyidagi shakllarni chizuvchi programma tuzilsin.



2. Grafik imkoniyatlar. Shakllarni bo'yash

Chiziqning joriy rangi deb, biror shakl chizig'ini chizish uchun ishlatiladigan rangga aytiladi. Chiziqning joriy rangi **SetColor** protsedurasi orqali o'rnatiladi. Chiziqning joriy rangi sifatida odatda oq rang o'rnatilgan bo'ladi.

Fonning joriy rangi sifatida odatda qora rang o'rnatilgan bo'ladi. Fon rangini o'zgartirish uchun **SetBkColor** protsedurasidan foydalaniladi.

Joriy holat – bu koordinatalar ko'rsatkichi (x,y) o'rnatilgan nuqtadi. Qoidaga binaon joriy holat biror shaklni chizish bilan o'zgaradi. Ba'zi protsedura va funksiyalar bajarilgandan keyin joriy holat o'zgarishiga e'tibor bering:

ClearDevice;	(0,0);
Line (x1,y1,x2,y2:Integer);	Joriy holat o'zgarmaydi;
Rectangle (x1,y1,x2,y2:Integer);	Joriy holat o'zgarmaydi;
PutPixel (x,y, color);	Joriy holat o'zgarmaydi;
Circle (X,Y:Integer; R: Word);	(x,y);
Arc (x,y:Integer; StAng, EndAng, Radius: Word);	(x,y);
Ellipse (x,y:Integer; StAng, EndAng, XR, YR: Word);	(x,y);

3. Koordinata ko'rsatkichini o'rnatish

MoveTo(x,y); - protsedurasi koordinata ko'rsatkichini (x,y) nuqtaga o'rnatadi. Bu protsedura bajarilganda nuqta ekranda yoritilmaydi.

MoveRel(dx,dy); - protsedurasi koordinata ko'rsatkichini joriy holatga nisbatan (dx,dy) ga siljitadi. Bu protsedura bajarilganda nuqta ekranda yoritilmaydi.

LineTo(x,y); - protsedurasi joriy holatdan boshlab (x,y) nuqtaga kesma chizadi. Bu protsedura bajarilgandan keyin joriy holat (x,y) nuqtaga o'zgaradi.

LineRel(dx,dy); - protsedurasi joriy holatdan boshlab, joriy holatga nisbatan (dx,dy) nuqtagacha kesma chizadi. Bu protsedura bajarilgandan keyin joriy holat oldingi holatiga nisbatan (dx,dy) nuqtaga o'zgaradi.

GetX: Integer; - funksiyasi joriy holatning X koordinatasini aniqlaydi.

GetY: Integer; - funksiyasi joriy holatning Y koordinatasini aniqlaydi.

GetMaxX: Integer; - funksiyasi X koordinatasini eng katta qiymatini aniqlaydi.

GetMaxY: Integer; - funksiyasi Y koordinatasini eng katta qiymatini aniqlaydi.

4. Bo'yalgan shakllarni chizish

SetLineStyle(Ln, P, T); - protsedurasi chiziq uslubini o'rnatadi. Bu protsedura quyidagi qiymatlarni qabul qilishi mumkin.

Ln – chiziq uslubi		T – chiziq qalinligi	
0	Usluksiz chiziq	1	Normal
1	Punktir chiziq	3	Qalin
2	Shtrixpunktir chiziq	P – chiziq shabloni	
3	Shtrixli	0 – qiymat qabul qiladi	

SetFillStyle(Style,Color:word); - protsedurasi **Color** rangi bilan **Style** uslubida shaklni to'ldiradi.**Style** quyidagi o'zgarmaslarni (**const**) qabul qilishi mumkin.

0=EmptyFill	Sohani fon rangiga bo'yash
1=SolidFill	Sohani belgilangan rangda to'ldirish
2=LineFill	Sohani qalin gorizontaal chiziq bilan to'ldirish
3=LtSlashFill	Sohani ingichka “//” chiziqlar bilan to'ldirish
4=SlashFill	Sohani qalin “//” chiziqlar bilan to'ldirish
5=BkSlashFill	Sohani qalin “\\” chiziqlar bilan to'ldirish
6=LtBkSlashFill	Sohani ingichka “\\” chiziqlar bilan to'ldirish
7=HatChFill	Sohani to'r bilan to'ldirish
8=XHatChFill	Sohani egri to'r bilan to'ldirish
9=InterLeaveFill	Sohani zich shtrix bilan to'ldirish
10=WideDotFill	Sohani kam sondagi nuqtalar bilan to'ldirish
11=CloseDotFill	Sohani ko'p sondagi nuqtalar bilan to'ldirish
12=UserFill	Sohani programmist belgilagan uslubda to'ldirish

Bar(x1,y1,x2,y2:Integer); - protsedurasi o'rnatilgan rang yuqori chap burchagi (x1,y1) va quyi o'ng burchagi (x2,y2) nuqtalar bo'lgan bo'yalgan to'g'ri to'rtburchak chizadi.

Bar3D(x1,y1,x2,y2:Integer; Depth:Word; Top:Boolean); - protsedurasi bo'yazlgan parallelepiped chizadi.

FillEllipse(x,y:Integer; xr,yr:Word); - protsedurasi bo'yalgan elips chizadi.

FloodFill(x,y:Integer; Border:Word); - protsedurasi chegarasi **Border** bo'lgan sohani joriy rangga bo'yaydi. (x,y) bo'yalishi lozim bo'lgan sohadagi ixtiyoriy nutqa.

PieSlice(x,y:Integer; StAng,EndAng,Radius:Word); - protsedurasi aylana sectorini chizadi.

Sector(x,y:Integer; StAng,EndAng,XR,YR:Word); - protsedurasi elips sectorini chizadi.

Grafik rejimdagi matnlar

SetTextStyle(Font, Direction, Size); - protsedurasi kerakli shriftni o'rnatadi.

Font – shriftni aniqlovchi o'zgaruvchi. (0,1,2,3,4 – qiymatlarini qabul qilishi mumkin). **Direction** – matnni chop qilish yo'nalishi. (0 – chapdan o'ngga, 1 – pastdan yuqoriga). **Size** – matn kattaligini aniqlaydi.

OutTextXY(x,y:Integer; Text:String); -protsedurasi x,y nuqtaga matnni chiqarish.

SetTextJustify(Horiz, Vert); - protsedurasi OutTextXY protsedurasi orqali chiqariladigan matnni tekislash uchun ishlatiladi.

Gorizantal tekislash (Horiz)		Vertikal tekislash (Vert)	
LeftText=0	Chapga tekislash	BottomText=0	Pastki tomonga tekislash
CenterText=1	Markazga tekislash	CenterText=1	Markazga tekislash
RightText=2	O'ngga tekislash	TopText=3	Yuqori tomonga tekislash



Nazorat savollari.

1. Chiziqning joriy rangi nima?
2. Joriy holat nima?
3. Koordinata ko'rsatkichini o'rnatish protsedura va funksiyalari?
4. SetLineStyle protsedurasini tushuntirib bering?
5. SetFillStyle protsedurasini tushuntirib bering?
6. Grafik rejimda matnlar bilan ishlovchi protseduralar?

Mustaqil yechish uchun masalalar:

1. Shakllarni bo'yash protseduralaridan foydalangan holda, ixtiyoriy mavzuda rasm chizing.
2. Xar hil rangli nishon rasmini chizuvchi programma tuzilsin.
- 3 $Y=\sin(x)$ funksiyasi grafikini chizuvchi programma tuzilsin.
4. $Y=\tan(x)$ funksiyasi grafikini chizuvchi programma tuzilsin.
- 5 $Y=X^2$ funksiyasi grafikini chizuvchi programma tuzilsin.
6. O'zbekiston Respublikasi bayrog'ini chizuvchi programma tuzilsin.
7. "Mashina" rasmini chizuvchi programma tuzilsin.

1-AMALIY MASHG'ULOT: Algoritmning asosiy tiplari: chiziqli, tarmoqlanuvchi va takrorlanuvchi algoritmlar.

1. Amaliy mashg'ulotni darsini olib borish texnologiyasi

Talabalar soni:	20 – 30 talaba
Vaqt:	2 soat
Dars shakli	Mavzu bo'yicha amaliy
Ma'ruza rejasi	1. Algoritmni tasvirlash usullari 2. Algoritmning asosiy turlari: chiziqli, tarmoqlanuvchi va takrorlanuvchi algoritmlar.
Dars maqsadi: Talabalarda algoritm tushunchasi, algoritmning asosiy xossalari, ularni tasvirlash usullari, algoritm turlari bo'yicha talabalarda tushuncha xosil qilish.	
Pedagog vazifalari:	O'quv natijalari:
Algoritmni tasvirlash usullarini ko'rsatish	Algoritmning so'z bilan, blok-sxemalar shaklida, formulalar orqali, algoritmik tillar orqali tasvirlash usullari tushuntirib beriladi.
Algoritm turlariga to'xtalish	Algoritmning chiziqli, tarmoqlanuvchi va takrorlanuvchi turlariga mos konstruksiyalar tushuntiriladi.
O'qitishning usul, texnikasi:	amaliy, suxbat, baxs-munozara, muhokama qilish, fikr almashish.
O'qitish vositalari:	Ma'ruza matni, amaliy ish uchun tarqatma materiallar, slaydlar
O'qitish shakli:	Frontal
O'qitish shart – sharoitlari:	Texnik vositalarni qo'llashga mo'ljallangan auditoriya. Proektor
Monitoring va baholash:	Og'zaki so'roq, savol-javob, tahlil natijalari.

AMALIY MASHG'ULOT TEXNOLOGIK KARTASI

Faoliyat bosqich-lari	Faoliyatining mazmuni	
	O'qituvchi	Talaba
1-bosqich. Kursga va mavzuga kirish (20 min)	1.1. Auditoriyaning va talabalarining darsga tayyorlik darajasi tekshiriladi. Ma'ruza mashg'ulotida o'tilgan nazariy bilimlar yuzasidan faollashtiruvchi savollar bilan talabalarga murojat etadi. 1.2. Mavzu, uning maqsadi, kutiladigan o'quv natijalarini e'lon qiladi. (1-ilova). Mashg'ulot yakka, juftlikda, hamkorlikda ishlash texnologiyasini qo'llagan holda o'tishni ma'lum qiladi.	1.1. Savollarga javob beradilar. 1.2. Mavzuni yozadi va eshitadilar.
2 -bosqich.	2.1. Mavzu rejasi va tayanch tushunchalar bilan tanishtiradi. (3-ilova)	Tinglaydilar

Asosiy bo'lim (50 min)	2.2. Ma'ruza rejasining 1-3 punktlari bo'yicha tushuntiradi, har bir punkt nihoyasida umumlashtirib boradi. Jarayon komp'yuter slaydlarini namoyish qilish bilan olib boriladi.	Tinglaydilar. konspekt qilib boradilar.
	2.3. Tayanch iboralarga qaytiladi. Tinglovchilar ishtirokida ular yana bir bor takrorlanadi (Pinbord usulida). Mavzuga oid bo'lmagan iboralar olib tashlanib, kerakli tushuncha va iboralar qo'shiladi.	Har bir tayanch tushuncha va iboralarni muhokama qiladilar, tizimlashtirib konspekt qiladilar.
3-bosqich. Yakunlovchi (10 min)	3.1. Mavzu bo'yicha yakunlovchi xulosalar qiladi. Mavzu bo'yicha olingan bilimlarni qayerda ishlatish mumkinligi ma'lum qiladi.	Savollar beradilar
	3.2. Mavzu maqsadiga erishishdagi tinglovchilar faoliyati tahlil qilinadi va baholanadi.	BBB jadvali(4-ilova)
	3.3. Mavzu bo'yicha mustaqil o'rganish uchun topshiriqlar beradi.	Mustaqil o'rganish uchun topshiriqlarni yozib oladilar

ALGORITM HAQIDA TUSHUNCHA. ALGORITMNING ASOSIY HOSSALARI, ALGORITM IJROCHILARI, ALGORITMLARNI TASVIRLASH USULLARI, ALGORITM TURLARI, ALGORITMNING ASOSIY TIPLARI: CHIZIQLI, TARMOQLANUVCHI VATAKRORLANUVCHI ALGORITMLAR.

Reja:

2. Algoritm tushunchasi
3. Algoritm xossalari
4. Algoritmlarni tasvirlash usullari
5. Algoritmning asosiy turlari: chiziqli, tarmoqlanuvchi va takrorlanuvchi algoritmlar.

1-ilova

Tayanch tushunchalar:

Algoritm. Algoritmik til. Algoritm hossalari.

Chiziqli algoritm. Tarmoqlanuvchi algoritm.

Takrorlanuvchi algoritm.

Algoritmni tasvirlash usullari.

1. Algoritmning soʻzlar orqali ifodalanishi. Ushbu holda ijrochi uchun beriladigan har bir koʻrsatma jumlarlar orqali buyruq mazmunida beriladi.

2. Algoritmning formulalar yordamida berilishi. Algoritmni formulalar bilan berilish usulidan matematika, fizika, kimyo kabi aniq fanlarni oʻrganishda koʻplab foydalanamiz. Bu usulni baʼzan analitik ifodalash deb ham yuritishimiz mumkin.

3. Algoritmning jadval koʻrinishida berilishi. Algoritmning bu tarzda tasvirlanishidan ham koʻp foydalanamiz. Masalan, maktabda qoʻllanilib kelinayotgan V. M. Bradisning toʻrt xonali matematik jadvallari, turli lotoreya

yutuq jadvallari, musobaqalar jadvallari. Funktsiyalarning grafiklarini chizishda ham algoritmning qiymatlar jadvali koʻrinishlaridan foydalanamiz. Bu kabi jadvallardan foydalanish algoritmлари sodda boʻlgani tufayli ularni oʻzlashtirib olish osondir.

4. Algoritmning grafik shaklida tasvirlanishi. Algoritmning bu shakli sizga avvaldan tanish, chunki matematika kursida chizilgan grafiklarning koʻpchiligi algoritmning grafik usulda berilishiga misol boʻladi. Bundan tashqari shahar yoki turar joy mavzelari joylashgan biror uy hamda inshootlarini izlash va xarakatlanish boʻyicha berilgan xarita – sxemalar ham shunga misol boʻla oladi. Biz algoritmning informatikada koʻp

Algoritmarga quyidagi talablar qoʻyiladi :

1. Boshlanishi va tugashi koʻrsatilishi kerak.
2. Har qanday amal buyruq tarzida ifodalanishi shart.
3. Har bir amal ijrochiga tushunarli boʻlgan koʻrinishda ifodalangan boʻlishi shart.
4. Har bir amalda qatnashayotgan oʻzgaruvchilarning qiymatlari oldindan aniqlangan boʻlishi kerak.
5. Har qanday amal natijasi bir qiymatli boʻlishi kerak.
6. Bajariladigan amallar soni cheklangan boʻlishi kerak.
7. Yakuniy natijalarni ajratib koʻrsatish va chiqarish shart.
8. Qoʻyilgan masalani toʻla yechish uchun berilgan hamma maʼlumotlar va mumkin

Algoritm turlari

Hech qanday shart tekshirilmaydigan va faqat oldindan belgilangan tartib bilan ketma – ket bajariladigan jarayonlarni kuzatish mumkin. Bunday mazmundagi algoritmлар **chiziqli algoritmлар** deb yuritiladi.

Biroq turmushda uchraydigan juda koʻpgina jarayonlar esa shartlar asosida boshqariladi, ular tarmoqlanuvchi yoki takrorlanuvchi xususiyatga egadirlar.

Shartga muvofiq bajariladigan koʻrsatmalar bilan tuziladigan algoritmлар

Biroq turmushda uchraydigan juda koʻpgina jarayonlar esa shartlar asosida boshqariladi, ular tarmoqlanuvchi yoki takrorlanuvchi xususiyatga egadirlar.

Shartga muvofiq bajariladigan koʻrsatmalar bilan tuziladigan algoritmлар **tarmoqlanuvchi algoritmлар** deyiladi. Bunday algoritmлар bilan biz har kuni duch kelamiz. Koʻchadan oʻtishimiz

svetoforning yongan chirog'i rangiga, ko'chaga qanday kiyimda chiqishimiz ob- havoga, elektr jihozini ishlatishimiz uning qanday kuchlanishli manba'ga ulashimizga bog'liqdir.

Masalani echish jarayonida bir hil amallarning har bir yangi parametr uchun takrorlanishini talab qiluvchi algoritmlar **takrorlanuvchi algoritmlar** deyiladi.

Bunday algoritmlarga yil fasllarini, hafta kunlarini misol sifatida keltirish mumkin. Masalani hal qilish uchun tuzilgan algoritmlarni EHM yordamida amalga oshirish uchun, ya'ni EHM bilan bevosita ishlashdan oldin qanday bosqichlarni bajarish kerakligini ko'rib chiqaylik. Istalgan hayotiy yoki matematik, fizik va hokazo masala shartlarini ifoda qilish dastlabki ma'lumotlar va fikrlarni tasvirlashdan boshlanadi va ular qat'iy ta'riflangan matematik yoki fizik va hokazo tushunchalar tilida bayon qilinadi. So'ngra echishning maqsadi, ya'ni masalani echish natijasida ayni nimani yoki nimalarni aniqlash zarurligi ko'rsatiladi. Masala shartining aniq ifodasi masalaning matematik (fizik va hokazo) qo'yilishi deb ham ataladi va istalgan masalani echish eng avval uning qo'yilishidan boshlanadi. Masalani qo'yilishida boshlang'ich ma'lumotlar yoki argumentlar hamda qiymatlari aniqlanishi kerak bo'lgan kattaliklar, ya'ni natijalar ajratiladi. Masalani qo'yish uni echishning **birinchi bosqichi** bo'ladi. Masalani hal qilish uchun uni biror formulalar yoki jarayonlar orqali ifodalash kerak bo'ladi. Bu esa masalaning matematik modeli, ya'ni **ikkinchi bosqichi** hisoblanadi. Masalaning matematik modeli yaratilgandan so'ng, uni echish usuli izlana boshlanadi. Ayrim hollarda masalaning qo'yilishidan keyin to'g'ridan – to'g'ri, masalani echish usuliga o'tishga to'g'ri keladi. Bunday masalalar oshkor ko'rinishdagi matematik model bilan ifodalanmasligi mumkin. Bu bosqich masalalarni EHM da echishning **uchinchi bosqichini** tashkil qiladi. Navbatdagi bosqichda, ya'ni **to'rtinchi bosqichda**, masalani EHM dan foydalanib echish uchun uning algoritmi tuziladi. Algoritmini turli – tuman ko'rinishda yozish mumkin. Informatika kursining asosiy vazifalaridan biri ham algoritm tuzish usullarini o'rganishdan iboratdir. Bu jarayonda talabalarda, o'quvchilarda masalani echishning algoritmik usuli, ya'ni algoritmik fikrlash usuli vujudga keladi.

Algoritmning EHM da bajarilishi uchun bu algoritm dasturlash tilida yozilgan bo'lishi lozim. Masalani echishning bu bosqichi **beshtinchi bosqich** bo'lib, unda biron bir usulda yozilgan algoritm ma'lum bir dasturlash tiliga ko'chiriladi. Masalan, agar algoritm blok – sxema ko'rinishda tasvirlangan bo'lsa, uni Beysik dasturlash tiliga ko'chirish uchun har bir blokni tilning mos buyruqlari bilan almashtirish etarli. **Oltinchi bosqich** – dastur ko'rinishda yozilgan algoritmni EHM yordamida bajarish. Bu bosqich natija olish bilan tugallanadi. Bu bosqich dastur tuzuvchilar uchun eng qiyin hisoblanadi. Chunki, dasturni mashina xotirasiga kiritishda ayrim xatoliklarga yo'l qo'yilishi mumkin. Shuning uchun dasturni EHM xotirasiga kiritishda juda ehtiyot bo'lish kerak. Nihoyat, masalani echishning yakunlovchi **ettinchi bosqichi** olingan natijalarni tahlil qilishdir. Bu bosqich olingan natijalar qanchalik haqiqatga yaqinligini aniqlash maqsadida bajariladi.

Shunday qilib, masalalarni EHM da echish bosqichlari bilan tanishib chiqdik. Shuni ta'kidlash kerakki, har doim ham bosqichlar bir – biridan yaqqol ajralgan holda bo'lmasdan, bir - biriga qo'shilib ketgan bo'lishi ham mumkin.

Algoritmning quyidagi uch turi mavjud:

Hech qanday shart tekshirilmaydigan va tartib bilan faqat ketma – ket bajariladigan algoritmlar

chiziqli algoritmlar deb yuritiladi .

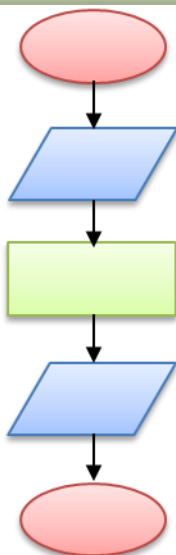
Shartga muvofiq bajariladigan ko'rsatmalar bilan tuziladigan algoritmlar

tarmoqlanuvchi algoritmlar deyiladi

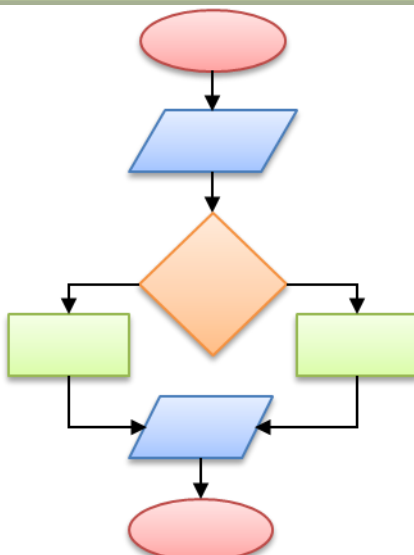
Masalani echish jarayonida bir hil amallarni takrorlanishini talab qiluvchi algoritmlar

takrorlanuvchi algoritmlar deyiladi.

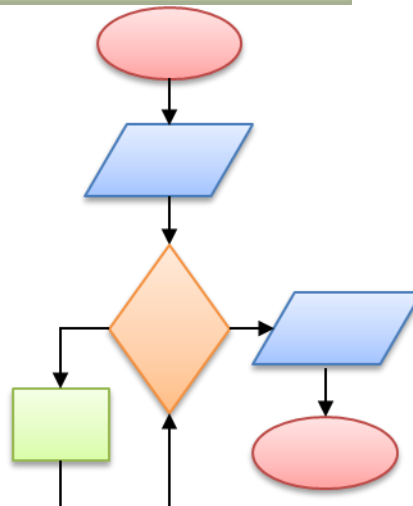
Chiziqli



Tarmoqlanuvchi



Takrorlanuvchi



ALGORITMLARNING BERILISH USULLARIGA DOIR MISOLLAR

Algoritmlar uch hil usulda qurilishi mumkin.

A) Algoritmni so'zlar orqali qurish. Bunda algoritmning har bir buyruq-amali ijroChiga tushunarli bo'lgan so'zlar orqali ifodalanadi.

1-misol: AB kesmani teng ikkiga bo'lish algoritmi.

1. Boshlansin.
2. Sirkulning bir uchi A nuqtaga qo'yilsin.
3. Radiusi AB bo'lgan aylana Chizilsin.
4. Sirkulning uchini V nuqtaga quyilsin.
5. Radiusi BA bo'lgan aylana Chizilsin.
6. Aylanalarning kesishish nuqtalarini CD kesma bilan birlashtirilsin.
7. CD va AB kesmalarning kesishish nuqtasi M ni belgilansin.
8. M nuqtani izlangan nuqta deb hisoblansin.
9. Ishni tugatilsin.

B) Matematik formulalar usuli. Bu usulda algoritmning har bir amali matematik formulalar yordamida hosil qilinadi. Algoritm amallarini ifodalashda oddiy matematik yozuvlardan foydalanish mumkin.

2 - misol: $ax^2 + bx + c = 0$ kvadrat tenglamaning echimi topilsin.

1. Boshlansin.
2. Aniqlansin a, b, c .
3. Hisoblansin $D := b^2 + 4ac$.
4. Agar $D < 0$ bo'lsa chiqarilsin «Echimi yo'q» ; 7 ga o'tilsin.

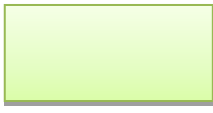
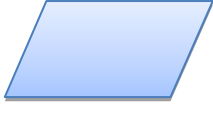
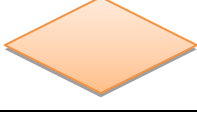
$$x_1 := \frac{-b + \sqrt{D}}{2a}; \quad x_2 := \frac{-b - \sqrt{D}}{2a}.$$

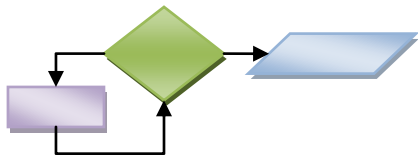
5. Hisoblansin
6. Chiqarilsin x_1, x_2 .
7. Ishni tugatilsin.

S) Blok-sxemalar usuli. Bunda algoritmning har bir buyrug'i maxsus geometric shakllar yordamida ifodalanadi. Blok-sxemalarni qurishda 1-jadvaldagi shakllardan foydalanish mumkin.

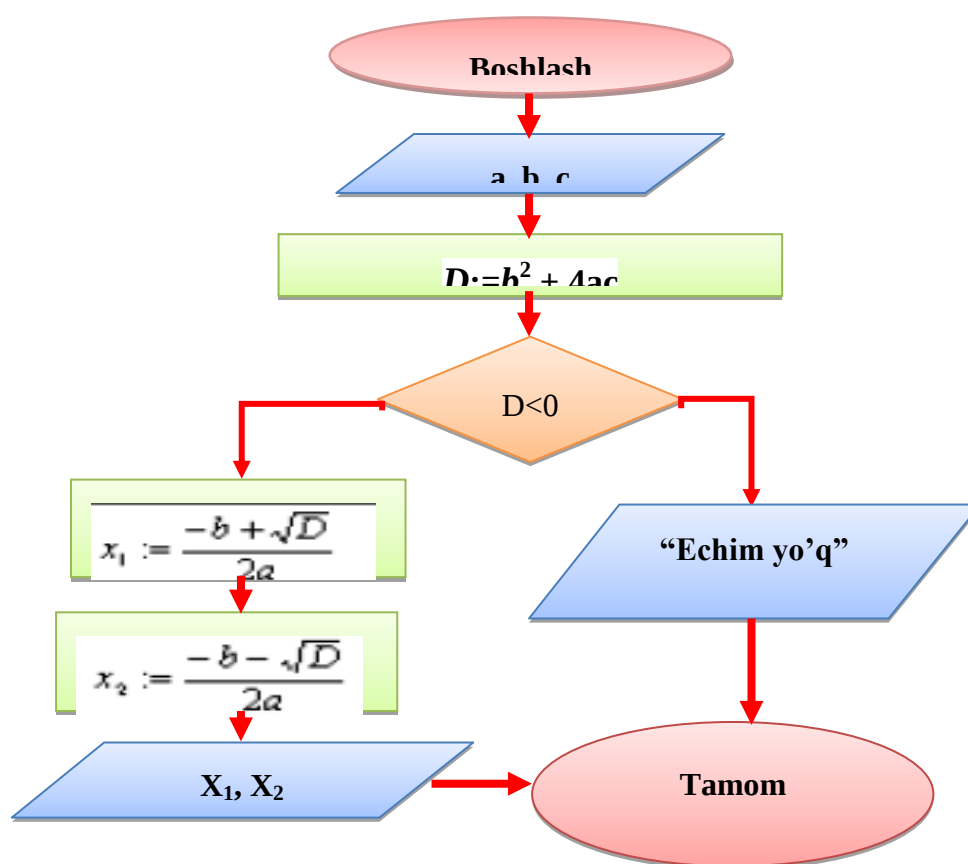
1-jadval

BLOK – SXEMA JADVALI

Geometrik figura	Vazifasi
	Algoritmning boshlash va tugatish;
	Arifmetik ifodalarni hisoblash;
	Boshlang'ich ma'lumotlarni kiritish;
	Boshqarishni shart asosida o'zgarishi;

	Takrorlanish jarayoni tasvirlash;
	Qism dasturlarga murojaat qilish;
	Natijalarni tashqi qurilmalarga chiqarish;
	Amallarnibajarish yo'nalishi;

2 - misol: $ax^2 + bx + c = 0$ kvadrat tenglamaning echimi topish blok – sxemasi tuzilsin.



Nazorat uchun savollar va topshiriqlar

17. Algoritm deyilganda nima tushuniladi? Algoritm so'zi qayerdan kelib chiqadi?
18. Algoritmning qanday xossalarini bilasiz? Algoritmni yozishning qanday
19. usullarini bilasiz?
20. Blok-sxema nima? Blok-sxemaning asosiy elementlarini ayting.
21. Algoritmning turlarini sanab o'ting.
22. Masalalarni EHM da yechish bosqichlarini sanab o'ting.
23. Al-Xorazmiyning xæti va ijodi xaqida nimalarni bilasiz?
24. Masalani echish algoritmini qanday usullar bilan ifodalash mumkin?
25. Algoritmga qo'yiladigan talablarni sanab o'ting.
26. Algoritmning qanday xossalarini bilasiz?
27. Masala echimi algoritmini tuzishda qanday qoidalarga amal qilish lozim?
28. CHiziqli algoritm deb nimaga aytiladi?
29. Tarmoqlanuvchi algoritmni ta'riflang va unga xætiy masalalar keltiring.
30. Takrorlanuvchi algoritmning vazifasini tushuntiring.
31. Algoritmni ifodalashda blok-sxemalardan foydalanishning afzalliklarini aytib bering.
32. Blok-sxema elementlarini chizib ko'rsating va ularning vazifalarini ayting.
33. Chiziqli tenglamalarni echish algoritmini tuzing.
34. Kvadrat tenglama ildizlarini aniqlash algoritmini ishlab chiqing va uni blok-sxemalar orqali ifodalang.
35. Xalqa yuzini aniqlash algoritmini yarating.

3. B/B/B jadvali

4-ilova

Bilaman	Bilmoqchiman	Bilib oldim

4. Uchta turdagi algoritmlarga mos «VENN» diagrammasini yaratish

2-AMALIY MASHG'ULOT: Dasturlash tillari va ularning klassifikatsiyasini o'rganish.

AMALIY MASHG'ULOTNI OLIB BORISH TEXNOLOGIYASI

Talabalar soni:	20 – 30 talaba
Vaqt:	2 soat
Dars shakli	Mavzu bo'yicha amaliy
Ma'ruza rejasi	6. Kompyuter dasturlari. 7. Dasturlash tili haqida tushuncha. Dasturlash tillari turlari. 8. Dasturlash tillarining sintaktik jihatdan turlari. 9. Kompilyatsiya va interpretatsiya qilinuvchi tillar. 10. Komp'yuterda masala echish bosqichlari.
O'quv mashg'ulotining maqsadi: Talabalarda dasturlash tillari, ularning turlari, dastur kompilyatsiyasi	

va interpretatorlari tushunchalarini hosil qilish.	
Tayanch tushuncha va iboralar	Kompyuter dasturlari; amaliy dasturlar; tizimli dasturlar; uskunaviy tizimlar; Dasturlash; dasturchi; Dasturlash tillari turlari; assembler tili; fortran tili; beysik tili; quyi darajadagi DT; o'rta darajadagi DT; yuqori darajadagi DT; Kompilyatsiya va interpretatsiya qilinuvchi tillar; Komp'yuterda masala echish bosqichlari.
Pedagogik vazifalar:	O'quv faoliyati natijalari:
Kompyuter dasturlari haqida tushuncha beradi	Kompyuter dasturlarining 3 turi haqida tushunchaga ega bo'ladi
Dasturlash tili haqida tushuncha berish.	Dasturlash tillari va ularning turlari haqida bilimga ega bo'ladilar.
Dasturlash tillarining sintaktik jihatdan turlari.	Dasturlash tillari sintaktik jihatdan quyi, o'rta va yuqori darajaga bo'linish haqida ma'lumot oladilar.
Kompilyatsiya qilinuvchi va interpretatsiya qilinuvchi tillar haqida ma'lumot berish.	Dasturlash tillari kompyuterda bajarilishiga qarab kompilyatsiya qilinuvchi va interpretatsiya qilinuvchi tillarga bo'linishi haqida ma'lumotga ega bo'ladilar.
Komp'yuterda masala yechish bosqichlarini tushuntirish.	Komp'yuterda masala yechish bosqichlari haqida bilimga ega bo'ladilar.
O'qitish vositalari	ma'ruza matni, tarqatma material, komp'yuter slaydlari, doska
O'qitish usullari	ma'ruza, tushuntirish, instruksiya berish, namoyish, Blits-so'rov, "Aqliy hujum", B/B/B (Bilaman/ Bilmoqchiman/ Bilib oldim) usuli
O'qitish shakllari	frontal, kollektiv ish
O'qitish sharoiti	komp'yuter, mul'timedia proektor (yoki proektor va LCD panel)

AMALIY MASHG'ULOT TEXNOLOGIK KARTASI

Ish bosqichlari	O'qituvchi faoliyatining mazmuni	Tinglovchi faoliyatining mazmuni
1-bosqich. Mavzuga kirish (10min)	1.2. O'quv mashg'uloti mavzusi, rejasiva maqsadi tushuntiriladi (1 - ilova)	Mavzu nomini yozib oladilar
	1.2. Blits-so'rov usulida mavzu bo'yicha ma'lum bo'lgan tushunchalarni faollashtiradi. (1-ilova)	Savollarga javob beradilar
2 -bosqich.	2.3. Mavzu rejalari asosida tayanch tushunchalar bilan tanishtiradi. (2-ilova)	Mavzu rejasini yozib oladilar Tinglaydilar.

Asosiy bo'lim (60 min)	2.4. Mavzuning 1- va 2- reja savollari bo'yicha "Aqliy hujum" uyushtiriladi. Har bir talaba fikri tinglanadi. Har bir savol muhokamasidan so'ng to'g'ri fikrlar tasdiqlanadi. Jarayon komp'yuter slaydlarini namoyish qilish bilan olib boriladi. (3-ilova)	Tinglaydilar. "Aqliy hujum" da ishtirok etadilar
3-bosqich. Yakunlovchi (10 min)	3.5. Mavzu bo'yicha yakunlovchi xulosalar qiladi. Mavzu bo'yicha olingan bilimlarni qayerda ishlatish mumkinligi ma'lum qiladi.	Savollar beradilar
	3.6. Mavzu bo'yicha mustaqil o'rganish uchun topshiriqlar beradi.	Mustaqil o'rganish uchun topshiriqlarni yozib oladilar
	3.7. Mavzu bo'yicha bilimlarni chuqurlashtirish uchun adabiyotlar ro'yxatini beradi	Yozadilar
	3.8. Mavzuning keyingi savollari buyicha tayyorlanib kelish topshiriladi.	Yozadilar

Dasturlash tillari va ularning turlari. Dasturlash tillari va ularning klassifikatsiyasi, mashinaga mo'ljallangan va protseduraga mo'ljallangan dasturlash tillari, yuqori darjali dasturlash tillari, interpretatorlar va kompilyatorlar, dasturlarni translyasiyalash.

1-ilova

6. Kompyuter dasturlari.
7. Dasturlash tili haqida tushuncha. Dasturlash tillari turlari.
8. Dasturlash tillarining sintaktik jihatdan turlari.
9. Kompilyatsiya va interpretatsiya qilinuvchi tillar.
10. Komp'yuterda masala echish bosqichlari.

TAYANCH TUSHUNCHALAR:

2-ilova

Kompyuter dasturlari; amaliy dasturlar; tizimli dasturlar; uskunaviy tizimlar; Dasturlash; dasturchi; Dasturlash tillari turlari; assembler tili; fortran tili; beysik tili; quyi darajadagi DT; o'rta darajadagi DT; yuqori darajadagi DT; Kompilyatsiya va interpretatsiya qilinuvchi tillar; Komp'yuterda masala echish bosqichlari.

Komp'yuter dasturlari uch turga bo'linadi.

Komp'yuter dasturlari- amaliy, tizimli, uskunaviy tizimlarga bo'linadi.

Amaliy dasturlar- bu foydalanuvchi bevosita ishlashi uchun mo'ljallangan dasturlardir.

Tizimli dasturlar- bu kompyuter qurilmalarining ishchi xolatini nazorat qiluvchi va boshqaruvchi dasturlar. Bundan tashqari u ko'p tizimli qobiqli dasturlar bo'lib, foydalanuvchini kompyuter bilan qulay muloqatini ta'minlaydi. Tizimli dasturning asosiy sinfi, bu drayver bo'lib,

u OT xususan tashqi yoki ichki qurilmalar bilan ishlash imkonini beradi. Hozirgi kunda dasturlash tilining bir necha turlari mavjud:

Dasturlash tillari hakida

Tezkor elektron xisoblash mashinalarining paydo bo'lishi **dasturlash tili** deb ataluvchi turli-tuman belgilar sistemalarining paydo bo'lishiga olib keldi. SHunday qilib, hisoblash mashinalarida bajarilishi kerak bo'lgan jarayonlarni tavsiflash uchun qo'llaniladigan belgilar (simvollar) sistemasini **dasturlash tili** deb yuritamiz.

EHMLar uchun dastur tuzish va uni ishlatish juda murakkab va ko'p mehnat talab etadigan jarayon bo'lib, uning uchun ko'p aqliy mexnat va vaqt talab qilinadi. Shuning uchun yangi algoritmik tillarni yaratuvchilar dasturlashni sodda va xayotimizning turli sohalarida mexnat qiluvchilar uchun tushunarli bo'lishiga xarakat qilishadi. Hozirgi kunda dasturlash tillarining bir necha turi mavjud:

Bundan ko'rinadiki dasturlash tillarining soni bir necha yuzdan ortiq ekanligi.

Beysik dasturlash tili 1964 y. AQSh ning Dortmund kollejining ilmiy xodimlari Jon Kemeni va Tomes Kurts tomonidan yaratilgan. Bu til turli hisoblashlarga doir masalalarni kompyuter bilan muloqat holda xal qilish uchun yaratildi.

Basic so'zi Beginners Allpurpose Symbolic Instruction Code –degan kengaytmasidan iborat bo'lib, ya'ni boshlovchilar uchun mo'ljallangan ko'p maqsadli, belgili ko'rsatmalar tili degan ma'noni bildiradi. Ushbu til kompyuter xotirasiga qo'yiladigan talablarning juda kamligi sababli, ShK larda ishlatiladigan til bo'lib qoldi. Bu tilning bir necha ko'rinishi mavjud bo'lib, maqsad foydalanuvchilarning muloqotida osonlik tug'dirish.

Assembler tili inglizchasiga Assembler- bu EHM uchun mashina kodlarida dastur yozish ishini yengillashtirish maqsadida 40 yillar ohiri 50 yillarning boshida yaratilgan.

Fortran tili –inglizcha Formula Translator – formulani translyatsiya qilish so'zlaridan qisqartirib olingan bo'lib, formula tarjimoni degan ma'noni bildiradi va bu til muhandislik, ilmiy-texnik masalalarni yechishga mo'ljallangan til hisoblanadi. Ushbu tilning qator variantlari yaratilgan, ulardan eng mashhurlari Fortran II va Fortran IV. Fortran tilining asosiy g'oyalari keyingi Algol-60, 68, PL-1 va boshqa tillarda rivojlantirilgan. Bu til 1954 yil yaratilgan.

Shunday maqsadni 1970 yili Shvesiyalik Oliy texnika o'quv yurtining professori Niklaus Virt o'z oldiga qo'ydi va u tomonidan tavsiya etilgan algoritmik tilni yaratdi va ulug' fransuz olimi B.Paskal (1623-1662) nomi bilan atadi.

Dasturlash tillarining sintaktik jihatdan turlari

Dasturlash tillarining sintaktik jihatdan turlari 3 turga(Informatika darsliklariga asosan) bo'linadi:



Quyi darajadagi dasturlash tili “Mashina tili” deb ham ataladi. Ushbu tilda dasturlar to‘g‘ridan-to‘g‘ri Operativ Xotira(OX) katakchalari va protsessor reyestrlari bilan ishlab tuziladi. Ushbu tildagi buyruqlar Markaziy Protsessor(MP)ning operatsiyalariga to‘g‘ri keladi. Buyruqlar ikkilik kodda yozilgan:

1: 10110000 01100001

2: (AL reyestriga 16-lik 61 sonini joylash)

Bir paytlar perfokartalar yordamida aynan mashina tilida dasturlar yozilgan.

O‘rta daraja dasturlash tillarida protsessor buyruqlarini mnemonik kodlarga(buyruqqa mos qisqartirilgan so‘zlar) almashtirilgan. Assembler tili bunga misoldir. O‘rta darajadagi dasturlash tillarida ham bir protsessor operatsiya deyarli bir buyruqqa mos keladi. Masalan, yuqoridagi mashina kodi Assemblerda quyidagicha yoziladi:

1: MOV AL, 61h

2: (Yuqoridagidan ko‘ra tushunarliroq)

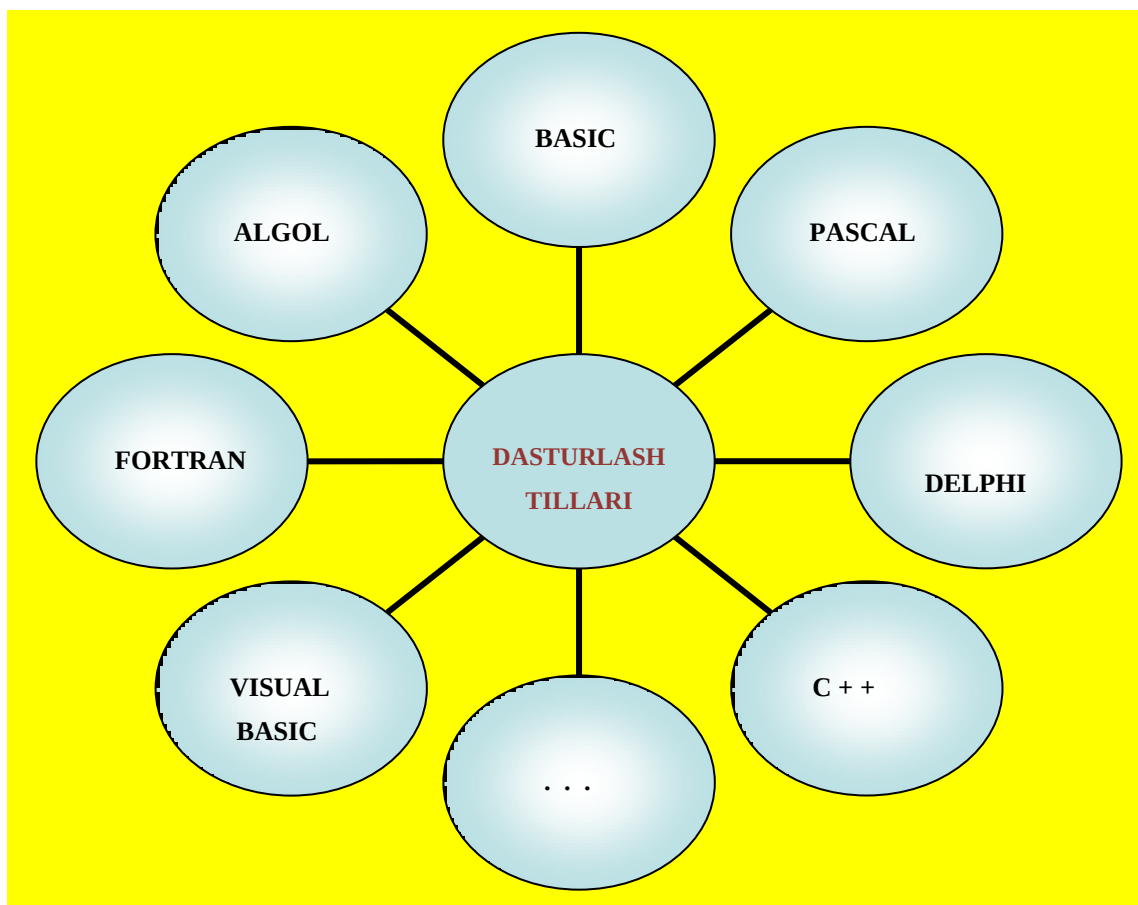
Ko‘rib turganingizdek, Assembler tili mashina kodidan bir pog‘ona yuqorida turadi xolos.

Yuqori darajadagi dasturlash tillari esa, asosan, dasturlash jarayonini tezlashtirish uchun yaratilgan. Shuni eslatib o‘tish lozimki, har qanday dastur bajarilishidan oldin mashina kodiga o‘tkaziladi. Ushbu darajadagi dasturlash tillarida yozilgan dastur ma’lum ma’noli so‘zlardan(odatda ingliz tilidagi) tashkil topadi. Masalan:

1: if (var == 6) {

2: print "Salom";

Ko‘rib turibsizki, dastur qismi ingliz tilidagi ma’noli so‘zlardan tashkil topgan. Hozirgi zamonaviy tillarning barchasi yuqori darajaga mansub.



Kompilyatsiya va interpretatsiya qilinuvchi tillar



Dasturlash tillari komputerda bajarilishiga qarab kompilyatsiya qilinuvchi va interpretatsiya qilinuvchi tillarga bo'linadi.

Kompilyatsiya qilinuvchi dasturlash tillarida dastur kodi kompilyator tomonidan mashina kodiga o'tkaziladi. Operatsion tizim(OT) esa, shu kodni to'g'ridan-to'g'ri ishlataveradi. Kompilyatsiya jarayoni komputer protsessori va OT talablariga mos ravishda amalga oshiriladi. Shuning uchun, bir OT uchun kompilyatsiya qilingan dasturning mashina kodi ikkinchi OT da ishlamaydi. Ushbu turdagi tillarga quyidagilarni misol qilib keltirishimiz mumkin: C, C++, Pascal va h.k.

Microsoft Windows OTlarida kompilyatsiya qilingan dastur nomi *.exe ko'rinishidagi fayl bo'ladi. Linux, Unix(va shularning davomchilari) kabi OT larda esa fayl kengaytmasining ahamiyati yo'q.

Kompilyatsiya qilinuvchi dasturlash tillarining asosiy yutuqlaridan biri — u OT dan boshqa biror dastur yoki kutubxona(Library, mas. DLL) o'rnatishni talab qilmaydi. Bundan tashqari, interpretatsiya qilinuvchi tillarga nisbatan ancha tez ishlaydi.

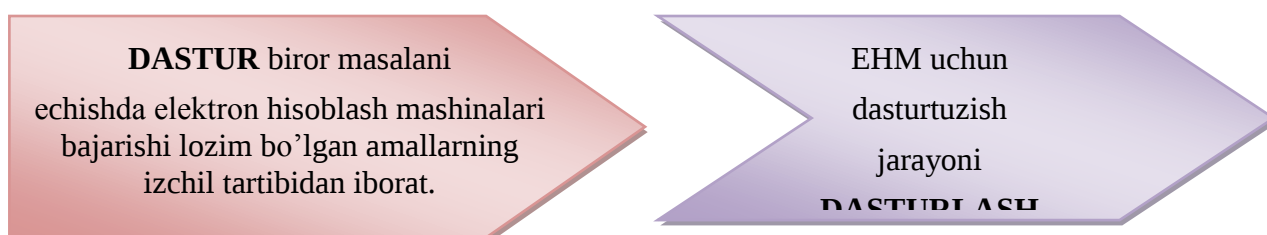
Interpretatsiya qilinuvchi dasturlash tillarida tuzilgan dastur kodi kompilyatsiya qilinmaydi. Ushbu turdagi dasturni ishlatishdan oldin dastur kodi interpretatsiya qilinadi. Interpretatsiya

qilinuvchi dasturlash tillarida tuzilgan dastur mos interpretator oʻrnatilgan kompyuterlardagina ishlaydi. Ushbu turdagi dasturlash tillariga PHP, Python, Ruby kabi tillar kiradi.

Interpretatsiya qilinuvchi dasturlash tillari kompilyatsiya qilinuvchilaridan, asosan, yozilgan dasturning deyarli hamma platformalarda ishlashi bilan ajralib turadi. Dastur biror turdagi OT yoki protsessor uchun yozilmaydi — faqat interpretatorgina turli platformalar uchun yoziladi.

Interpretatsiya qilinuvchi dastur kodi bajarilishidan oldin interpretator tomonidan oraliq kodga “kompilyatsiya” qilinadi. Shu oraliq kod interpretator tomonidan bajariladi. Python kabi tillar oraliq kodni saqlab qoʻyadi, dastur kodi oʻzgarmaguncha shu oraliq kodni ishlatadi.

Dastur biror masalani echishda elektron hisoblash mashinalari bajarishi lozim boʻlgan amallarning izchil tartibidan iborat. EHM uchun dastur tuzish jarayoni dasturlash deyiladi. Dasturlash echilishi kerak boʻlgan masala algoritmini EHM tiliga, yaʼni «mashina tili»ga oʻtkazishdir. EHM uchun dastur tuzish – masalani echish usulini mashina buyruqlarining shunday majmui (dasturi)ga, keltirish demakki, bu buyruqlar xotiraga joylashib, tartib bilan amalga oshadi va tegishli hisoblashlarni bajaradi.



KOMPYUTERDA MASALA YECHISH BOSQICHLARI (KMEB).

1. Masalaning qoʻyilishi- bunda berilgan masalaning toʻgʻri qoʻyilganligi va uni yechish uchun qanday maʼlumotlar kerakligi va qanday natija olinishini bilish kerak.
2. Masalaning matematik modelini tuzish-bu bosqichda matematik formula tanlanadi matematik modeli tuziladi.
3. Masalani yechishni sonli usuli-bu bosqichda Hosil qilingan mat-k masalaning yechish usuli Tanlanadi va buning uchun sonli usuldan foydalaniladi.
4. Algoritmini tuzish-bunda mashina tushunadigan biror bir algoritmik tilda dGʻtuzish.
5. Dasturni kiritish va chiqarish. Mavjud muhitdan foydalangan xolda.

Har bir til ham oʻz alfavitiga ega.

Agar tuzilgan programmada alfavitda yoʻq xarf yoki belgilar uchrasa, mashina bunday belgini tushunmaganligi haqida xabar beradi.

Biz ishlayotgan har qanday dastur biror -bir shaxs tomonidaaan yozilgan boʻlib, ular dasturchilar deb aytiladi. Dasturchilar dasturni maxsus tillarda yozishadi –ularni dasturlash tillari deb aytiladi. Dastur koʻrinishlariga nisbatan quyi va yuqori dasturlash tillari sinflariga boʻlinadi. Quyi dasturlash tillariga misollar “quyi darajadagi dasturlash” assambler tilini keltirishimiz mumkin. Yuqori dasturlash tillariga Paskal, TSi, SQQ, S# tillarini keltirishimiz mumkin.

BBB JADVALI

BILAMAN	BILMOQCHIMAN	BILIB OLDIM



BLITS – SO'ROV:3-ilova

8. Kompyuter dasturlari deganda nimani tushunasiz?
9. Dasturlash tillarining qanday turlarini bilasiz va ularning roli.
10. Dasturlash tillarining bir-biridan farqi?
11. Zamonaviy dasturlash tillari qaysilar va ularning xususiyatlari?
12. Dasturlarni kompilyatsiya qilish.
13. Dasturlarni interpretatsiya qilish.
14. OTlarida kompilyatsiya qilingan dastur nomi kengaytmasi qanday bo'ladi

3-AMALIY MASHG'ULOT: Paskal dasturlash tilida chiziqli dastur tuzish va natija olish

AMALIY MASHG'ULOTNI OLIB BORISH TEXNOLOGIYASI

Vaqt – 2 soat	Talabalar soni: 25-30 nafar
O'quvmashg'ulotishakli	Nazariy bilimlarni mustahkamlash va amaliy ko'nikmalarni shakllantirishamaliy mashg'uloti
O'quv mashg'ulotirejasi	1. Paskal dasturini ishga tushirish. 2. Paskal muhiti bilan tanishish. 3. Paskalda chiziqli dasturni kiritish va natija olish. 4. Paskalda dastruni saqlash, xotiradagi dasturni yuklash.
O'quv mashg'ulotining maqsadi: Amaliy mashg'ulot davomida nazariy bilimlarga asoslangan amaliy ko'nikmalarni shakllantirish.	
O'quv faoliyatining natijalari:	
Pedagog vazifalari: ▪ Paskal dasturini yuklash. ▪ Paskal dasturi oynasini boshqarish ▪ Paskal oynasiga dastur kiritish, natija olish, tahlil qilish. ▪ Paskalda kiritilgan dasturlarni saqlash, xotirada mavjud dasturlarni yuklash.	O'quv natijalari: •Dasturni ishga tushirish usullari bilan tanishtiradi. • Paskal muhitida ishlash, menyu buyruqlari, funksional tugmachalardan foydalanishni o'rgatadi. •Dastur kiritish, hatoliklarni bartaraf etish, natija olish texnologiyalarini o'rgatadi. • Paskal muhitidagi menyu buyruqlari kiritilgan dasturni xotiraga saqlash, xotirada mavjud dasturdan foydalanish uchun yuklash kabi imkoniyatlari orqali tushuntiradi.
O'qitishning usul, texnikasi:	suxbat, baxs-munozara muhokama qilish, fikr almashish dastur tuzish.
O'qitish vositalari:	Tarqatma materiallar, Pascal dasturi, slaydlar
O'qitish shakli:	Frontal, kichik guruhlarda hamkorlikda ishlash texnologiyasi.
O'qitish shart – sharoitlari:	Texnik vositalarni qo'llashga mo'ljallangan auditoriya. Proektor

AMALIY MASHG'ULOTNING TEXNOLOGIK KARTASI

Bosqichlar va aqti	Faoliyat mazmuni	
	o'qituvchi	talaba
1-bosqich. Kirish (15 min)	1.1. Auditoriyaning va talabalarning darsga tayyorlik darajasi tekshiriladi. Ma'ruza mashg'ulotida o'tilgan nazariy bilimlar yuzasidan faollashtiruvchi savollar bilan talabalarga murojat etadi (1- ilova). 1.2. Mavzuni, uning maqsadi, kutiladigan o'quv natijalarini e'lon qiladi. Mashg'ulot yakka, juftlikda, hamkorlikda ishlash texnologiyasini qo'llagan holda o'tishni ma'lum qiladi. Baholash turlari, mezonlarini e'lon qiladi.	1.1. Savollarga javob beradilar. 1.2. Mavzuni yozadi va «Nilufar guli» jadvalini to'ldiradi.
2-bosqich asosiy (55 min)	2.1. Dasturlash tillari haqidagi sxemani to'ldirish. (1 - ilova). Kattaliklar, o'zgaruvchi va o'zgarmaslar tushunchasini kiritadi. (2 - ilova) 2.3. Paskalda o'zgaruvchi tiplari, ularning nomlanishini tushuntiradi. (3 - ilova) 2.4. Masalani EHMda echish bosqichlari. (4 - ilova) 2.5. faollashtiruvchi savollar. (5 - ilova) 2.6. CHiziqli dasturlar tuzishda ishlatiladigan operatorlar bilan tanishtiradi va amaliy ishni bajaradi. (6 - ilova)	Eshitadilar kerakli ma'lumotlarni yozib oladilar va sxemalarni chizib oladilar, kerakli qismlarini to'ldiradilar.
3-bosqich YAkuniy (10 min)	3.1. Mashg'ulotni yakunlaydi talabalarni baholaydi (7-ilova) va faol ishtirokchilarni rag'batlantiradi. 3.2. Mustaqil ish sifatida misol va mashqlar to'plamidagi misollarni ishlab kelishni topshiradi.	3.1. Eshitadilar. 3.2 Topshiriqni oladilar.

Faollashtiruvchi savollar:

1. *Dastur xamda dasturlau deganda nimani tuuunasiz?*
2. *Paskal dasturlau tilining alifbosiga nimalar kiradi?*
3. *Kattalik deb nimaga aytiladi?*
4. *Qanday qiymat tiplarini bilasiz?*
5. *Sonli tipni aytib bering.*

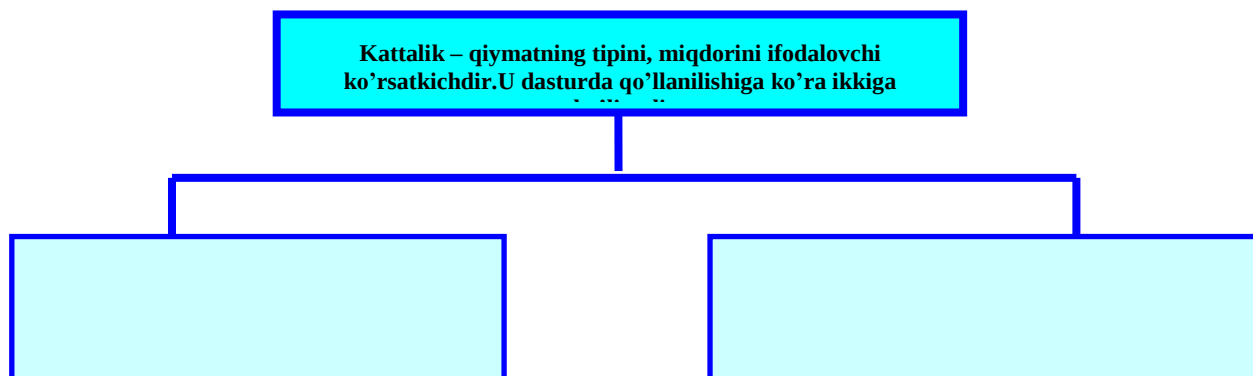
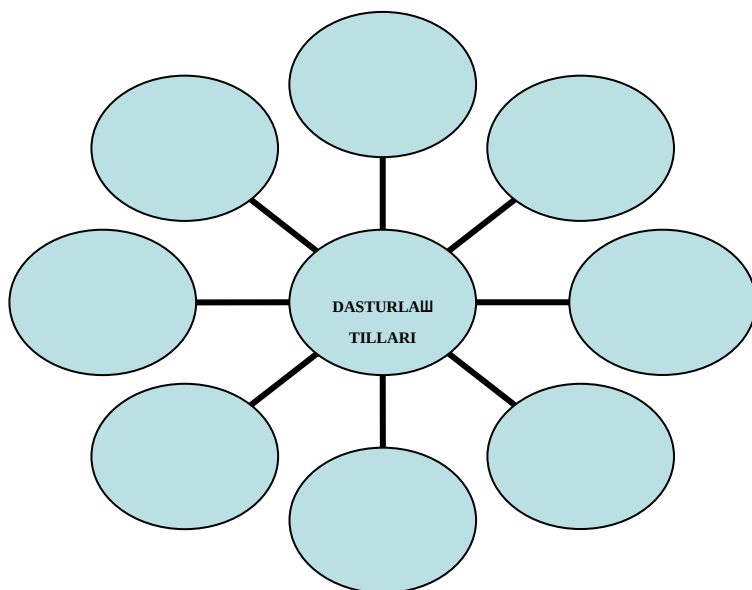
Guruhni kichik ijodiy guruhlarga ajratish va reklamalarni tayyorlash.

Reklamaning shartlari :

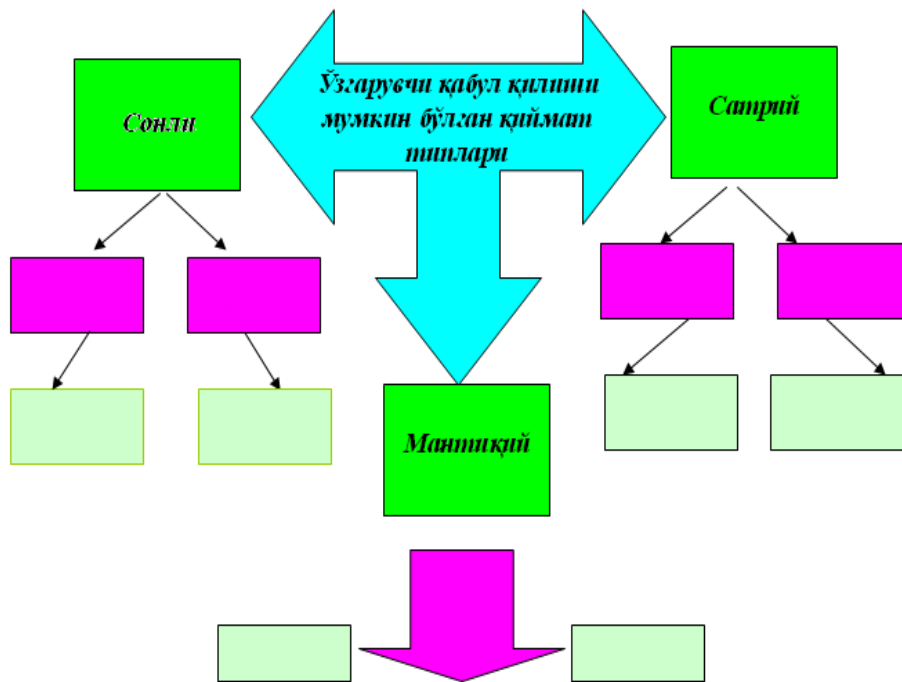
1. Har bir guruh o'z nomini, shiorini, emblemasini ishlab chiqishi kerak.
2. Reklamaning axborot materialini yozishi talab etiladi.
3. Axborot materialida paskal dasturlash tilining mazmun-mohiyati yoritilgan bo'lishi talab etaladi.
4. Reklamaning taqdimotini uyushtirishi kerak.

Harbir ijodiy guruhlarga topshiriqlar berilib, bajarilishi tahlil qilinib, mos jadvalda qayd etilib borilishi maqsadga muvofiqdir.

Qanday dasturlash tillarini bilasiz?

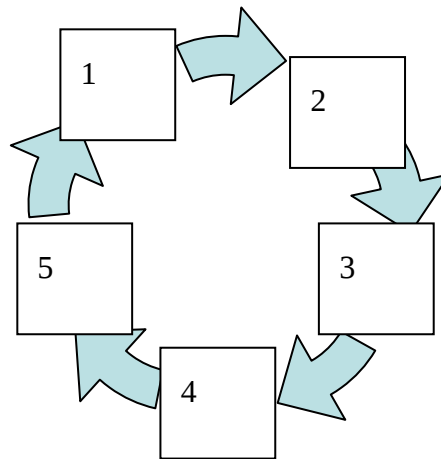


Паскалда қиймат типлари, номлари



4-ilova

Masalani EHMda echish bosqichlarini tartib raqamlarga mos holda yozing va izoh bering.



5 – ilova

FAOLLASHTIRUVCHI SAVOLLAR:

1. Kattalik nima, uning qanday turlari mavjud?
2. Kattalikning qanday tiplarini bilasiz?
3. Operator deb nimaga aytiladi?
4. Dastur deb nimaga aytiladi?
5. O'zlashtirish operatorini tushuntirib bering.
6. Argumentlarni kiritish operatorini tushuntiring.
7. Natijani chop etish operatorini tushuntiring.

Jadvalni to'ldiring

6-ilova

№	Chiziqli algoritmgga doir masalalar hal qilishda qo'llaniluvchi operatorlar	Tarmoqlanuvchi algoritmgga doir masalalarni hal qilishda qo'llaniluvchi operatorlar	Takrorlanuvchi algoritmgga doir masalalarni hal qilishda qo'llaniluvchi operatorlar
1			
2			
3			
n			

Insert jadvalini to'ldirish tavsiya etiladi.

O'qish jarayonida olingan ma'lumotlarni alohida o'zlari tizimlashtiradilar - jadval ustunlariga "kiritadilar" matnda belgilangan quyidagi belgilarga muvofiq:

"V"- men bilgan ma'lumotlarga mos;

"-" - men bilgan ma'lumotlarga zid;

"+" - men uchun yangi ma'lumot;

"?" - men uchun tushunarsiz yoki ma'lumotni aniqlash, to'ldirish talab etiladi.

Insertjadvali

V	+	-	?

Amaliy ishini o'tkazish uchun ko'rsatmalar:

- I. O'quvchilar o'qituvchi rahbarligida nazariy ma'lumotlar va mavzu yuzasidan yaratilgan dastur bilan tanishtiriladi (10-15 daq).
- II. O'quvchilarga mavzu bo'yicha masalalarni dasturini tuzib, natijalar olish taklif etiladi.
- III. O'quvchilar amaliy ish masalalarini o'qituvchiga topshiradilar va baholanadilar.

Amaliy mashg'ulotning 1-amaliy ishi.

Mavzu: TURBO PASCAL dasturlash tilida chiziqli dasturlar tuzish va natija olish.

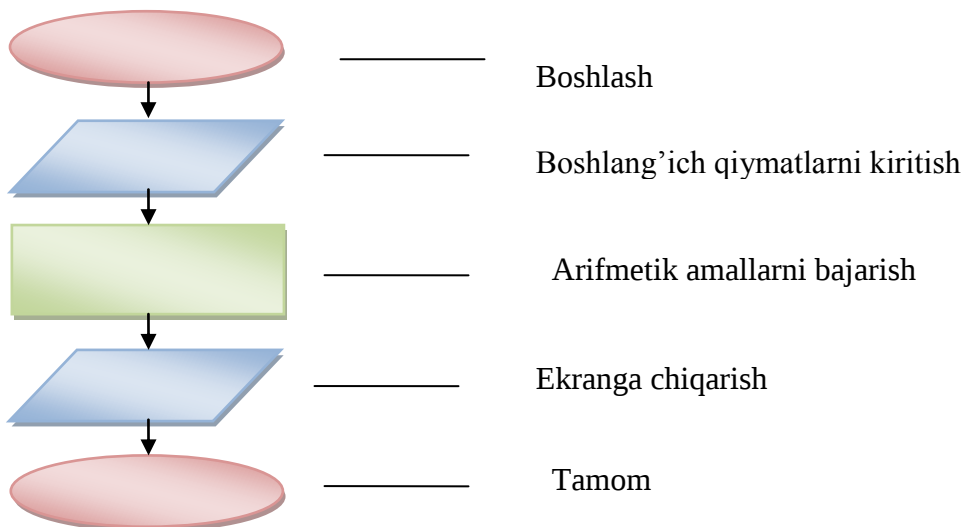
MAQSAD: Ma'ruzada olingan bilimlarni mustaxkamlash, dasturlash ko'nikmalarini shakllantirish, yechimlarning turli variantlarini aniqlash.

TOPSHIRIQLAR: Berilgan variantlarga ko'ra dasturlar tuzish va ularni Turbo-Pascal dasturlash muhitida bajarish hamda himoyalash.

NAZARIY MA'LUMOT: chiziqli dastur tuzish uchun quyidagi operatorlar zarur bo'ladi. O'zlashtirish operatori, read(a); va readln(a); -kiritish operatorlari write(a); va writeln(a);- Chiqarish operatori.

Chiziqli dastur tuzish uchun quyidagilarni eslab olish kerak:

- Chiziqli dasturda buyruqlar ketma-ket bajariladi.
- Chiziqli dastur algoritmi quyidagicha ko'rinish olishi mumkin:



Kerakli dasturiy va metodik qo'llanmalar: Turbo-Pascal dasturlash muhiti, tarqatma materiallar, Turbo-Pascal dasturida tuzilgan chiziqli dasturlar.

AMALIY ISHNI BAJARISH TARTIBI.

I. O'quvchilar o'qituvchi rahbarligida nazariy ma'lumotlar va mavzu yuzasidan yaratilgan dastur bilan tanishtiriladi (10-15 daq).

1-masala. Parallelopipedning hajmini hisoblash dasturi tuzilsin. Quyida dasturni ishlash jarayoni ko'rsatilgan. (Foydalanuvchi tomonidan kiritilgan qiymatlar qalin shriftlarda ko'rsatilgan).

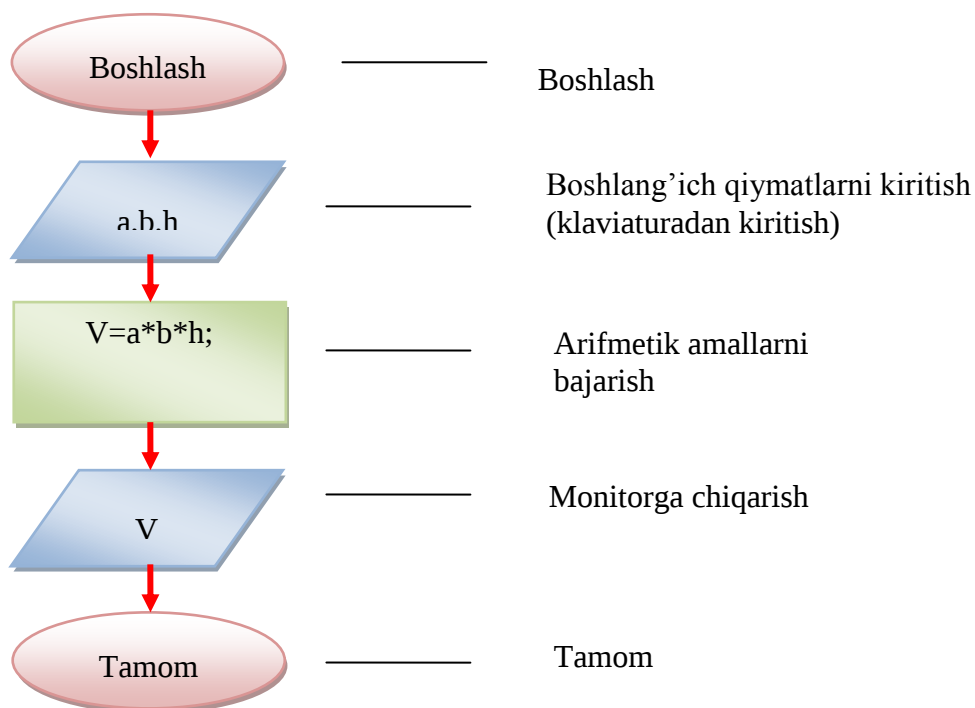
Parallelopipedning hajmini hisoblash.

Eni (sm): **7.5**; Bo'yi (sm): **2.5**; Balandligi (sm): **3**; Parallelopiped hajmi: 56.25 kub.sm

Ushbu masalani yechish uchun quyidagi bosqichlardan o'tish kerak.

- Matematik formula.
- Algoritm tuzish.
- Dastur tuzish.

$V=a*b*h$ formuladan foydalaniladi.



```

Program M_1;
  Var      V,a,b,h:real;
Begin
  Writeln('Parallelopipedning hajmini hisoblash. '); Write('Eni (sm): ');
  Readln(a);      {kiritish operatori}
  Write('Bo'yi (sm): '); Readln(b);
  Write('Balandligi (sm): '); Readln(h);

```

II. O'quvchilarga quyidagi masalalarni dasturini tuzib, natijalar olish taklif etiladi.

1. Parallelopipedning hajmini topish dasturi tuzilsin. Quyida dasturni ishlash jarayoni ko'rsatilgan.

(Foydalanuvchi tomonidan kiritilgan qiymatlar qalin shrift bilan ko'rsatilgan).

Parallelopiped hajmini hisonblash.

Qiymatlarni kiriting: Bo'yi (sm) → **9**; Eni (sm) → **7.5**; Balandligi (sm) → **5**; Hajmi: 337.50 kub.sm.

2. Kubni hajmini hisoblash dasturi tuzilsin. (Foydalanuvchi tomonidan kiritilgan qiymatlar qalin shrift bilan ko'rsatilgan).

Kubni hajmini hisoblash dasturi.

Kubni tomoni uzunligini (sm) kiriting va <Enter> ni bosing. → **9.5**; Kub hajmi: 857.38 kub.sm;

3. Bir nechta daftar va qalamni sotib olingandagi harajatni hisoblash dasturi tuzilsin.

Quyida dasturni ishlash jarayoni ko'rsatilgan.

Harajatni hisoblash:

Daftar narxi (so'm) → **100**; Daftarlar soni → **5**; Qalam narxi (so'm) → **50**; Qalamlar soni → **3**; Jami harajat: 650 so'm; Dasturni tugatish uchun <Enter> ni bosing.

4. *Bir necha kilogrammdan iborat olmani narxini hisoblash dasturi tuzilsin.*

Quyida dasturni ishlash jarayoni ko'rsatilgan.

Harajatni hisoblash dasturi.

Boshlang'ich qiymatlarni kiriting: ; Bir kg olma narxi(so'm) → **425**; Necha kg → **2.5**; Harajat: 1062.5.

5. *Agar uchburchakning ikki tomoni va ular orasidagi burchak ma'lum bo'lsa, uchburchak yusini topish dasturi tuzilsin. Quyida dasturni ishlash jarayoni ko'rsatilgan.*

Uchburchak yuzini topish:

Bit satrda probel orqali uchburchakni ikki tomonini (sm) kiriting → **25 17**; Ular orasidagi burchak → **30**

Uchburchak yuzini: 106.25 sm.kv.;

6. *Elektr zanjirdagi tok kuchni hisoblash dasturi tuzilsin.*

Quyida dasturni ishlash jarayoni ko'rsatilgan.

Elektr zanjirdagi tok kuchni hisoblash:

Tok quvvati (volt) → **36**; Qarshilik (OM) → **1500**; Tok kuchi: 0.024 Amper.

7. *Minutda berilgan qiymatlarni soatga aylantirish dasturi tuzilsin.*

Quyida dasturni ishlash jarayoni ko'rsatilgan.

Minutda vaqtni kiriting → **190**; 190 minut bu – 3 soat 10 minut.

8. *Yuguruvchni tezligini topish dasturi tuzilsin. Quyida dasturni ishlash jarayoni ko'rsatilgan.*

Yuguruvchni tezligini topish.

Masofani kiriting (metr): **1000**; Vaqti (min.sekund): **3.25**; Yuguruvchni tezligini: 17.56 km/soat.

9. *Uchburchakning asosi uzunligi va balandligiga ko'ra uchburchakni yuzasini topish dasturi tuzilsin*

Quyida dasturni ishlash jarayoni ko'rsatilgan.

Boshlang'ich qiymatlarni kiriting:

Asosi (sm) → **8.5**; Balandligi (sm) → **10**; Uchburchak yuzi: 42.5 sm.kv.

10. *a burchak radian o'lchovida berilgan, uni gradusda ifodalovchi dastur tuzing.*

III. Amaliy ish himoyasi.

7-ilova.

Amaliy ishni baholash mezonlari va ko'rsatkichlari (ball)

№	F.I.SH.	Nima uchun (1,0)	1-savol (1,0)	2-savol (1,0)	1-misol (1,0)	2-misol (1,0)	Jami ballar (5,0)
1.							
2.							

3.							
4.							
5.							

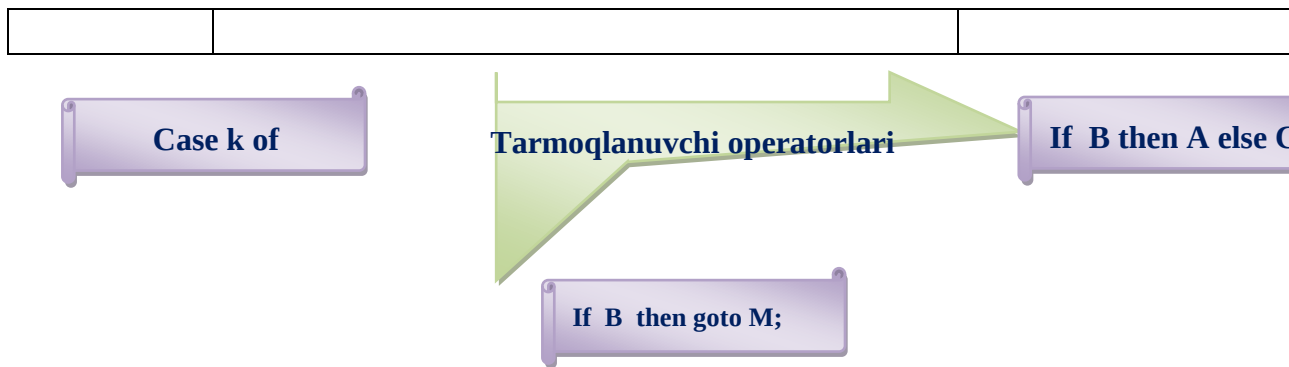
4-AMALIY MASHG'ULOT: Paskalda tanlash operatoriga doir dasturlar

AMALIY MASHG'ULOTNI OLIB BORISH TEXNOLOGIYASI

Vaqt – 2 soat	Talabalar soni: 25-30 nafar
O'quvmashg'ulotishakli	Nazariy bilimlarni mustahkamlash va amaliy ko'nikmalarni shakllantirish amaliy mashg'uloti
O'quv mashg'ulotirejasi	1. Tarmoqlanish algoritmi 2. Tarmoqlanish algoritmini tashkil etish usullari 3. Case tanlash operatori
O'quv mashg'ulotining maqsadi: Amaliy mashg'ulot davomida nazariy bilimlarga asoslangan amaliy ko'nikmalarni shakllantirish.	
O'quv faoliyatining natijalari:	
Pedagogik vazifalar: - amaliy topshiriqlar asosida nazariy bilimlarni mustahkamlash; - amaliy ish orqali tanlash operatorining amaliy ahamiyatini ochib berish; - juftlikda ishlash, o'z xulosalarini asoslash ko'nikmalarini rivojlantirish	Talaba: - Tanlash operatori bo'yicha olingan nazariy bilimlarini amalda qo'llay olish darajasiga etadi.
O'qitish uslubi va texnikasi	suxbat, baxs-munozara, muhokama qilish, fikr almashish dastur tuzish.
O'qitish vositalari	Tarqatma materiallar, Pascal dasturi, slaydlar
O'qitish shakli	Frontal, kichik guruhlarda hamkorlikda ishlash texnologiyasi.
O'qitish sharoitlari	Texnik vositalarni qo'llashga mo'ljallangan auditoriya. Proektor

AMALIY MASHG'ULOTNING TEXNOLOGIK KARTASI

Bosqichlar, vaqti	Faoliyat mazmuni	
	O'qituvchi	Talaba
1-bosqich. Kirish (15 min)	1.1. Mavzuni, uning maqsadi, kutiladigan o'quv natijalarini e'lon qiladi. Mashg'ulot yakka, juftlikda, hamkorlikda ishlash texnologiyasini qo'llagan holda o'tishni ma'lum qiladi. Baholash turlari, mezonlarini e'lon qiladi. 1.2. Mavzu bo'yicha talabalar nazariy bilimlarini umumlashtiradi.	1.3. Mavzuni yozadi va «Nilufar guli» jadvalini to'ldiradi. 1.4. Savollarga javob beradi, eshitadi.
2-bosqich asosiy (55 min)	Pedagog vazifalari: ▪ Case operatorining vazifasini tushuntirish. ▪ 2-amaliy ishni bajarish bosqichlarini tushuntirish va namunaviy vazifani bajarib ko'rsatish. ▪ Mustaqil amaliy ish topshiriqlarini berish ▪ 1-FSMU, 2-B/BX/B jadvalni va 3-ilovalarni berish va ularni to'ldirish so'raladi.	O'quv natijalari: • Case operatorining amaliy ahamiyatini tushunadilar. • amaliy mashg'ulotni o'zlashtiradilar va mustaqil topshiriqlarni bajaradilar.
3-bosqich Yakuniy (10 min)	3.1. Mashg'ulotni yakunlaydi talabalarni baholaydi (4-ilova) va faol ishtirokchilarni rag'batlantiradi. 3.2. Mustaqil ish sifatida misol va mashqlar to'plamidagi misollarni ishlab kelishni topshiradi.	3.1. Eshitadilar. 3.2. Topshiriqni oladilar.



CASE tanlash operatori.

Bu operatoridan dasturda 3 ta va undan ortiq shartlar asosida ishlashga to'g'ri kelganda foydalanish maqsadga muvofiqdir.

Tanlash operatorining umumiy ko'rinishi quyidagicha:

```

case<tanlash uchun parametr> of
<tanlov holati>
else
<operator>

```

holatga ko'ra else <operator> dasturda qatnashmasligi ham mumkin.

2-amaliy ish. Case tanlash operatori.

MAQSAD: Ma'ruzada olingan bilimlarni mustaxkamlash, dasturlash ko'nikmalarini shakllantirish, yechimlarning turli variantlarini aniqlash. Case operatorini qo'llay olishni o'rganish.

TOPSHIRIQLAR: Berilgan variantlarga ko'ra dasturlar tuzish va ularni Turbo-Pascal dasturlash muhitida bajarish hamda himoyalash.

NAZARIY MA'LUMOT: Case operatori shartli o'tish operatori va shartsiz o'tish operatoridan farqli ravishda 3 va undan ortiq shartlar bilan ishlashda juda qulay operator hisoblanadi.

I. O'quvchilar o'qitiuvchi rahbarligida nazariy ma'lumotlar va mavzu yuzasidan yaratilgan dastur bilan tanishtiriladi (10-15 daq).

1-Masala. Kiritilgan K1 va K2 sonlar bilan qo'shish, ayirish, bo'lish, ko'paytirish belgilaridan foydalanish holatiga ko'ra beriladigan natijalarni aniqlovchi dastur tuzing.

label 10;

var belgi:char; k1,k2:real;

begin

readln(belgi); readln(k1,k2) ;

case belgi of {case operatori boshlandi}

'+':writeln('SUMMA ',K1+K2); '-':WRITELN('AYIRMA',K1-K2);

'/':WRITELN('BO'LINMA',K1/K2); '*':WRITELN('KO'PAYTMA',K1*K2);

end; {case operatori tugadi}

II. O'quvchilarga quyidagi masalalarni dasturini tuzib, natijalar olish taklif etiladi.

1. Xisoblang $y = \begin{cases} 0, \text{ agar } -2 < x < 2 & \text{bo'lsa} \\ 4, \text{ aks holda} \end{cases}$, 2. Xisoblang
- $$y = \begin{cases} x^2 + 4x + 5, \text{ agar } x < 2 & \text{bo'lsa} \\ \frac{41}{x^2 + 4x + 5}, \text{ aks holda} \end{cases}$$
3. Xisoblang $y = \begin{cases} 0, \text{ agar } x < 0 & \text{bo'lsa} \\ x, \text{ agar } 0 < x < 1 & \text{bo'lsa} \\ x^4, \text{ aks holda} \end{cases}$ 4. Xisoblang $y = \begin{cases} 0, \text{ agar } x < 0 & \text{bo'lsa} \\ x^2 - x, \text{ agar } 0 < x < 1 & \text{bo'lsa} \\ x^4 - \sin(x^2 - 1), \text{ aks holda} \end{cases}$

III. Amaliy ish himoyasi.

1-Ilova O'quv topshiriq:

FSMU texnologiyasi qoidasi.

Ushbu texnologiya munozarali masalalarni xal etishda hamda o'quv jarayonini baxs-munozarali o'tkazishda qo'llaniladi, chunki bu texnologiya talabalarni o'z fikrini ximoya qilishga, erkin fikrlash va o'z fikrini boshqalarga o'tkazishga, ochiq xolda baxslashishga hamda shu bilan birga baxslashish madaniyatini o'rnatadi. Tinglovchilarga tarqatilgan oddiy qog'ozga o'z fikrlarini aniq va qisqa xolatda ifoda etib, tasdiqlovchi dalillar yoki inkor etuvchi fikrlarni bayon etishga yordam beradi.

F – fikringizni bayon eting; **S** – fikringiz bayoniga sabab ko'rsating; **M** – ko'rsatgan sababingizni

FSMU texnologiyasi yordamida “Nima uchun Case operatorini bilishimiz shart?” savolini yo'naltiruvchi uslubiy tavsiyalar asosida echish:

Savol	Nima uchun Case operatorini bilishimiz shart?
(F) Fikringizni bayon eting	
(S) Fikringiz bayoniga sabab ko'rsating	
(M) Ko'rsatgan sababingizni isbotlovchi dalil keltiring	
(U) Fikringizni umumlashtiring	

B/BX/B JADVALI ilova)

(2 -

Bilaman	Bilishni xohlayman	Bilib oldim

Nazorat ishlari (3-ilova)

3. Masalani EHMda echish bosqichlarini ketma-ketligini o'rnatig.

1. Komp'yuterga kiritish va natija olish
2. Algoritmini tuzish
3. Masalaning qo'yilishi
4. Dastur tuzish
5. Matematik modelini tuzish

4. Har bir bosqichiga izox bering.

№	Bosqich nomi	Izox
1	Masalaning qo'yilishi	
2	Matematik modelini tuzish	

3	Algoritmini tuzish	
4	Dastur tuzish	
5	Komp'yuterga kiritish va natija olish	

Baholash mezonlari va ko'rsatkichlari (ball) 4-ilova.

Nº	F.I.SH.	Nima uchun (1,0)	1-savol (1,0)	2-savol (1,0)	1-misol (1,0)	2-misol (1,0)	Jami ballar (5,0)
1.							
2.							
3.							
4.							
5.							

5-AMALIY MASHG'ULOT:Paskalda tarmoqlanuvchi dasturlar.

AMALIY MASHG'ULOTNI OLIB BORISH TEXNOLOGIYASI

Vaqt –2 soat	Talabalar soni: 25-30 nafar
O'quvmashg'ulotishakli	Nazariy bilimlarni mustahkamlash va amaliy ko'nikmalarni shakllantirishamaliy mashg'uloti
O'quv mashg'ulotirejasi	1. Tarmoqlanuvchi algoritmlar 2. Shartsiz o'tish operatori 3. Shartli o'tish operatori
O'quv mashg'ulotining maqsadi: Ma'ruzada olingan bilimlarni mustahkamlash, dasturlash ko'nikmalarini shakllantirish, yechimlarning turli variantlarini aniqlash.	
O'quv faoliyatining natijalari:	
Pedagogik vazifalar: - amaliy topshiriqlar asosida nazariy bilimlarni mustahkamlash; -3-amaliy ish orqali tanlash operatorining amaliy ahamiyatini ochib berish; -juftlikda ishlash, o'z xulosalarini asoslash ko'nikmalarini rivojlantirish	Talaba: -Tarmoqlanish operatorlari va ularni tashkil etish turlari bo'yicha olingan nazariy bilimlarini amalda qo'llay olish ko'nikmasi shakllanadi.
O'qitish uslubi va texnikasi	suxbat, baxs-munozara, muhokama qilish, fikr almashish dastur tuzish.
O'qitish vositalari	Tarqatma materiallar, Pascal dasturi, slaydlar
O'qitish shakli	Frontal, kichik guruhlarda hamkorlikda ishlash texnologiyasi.
O'qitish sharoitlari	Texnik vositalarni qo'llashga mo'ljallangan auditoriya. Proektor

AMALIY MASHG'ULOTNING TEXNOLOGIK KARTASI

Bosqichlar,v aqti	Faoliyatmazmuni	
	o'qituvchi	talaba
1-bosqich. Kirish (15 min)	1.1.Mavzuni, uning maqsadi, kutiladigan o'quv natijalarini e'lon qiladi. Mashg'ulot yakka, juftlikda, hamkorlikda ishlash texnologiyasini qo'llagan holda o'tishni ma'lum qiladi. Baholash turlari, mezonlarini e'lon qiladi. 1.2.Mavzu bo'yicha talabalar nazariy bilimlarini umumlashtiradi.	1.1. Mavzuni yozadi. 1.2. Savollarga javob beradi, eshitadi.
2-bosqich asosiy (55 min)	2.1. Mavzuga doir masala olinadi. Talabalarni o'ylashga undaydi. Ularning fikri tinglanib qo'yilgan masalaning blok sxemasi tuziladi. So'ng Turbo Pascal muhitida dastur matni kiritilib natijasi olinadi 2.2.3-amaliy ish maqsadi va vazifalari tushuntiriladi. Namunaviy variant ishlab ko'rsatiladi. 2.3.mustaqil vazifa topshiriladi va uni bajarishda o'quv materiallari (ma'ruza matni, o'quv qo'llanma)laridan foydalanish mumkinligini eslatadi.	2.1.Tinglaydi, o'ylaydi, savollarga javob beradi, yozib oladi. 2.2.O'z topshirig'ini bajaradi, zarur holda o'qituvchidan yordam so'raydi. 2.3.Tinglaydi, o'ylaydi, savollarga javob beradi,

	2.4.O'zaro nazorat, muhokama amalga oshirilishini eslatadi. Qatorlar oralab talabalarning misol ishlash va o'zaro muhokamasini kuzatadi. 2-ilova bo'yicha topshiriqlar beriladi	yoziq oladi. 2.4.O'z topshirig'ini bajaradi, zarur holda o'qituvchidan yordam so'raydi. 2-ilovani to'ldirishadilar.
3-bosqich YAkuniy (10 min)	3.1. Mashg'ulotni yakunlaydi talabalarni baholaydi (4-ilova) va faol ishtirokchilarni rag'batlantiradi. 3.2. Mustaqil ish sifatida misol va mashqlar to'plamidagi misollarni ishlab kelishni topshiradi.	3.1. Eshitadilar. 3.2Topshiriqni oladilar.

Shartsiz o'tish operatori bajarilib kelayotgan operatorlar ketma-ketligini o'zgartirib, belgilangan satrga shartsiz o'tishni ta'minlaydi. Belgilangan satr avval aniqlanib, so'ngra shartsiz o'tish operatori GOTO orqali ko'rsatilishi kerak.

Turbo Paskalda satrni belgilash 2 usulda bajariladi:

- 1) 0 dan 9999 gacha bo'lgan butun sonlar orqali.
- 2) harfiy ifoda orqali.

SHartsiz o'tishda foydalanadigan barcha satrlar avval label (shartsiz o'tish satrlarini belgilovchi operator) operatoridan so'ng yozilishi kerak. Dastur jarayonida 2 ta satr 1 ta nomer bilan aniqlanmasligi kerak.Nomer yozilgach ikki nuqta quyiladi.

1-masala.Berilgan butun sonning absalyut qiymatini hisoblovchi dastur tuzing.
program absolyut;

label ST; {o'tiluvchi satrni aniqlash}

var X, Y: integer;

begin

ST:writeln('Biror son kiriting'); readln(x);

Y:=ABS(x); writeln('Bu sonning absolyut qiymati',Y);

Bu dastur uzluksiz davom etadi. Dasturni bajarilishini to'xtatish uchun Ctrl+Break tugmalari bosilishi kerak.

Shartli o'tish operatori bilan berilgan shart bajariligi yoki bajarilmasligiga qarab dastur bajarilishini o'zgartiradi.



3-amaliy ish.

MAQSAD: Ma'ruzada olingan bilimlarni mustaxkamlash, dasturlash ko'nikmalarini shakllantirish, yechimlarning turli variantlarini aniqlash.

TOPSHIRIQLAR:Berilgan variantlarga ko'ra dasturlar tuzish va ularni Turbo-Pascal dasturlash muhitida bajarish hamda himoyalash.

NAZARIY MA'LUMOT: Shartli o'tish operatori buyruqlar bajarilish ketma-ketligini shart asosida bajarilishini ta'minlaydi.

SHARTLI O'TISH OPERATORI ko'rinishi:

IF B THEN A1 ELSE A2; (To'liq ko'rinishi).

IF B THEN A1; (To'liqsiz ko'rinishi).

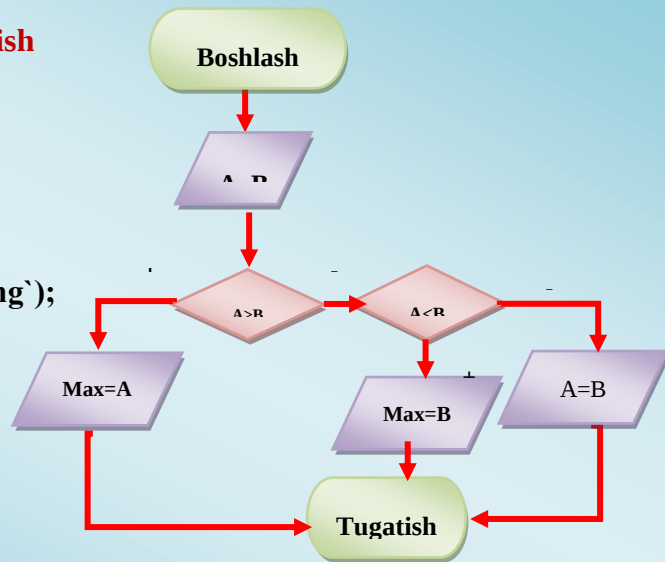
Bu yerda B - mantiqiy ifoda, A1 va A2 - ixtiyoriy ifoda. Bu operatorida B mantiqiy ifoda rost qiymat qabul qilganda A1 ixtiyoriy ifoda bajariladi, aks holda A2 ixtiyoriy ifoda bajariladi.

Shartsiz o'tish operatori ko'rinish: GOTO <belgi>. Bu operator yordamida boshqaruv ushbu belgi qo'yilgan satrga o'tadi. O'z navbatida belgini dasturning LABEL bo'limida e'lon qilinishi kerak.

I. O'quvchilar o'qituvchi rahbarligida nazariy ma'lumotlar va mavzu yuzasidan yaratilgan dastur bilan tanishtiriladi (10-15 daq).

2-masala. Ikki sonning kattasini topish dasturini tuzing

```
program misol_1;
  var A,B: integer; max: real;
begin
  writeln('A ning qiymatini kiriting');
  readln(A);
  writeln('B ning qiymatini kiriting');
  readln(B);
  if A>B then writeln ('max=',A)
  else
```



$$y = \begin{cases} x^2, & \text{agar } x < 0 \\ -5, & \text{agar } x = 0 \\ \sqrt{x}, & \text{agar } x > 0 \end{cases}$$

3-misol.

```
program misol_2;
  Label 1,2,3;
  Var x,y:real;
begin
  readln(x);
  if x>0 then goto 1;
  if x< then goto 2;
  y:=-5; goto 3;
  1: y:=sqr(x); goto 3;
```

II. O'quvchilarga quyidagi masalalarni dasturini tuzib, natijalar olish taklif etiladi.

1. Elektr zanjirdagi ikkita ulangan qarshilikni hisoblash dasturi tuzilsin. Qarshiliklar parallel yoki ketma-ket ulangan bo'lishi mumkin. Quyida dasturni ishlash jarayoni ko'rsatilgan.

Elektr zanjirdagi qarshilikni hisoblash.

Boshlang'ich qiymatlarni kiriting: Birinchi qarshilik (OM) → 15; Ikkichi qarshilik (OM) → 27.3; Ulanish turi (1-ketma-ket, 2-parallel) → 2; Zanjirdagi qarshilik: 9.68 OM

2. Kvadrat tenglamani ildizini topish dasturi tuzilsin. Dastur kiritilgan koeffitsiyetlar orqali diskriminantni tekshirishi va kerakli ma'lumot chiqarishi kerak.

Kvadrat tenglamani ildizini topish.

Bir satrda tenglama koeffitsiyetlarini kiriting va <Enter> tugmasini bosing.

→ 1 -5 6

Tenglama ildizlari:

X1=2, X2=3

3. Chegirma bilan harajatni hisoblash dasturi tuzilsin. Agar harajat 20000 so'mdan oshsa, 10% li chegirma taklif qilinadi. Quyida dasturni ishlash jarayoni ko'rsatilgan.

Chegirma bilan harajatni hisoblash.

Harajatni (so'm) kiriting va <Enter> ni bosing.

→ 21000

Sizga 10% li chegirma beriladi.

Chegirma bilan to'lanadigan summa: 18900 so'm.

5. O'quvchini bilimini tekshirish dasturi tuzilsin. Quyida dasturni ishlash jarayoni ko'rsatilgan. Alisher Navoiy qaysi yili tug'ilgan?

Yilni kiriting va <Enter> ni bosing.

→ 1401

Sizning javobingiz noto'g'ri. Alisher Navoiy 1441 yilda tug'ilgan.

6. O'quvchini bilimini tekshirish dasturi tuzilsin. Quyida dasturni ishlash jarayoni ko'rsatilgan. Qaysi satrda o'zgaruvchilarni e'lon qilish bo'limi to'g'ri ko'rsatilgan.

1. Const

2. Var

3. Type

4. Uses

To'g'ri javob nomerini kiriting va <Enter>ni bosing.

→ 2

Javobingiz to'g'ri.

7. Foydalanuvchi tomonidan kiritilgan ikkita sonni solishtiruvchi dasturni tuzing.

Quyida dasturni ishlash jarayoni ko'rsatilgan.

Bir satrda ikkita sonni kiriting va <Enter> ni bosing: → 22 45 22 45 dan kichik.

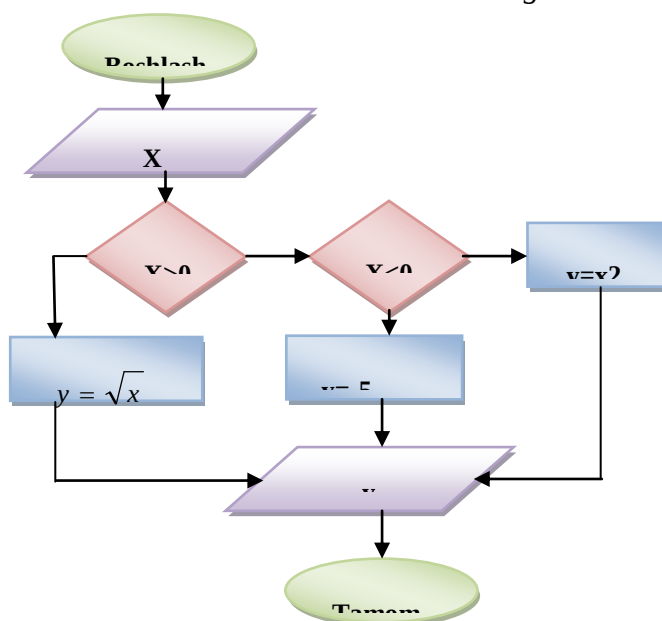
8. Foydalanuvchi tomonidan kiritilgan butun sonni 3 ga bo'linishini aniqlovchi dasturni tuzing. Quyida dasturni ishlash jarayoni ko'rsatilgan.

Butun sonni kiriting va <Enter> ni bosing. → 67 67 soni 3 ga bo'linmaydi.

9. Foydalanuvchi tomonidan kiritilgan butun sonni juft toqligini aniqlash dasturi tuzilsin

Quyida dasturni ishlash jarayoni ko'rsatilgan.

Butun sonni kiriting va <Enter> ni bosing. → 44 44 soni juft son.



10. Foydalanuvchi tomonidan kiritilgan oyning tartib nomeriga mos yil faslini aniqlovchin dasturni tuzing. Agar foydalanuvchi noto'g'ri qimat kiritgan bo'lsa, „Kiritishdagi hatolik“ xabarini bersin.

Quyida dasturni ishlash jarayoni ko'rsatilgan.

Oyning tartib nomerini kiriting (1 dan 12 gacha) → 11 Qish.

NAZORAT TOPSHIRIQLARI

“Insert” texnikasining qoidasi:2-ilova

1. Matnni o'qib chiqing.
2. Olingan ma'lumotlarni diqqat bilan o'rganib, sohalarga ajrating.

BBB texnikasining qoidasi.

- 1.“Insert” texnikasidan foydalanilgan holda matnni o'qib chiqing.
2. Olingan ma'lumotlarni individual soxalarga ajrating. Qalambilan qo'yilgan belgilar

1-o'quv topshiriq BBB jadvali

Nº	Mavzu savollari	Bilaman	Bilishni Istayman	Bilib oldim
1	2	3	4	5
1.	Tarmoqlanuvchi operatorlar turlari			
2.	Tanlash operatori			
3.	Shartsiz o'tish operatori va uni qo'llash			
4.	Shartli o'tish operatorining ko'rinishlari			
5.	if C then A; operatoridan foydalanish			
6.	if B then A else C; operatoridan foydalanish			

Baholash mezonlari va ko'rsatkichlari (ball)

Nº	F.I.SH.	Nima uchun (1,0)	1-savol (1,0)	2-savol (1,0)	1-misol (1,0)	2-misol (1,0)	Jami ballar (5,0)
1							
2							
3							

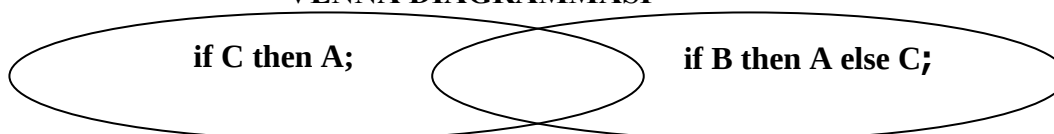
Toifalash sharhini tuzish qoidasi

1. Toifalar bo'yicha ma'lumotlarni taqsimlashning yagona usuli mavjud emas.
2. Bitta mini - guruhda toifalarga ajratish boshqa guruhda ajratilgan toifalardan farq qilishi mumkin.
3. Ta'lim oluvchilarga oldindan tayërlab qo'yilgan toifalarni berish mumkin emas bu ularning mustaqil tanlovi bo'la qolsin.

Toifalash jadvali

Tarmoqlanish operatori		
Case k of	To'liq bo'lmagan ko'rinishi: if C then A;	To'liq ko'rinishi: if B then A else C;

VENNA DIAGRAMMASI



6-AMALIY MASHG'ULOT:Paskalda takrorlanuvchi dasturlar.

AMALIY MASHG'ULOTNI OLIB BORISH TEXNOLOGIYASI

Vaqt –2 soat	Talabalar soni: 25-30 nafar
O'quvmashg'ulotishakli	Nazariy bilimlarni mustahkamlash va amaliy ko'nikmalarni shakllantirishamaliy mashg'uloti
O'quv mashg'ulotirejasi	1. Takrorlanish algoritmi 2. For..To..Do operatori 3. While..Do operatori 4. Repaet..Until operatori
O'quv mashg'ulotining maqsadi: Amaliy mashg'ulot davomida nazariy bilimlarga asoslangan amaliy ko'nikmalarni shakllantirish.	
O'quv faoliyatining natijalari:	
Pedagogik vazifalar: - amaliy topshiriqlar asosida nazariy bilimlarni mustahkamlash; -4-amaliy ish orqali tanlash operatorining amaliy ahamiyatini ochib berish; -juftlikda ishlash, o'z xulosalarini asoslash ko'nikmalarini rivojlantirish	Talaba: -Takrorlanish operatorlari va ularni tashkil etish turlari bo'yicha olingan nazariy bilimlarini amalda qo'llay olish ko'nikmasi shakllanadi.
O'qitish uslubi va texnikasi	suxbat, baxs-munozara, muhokama qilish, fikr almashish dastur tuzish.
O'qitish vositalari	Tarqatma materiallar, Pascal dasturi, slaydlar
O'qitish shakli	Frontal, kichik guruhlarda hamkorlikda ishlash texnologiyasi.
O'qitish sharoitlari	Texnik vositalarni qo'llashga mo'ljallangan auditoriya. Proektor

AMALIY MASHG'ULOTNING TEXNOLOGIK KARTASI

Bosqichlar,v aqt	Faoliyatmazmuni	
	o'qituvchi	talaba
1-bosqich. Kirish (15 min)	1.1.Mavzuni, uning maqsadi, kutiladigan o'quv natijalarini e'lon qiladi. Mashg'ulot yakka, juftlikda, hamkorlikda ishlash texnologiyasini qo'llagan holda o'tishni ma'lum qiladi. Baholash turlari, mezonlarini e'lon qiladi. 1.2.Mavzu bo'yicha talabalar nazariy bilimlarini umumlashtiradi.	1.3. Mavzuni yozadi. 1.4. Savollarga javob beradi, eshitadi.
2-bosqich asosiy (55 min)	2.1. Mavzuga doir masala olinadi. Talabalarni o'ylashga undaydi. Ularning fikri tinglanib qo'yilgan masalaning blok sxemasi tuziladi. So'ng Turbo Pascal muhitida dastur matni kiritilib natijasi olinadi 2.2.4-amaliy ish maqsadi va vazifalari tushuntiriladi. Namunaviy variant ishlab ko'rsatiladi. 2.3.mustaqil vazifa topshiriladi va uni bajarishda o'quv materiallari (ma'ruza matni, o'quv qo'llanma)laridan foydalanish mumkinligini eslatadi. 2.4.O'zaro nazorat, muhokama amalga oshirilishini eslatadi. Qatorlar oralab talabalarining misol ishlash va o'zaro muhokamasini kuzatadi.	2.1.Tinglaydi, o'ylaydi, savollarga javob beradi, yozib oladi. 2.2.O'z topshirig'ini bajaradi, zarur holda o'qituvchidan yordam so'raydi. 2.3.Tinglaydi, o'ylaydi, savollarga javob beradi, yozib oladi. 2.4.O'z topshirig'ini bajaradi, zarur holda o'qituvchidan yordam so'raydi.
3-bosqich YAkuniy (10 min)	3.1. Mashg'ulotni yakunlaydi talabalarni baholaydi (4-ilova) va faol ishtirokchilarni rag'batlantiradi. 3.2. Mustaqil ish sifatida 1-2-3-ilovalarni bajarib kelish topshiradi.	3.1. Eshitadilar. 3.2Topshiriqni oladilar.

6 – amaliy ish.

MAQSAD: Ma'ruzada olingan bilimlarni mustahkamlash, dasturlash ko'nikmalarini shakllantirish, yechimlarning turli variantlarini aniqlash. Qaytarilish jarayonlarini dasturlashda uch xil operatorlarni qo'lashni o'rganish

TOPSHIRIQLAR: Berilgan variantlarga ko'ra dasturlar tuzish va ularni Turbo-Pascal dasturlash muhitida bajarish hamda himoyalash.

NAZARIY MA'LUMOT: Sharti oldindan beriluchi qaytarilish operatori ko'rinish:

WHILE B DO S; Bu yerda B- mantiqi ifoda, S-ixtiyoriy ifoda.

Bu operatora: B mantiqiy ifoda yolg'on qiymat qabul qilgunga qadar S ixtiyoriy ifodani bajarilishi ta'minlanadi.

Sharti keyin beriluvchi qaytarilish operatori korinishi:

REPEAT S UNTIL B; Bu yerda B- mantiqi ifoda, S-ixtiyoriy ifoda.

Bu operaatorda: B-mantiqiy ifoda rost qitmat qabul qilgunga qadar S ixtiyoriy ifodani bajarilishi ta'minlanadi.

Parametrli qaytarilish operatori ko'rinishi:

1. For i:=n1 to n2 do

2. For i:=n2 downto n1 do

Be yerda: i – qaytarilish operatori, n1- parametrning quyi chegarasi, n2-parametrning yuqori chegarasi. $n1 < n2$ uhun bu operator o'rinli hisoblanadi.

Kerakli dasturiy va metodik qo'llanma: Kompyuter, Turbo-Pascal uhti, tarqatma materiallar.

Amaliy ishni bajarish tartibi:

I. O'quvchilar o'qitiuvchi rahbarligida nazariy ma'lumotlar va mavzu yuzasidan yaratilgan dastur bilan tanishtiriladi (10-15 daq).

1-masala. 1 dan 10 gacha bo'lgan toq sonlarning yig'indisini topish dasturi tuzilsin.

Program m_1;

Const n=10;

Var i,S:integer;

begin

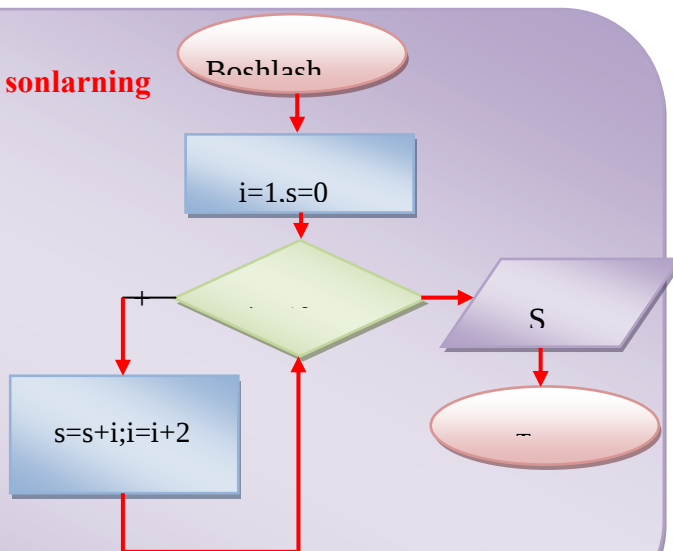
S:=0; i:=1;

while i<=n do

begin

S:=S+i; i:=i+2;

end;



2-misol. Ifoda qiymatini hisoblovchi dastur tuzing.

$$y = \sin 0 + \sin 0,1 + \sin 0,2 + \dots + \sin n$$

```
program 4_misol;
```

```
var x,y,n: real;
```

```
begin
```

```
    x:=0; y:=0;
```

```
    readln(n);
```

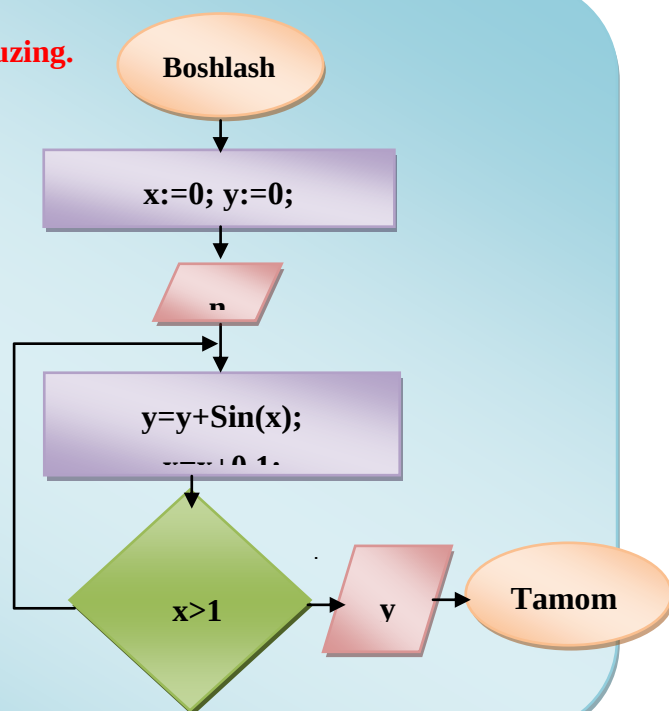
```
Repeat
```

```
    y:=y+Sin(x);
```

```
    x:=x+0.1;
```

```
until x>1;
```

```
writeln ('y=',y);
```



3-misol. $P = \prod_{k=2}^{10} (k + 2)$ yoki $P=3*4*5*6*7*8*9*10*11$

***12 ifoda qiymatini hisoblovchi dastur tuzing.**

```
Program 5_misol;
```

```
var k: byte; p: longint;
```

```
begin
```

```
    p:=1;
```

```
    for k:=2 to 10 do
```

```
    begin
```

```
        p:=p*(k+1);
```

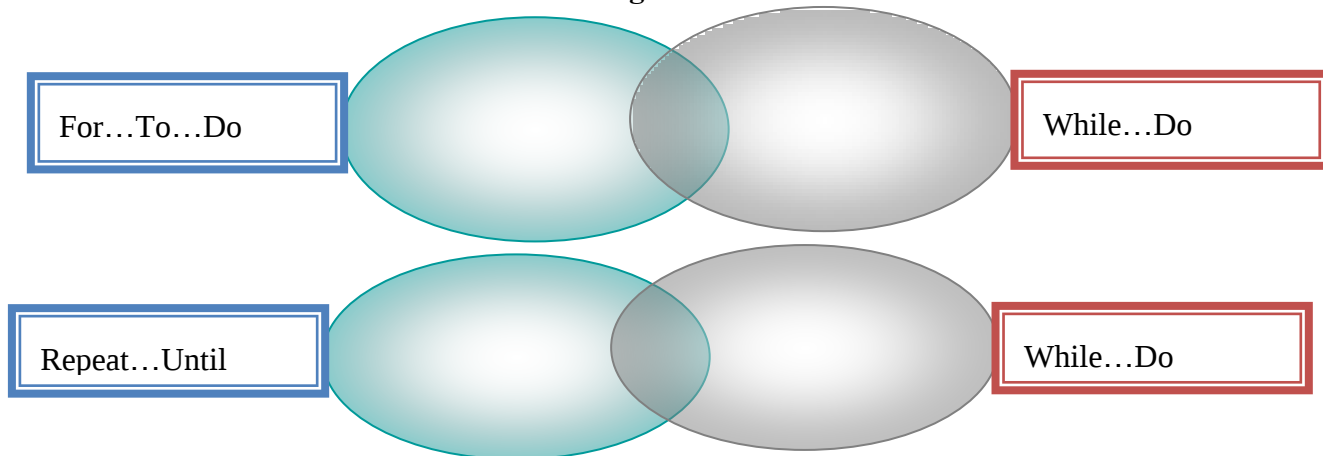
- II. O'quvchilarga quyidagi masalalarni dasturini tuzib, natijalar olish taklif etiladi.
1. N sonigacha bo'lgan juft sonlarni yig'indisini topish dasturi tuzilsin.
 2. M va N sonlarini EKUB ni toping.
 3. Shahar aholisi soni yiliga 1/40 ga oshadi. Necha yilda shahar aholisining soni 3 marta oshadi.
 4. $15x+20y+30z=270$ tenglamaning barcha x, y, z yechimlarini topish dasturini tuzing.
 5. Natural sonlar berilgan. Bu sonlarining yig'idisi toq son bo'ladimi.
 6. A va B ($A < B$) butun sonlar berilgan. Shu sonlar orasidagi barcha sonlarni kamayish tartibida chop etuvchi va ularni sonini aniqlash dasturini tuzing.
 7. A va N ($N > 0$) butun sonlari berilgan. $1 + A + A^1 + A^2 + \dots + A^n$ ifodani hisoblash dasturini tuzing.
 8. N ($N > 0$) butun son berilgan $2 + 1/(2!) + 1/(3!) + \dots + 1/(N!)$ ifodani hisoblash dasturini tuzing.
 9. A va B ($A < B$) butun sonlar berilgan. Shu sonlar orasidagi barcha sonlarni (A va B sonlaridan tashqari) o'sish tartibida chop etuvchi va ularni sonini aniqlash dasturini tuzing.

10. 6 xonali sonlar uchun “Baxtli bilet” dasturini tuzing. Bunda avvalgi 3 xona sonlar yig’indisi keingi 3 xona sonlar yig’indisiga teng bo’lish kerak. Barcha “Baxtli bilet” lar sonini toping.

11. N sonining barcha bo’luvchilarini aniqlash dasturini tuzing.

III. Amaliy ish himoyasi.

Venna diagrammasi 1-ilova.



2-ilova

«Blits-so’rov» savollari savol-javob vaqtida to’ldiriladi.

№	Savollar	Javoblar
1.	Takrorlanish algoritmi qanday tashkil etiladi?	
2.	For .. To .. Do operatorining vazifasi?	
3.	Repeat...Until operatorida shart qachon tekshiriladi?	
4.	While...Do operatorida shart qachon tekshiriladi?	
5.	Case operatori qachon qo’llaniladi?	
6.	Metkalar (niqoblar) qanday qo’llaniladi?	

3-ilova

Dasturi tuzish uchun topshiriqlar:

1-misol. Ifoda qiymatini hisoblovchi dastur tuzing. $y=1,1+1,01+1,001+...+1,000000001$

2-misol. Quyidagi tenglikni hisoblovchi dastur tuzing: $A=1/2+2/3+3/4+...+9/10$

3-misol. Ifodaning qiymatini topish dasturi. $1*1+2*2+...+n*n$

Baholash mezonlari va ko’rsatkichlari (ball) 4 - ilova

№	F.I.SH.	Nima uchun (1,0)	1-savol (1,0)	2-savol (1,0)	1-misol (1,0)	2-misol (1,0)	Jami ballar (5,0)
1							
2							
3							
4							
5							

7-AMALIY MASHG’ULOT: Paskalda murakkab sikllarga doir dasturlar.

AMALIY MASHG’ULOTNI OLIB BORISH TEXNOLOGIYASI

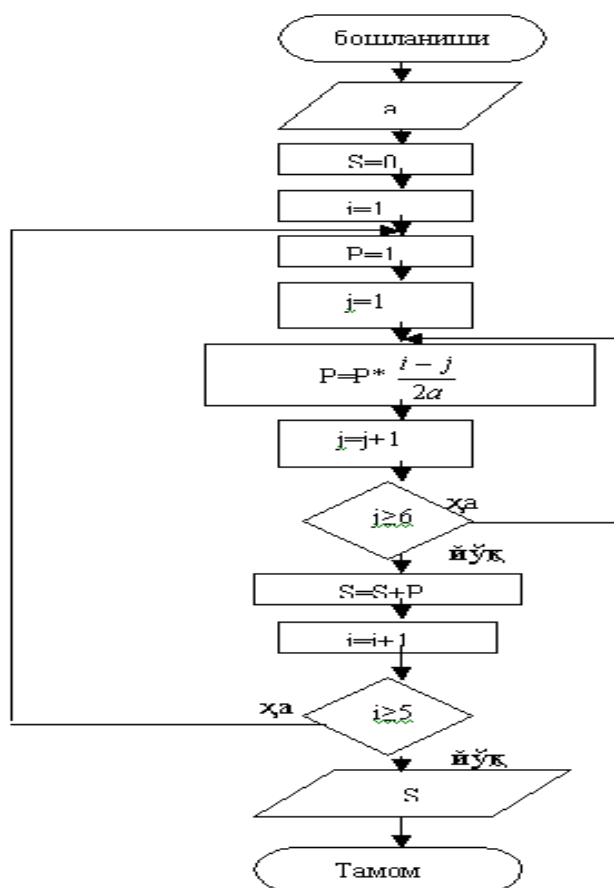
Vaqt –2 soat	Talabalar soni: 25-30 nafar
O'quvmashg'ulotishakli	Nazariy bilimlarni mustahkamlash va amaliy ko'nikmalarni shakllantirishamaliy mashg'uloti
O'quv mashg'ulotirejasi	1. Murakkab takrorlanish jarayonlari(MTJ) 2. MTJ ni For..To..Do operatori yordamida tashkil etish. 3. MTJ ni While..Do operatori yordamida tashkil etish. 4. MTJ ni Repaet..Until operatori yordamida tashkil etish.
O'quv mashg'ulotining maqsadi: Amaliy mashg'ulot davomida nazariy bilimlarga asoslangan amaliy ko'nikmalarni shakllantirish.	
O'quv faoliyatining natijalari:	
Pedagogik vazifalar: - amaliy topshiriqlar asosida nazariy bilimlarni mustahkamlash; -4-amaliy ish orqali tanlash operatorining amaliy ahamiyatini ochib berish; -juftlikda ishlash, o'z xulosalarini asoslash ko'nikmalarini rivojlantirish	Talaba: - Murakkab, yani ichma – ich joylashgan tsikllarni tashkil etish turlari bo'yicha olingan nazariy bilimlarini amalda qo'llay olish ko'nikmasi shakllanadi.
O'qitish uslubi va texnikasi	suxbat, baxs-munozara, muhokama qilish, fikr almashish dastur tuzish.
O'qitish vositalari	Tarqatma materiallar, Pascal dasturi, slaydlar
O'qitish shakli	Frontal, kichik guruhlarda hamkorlikda ishlash texnologiyasi.
O'qitish sharoitlari	Texnik vositalarni qo'llashga mo'ljallangan auditoriya. Proektor

AMALIY MASHG'ULOTNING TEXNOLOGIK KARTASI

Bosqichlar,vaqti	Faoliyatmazmuni	
	o'qituvchi	talaba
1-bosqich. Kirish (15 min)	1.1.Mavzuni, uning maqsadi, kutiladigan o'quv natijalarini e'lon qiladi. Mashg'ulot yakka, juftlikda, hamkorlikda ishlash texnologiyasini qo'llagan holda o'tishni ma'lum qiladi. Baholash turlari, mezonlarini e'lon qiladi. 1.2.Mavzu bo'yicha talabalar nazariy bilimlarini umimlashtiradi.	1.5. Mavzuni yozadi. 1.6. Savollarga javob beradi, eshitadi.
2-bosqich asosiy (55 min)	2.1. Mavzuga doir masala olinadi. Talabalarni o'ylashga undaydi. Ularning fikri tinglanib qo'yilgan masalaning blok sxemasi tuziladi. So'ng Turbo Pascal muhitida dastur matni kiritilib natijasi olinadi 2.2.5-amaliy ish maqsadi va vazifalari tushuntiriladi. Namunaviy variant ishlab ko'rsatiladi. 2.3.mustaqil vazifa topshiriladi va uni bajarishda o'quv materiallari (ma'ruza matni, o'quv qo'llanma)laridan foydalanish mumkinligini eslatadi. 2.4.O'zaro nazorat, muhokama amalga oshirilishini eslatadi. Qatorlar oralab talabalarning misol ishlash va o'zaro muhokamasini kuzatadi.	2.1.Tinglaydi, o'ylaydi, savollarga javob beradi, yozib oladi. 2.2.O'z topshirig'ini bajaradi, zarur holda o'qituvchidan yordam so'raydi. 2.3.Tinglaydi, o'ylaydi, savollarga javob beradi, yozib oladi. 2.4.O'z topshirig'ini bajaradi, zarur holda o'qituvchidan yordam so'raydi.
3-bosqich Yakuniy (10 min)	3.1. Mashg'ulotni yakunlaydi talabalarni baholaydi (4-ilova) va faol ishtirokchilarni rag'batlantiradi. 3.2. 2-3-ilovalaribajaribbaholanadi.	3.1. Eshitadilar. 3.2Topshiriqni bajaradilar.

1-misol. $S = \sum_{i=1}^5 \prod_{j=1}^6 \frac{i+j}{2a}$ yig'indini hisoblash dasturini tuzing, bu erda $a = 6$.

1. Blok–sxemasini tuzamiz:



2. Paskal tilida dasturini tuzamiz: For ..To yordamida

```

TPX.EXE
Program amaliy_5;
uses crt;
var i, j:integer;
    a,S,P:real;
begin
  clrscr;
  Write('a=');readln(a);
  S:=0;
  for i:=1 to 5 do
  begin
    P:=1;
    for j:=1 to 6 do
      P:=P*(i+j)/(2*a);
      S:=S+P;
    end;
    writeln('S=',S);
    readln;
  end.

```

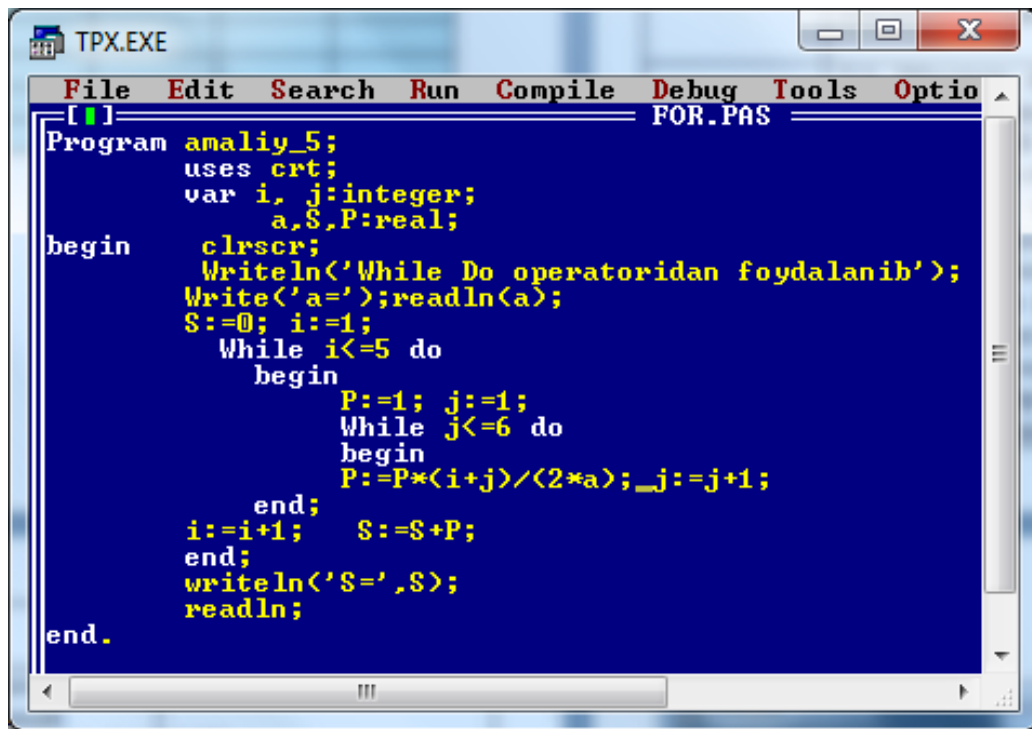
NATIJA:

```

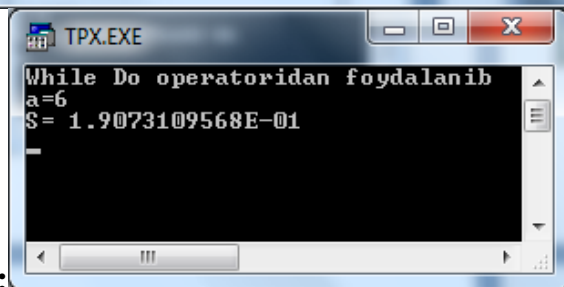
TPX.EXE
a=6
S= 1.9073109568E-01

```

2-USUL:

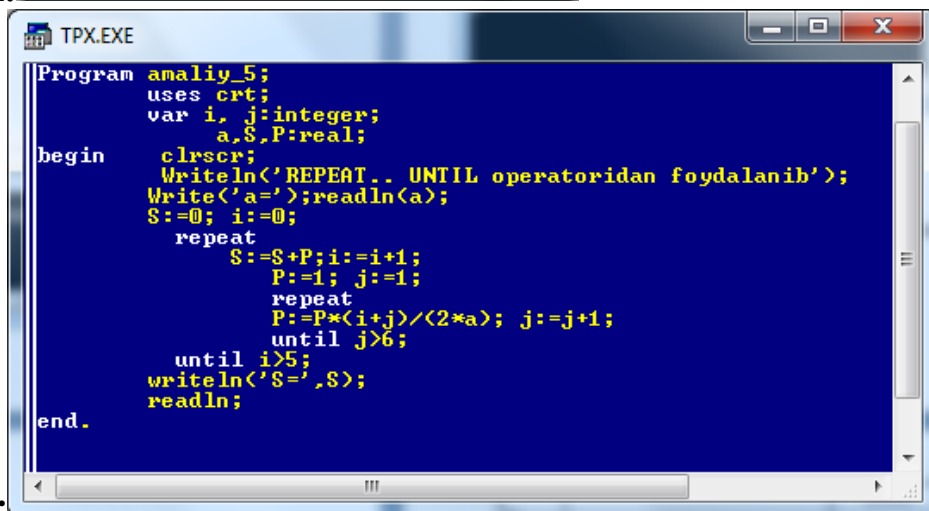


```
TPX.EXE
File Edit Search Run Compile Debug Tools Options
FOR.PAS
Program amaliy_5;
uses crt;
var i, j:integer;
    a,S,P:real;
begin
    clrscr;
    Writeln<'While Do operatoridan foydalanib'>;
    Write<'a='>;readln(a);
    S:=0; i:=1;
    While i<=5 do
    begin
        P:=1; j:=1;
        While j<=6 do
        begin
            P:=P*(i+j)/(2*a); j:=j+1;
        end;
        i:=i+1; S:=S+P;
    end;
    writeln<'S=' ,S>;
    readln;
end.
```



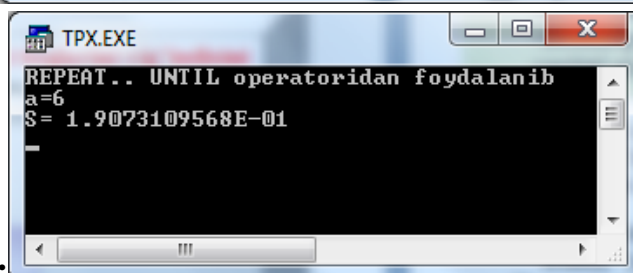
```
TPX.EXE
While Do operatoridan foydalanib
a=6
S= 1.9073109568E-01
```

NATIJA:



```
TPX.EXE
File Edit Search Run Compile Debug Tools Options
FOR.PAS
Program amaliy_5;
uses crt;
var i, j:integer;
    a,S,P:real;
begin
    clrscr;
    Writeln<'REPEAT.. UNTIL operatoridan foydalanib'>;
    Write<'a='>;readln(a);
    S:=0; i:=0;
    repeat
        S:=S+P; i:=i+1;
        P:=1; j:=1;
        repeat
            P:=P*(i+j)/(2*a); j:=j+1;
        until j>6;
    until i>5;
    writeln<'S=' ,S>;
    readln;
end.
```

3-USUL:

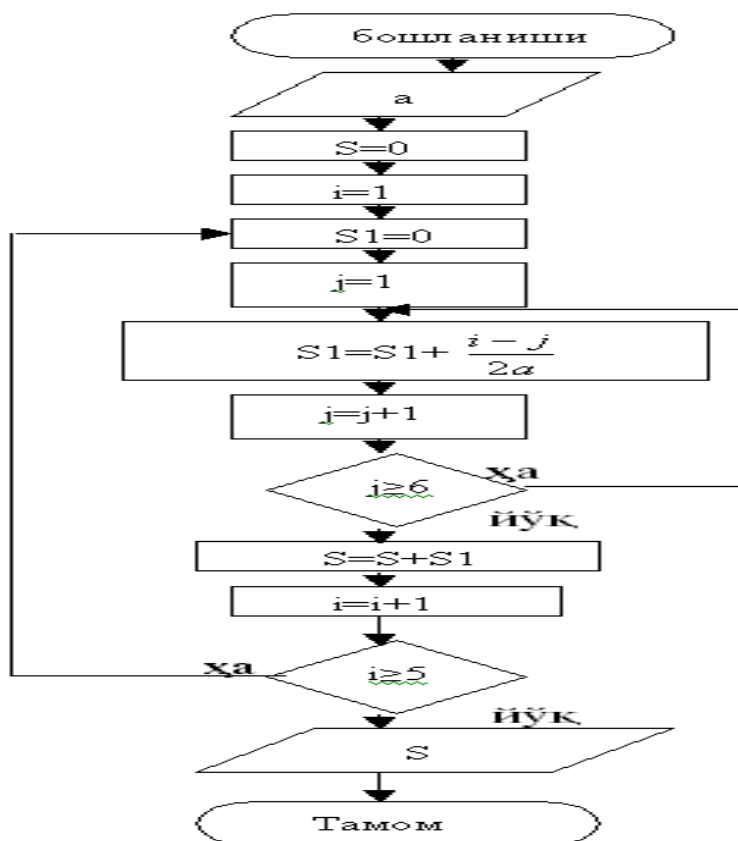


```
TPX.EXE
REPEAT.. UNTIL operatoridan foydalanib
a=6
S= 1.9073109568E-01
```

NATIJA:

2-misol. $S = \sum_{i=1}^5 \sum_{j=1}^6 \frac{i+j}{2a}$ yig'indini hisoblash dasturini tuzing, bu erda $a = 7$.

1. Blok-sxemasini tuzamiz:



2. Paskaltilidasturinituzamiz:

```

TPX.EXE
File Edit Search Run Compile Debug Too
FOR2.PAS
uses crt;
var i, j:integer;
    a,S,S1:real;
begin
  clrscr;
  write('a='); readln(a);
  S:=0;
  for i:=1 to 5 do
  begin
    S1:=0;
    for j:=1 to 6 do
      S1:=S1+(i+j)/(2*a);
    end;
    S:=S+S1;
    write('S=', S);readln;
  end;
end.
  
```

```

TPX.EXE
a=7
S= 1.3928571429E+01
  
```

Natija:

Mustaqil ish topshiriqlar:

1-variant $S = \frac{a+b}{2} \sum_{i=1}^4 \prod_{j=1}^5 \frac{\ln(i^2+1)}{2 \cdot i \cdot j}$, бу ерда $a = 5$; $b = 6$.

2-variant $S = \frac{b-a}{2a} \sum_{k=1}^5 \sum_{\ell=1}^3 \frac{e^{k-\ell}}{\sqrt{k+\ell}}$, бу ерда $a = -3,5$; $b = 3$.

3-variant

$$P = \prod_{n=1}^3 \sum_{m=1}^5 \frac{\sin(n+m^2)}{\sqrt{n+\frac{m}{2}}}$$

4-variant

$$S = a \cdot \sum_{k=1}^3 \prod_{\ell=1}^2 \frac{2,5^{k+\ell}}{\sqrt[3]{k+\ell}}, \quad \delta y \quad ep\partial a \quad a = 5,9.$$

5-variant

$$P = \frac{6,3}{2^x} \prod_{i=1}^4 \prod_{j=1}^5 \frac{2,5^{i+j}}{i+j}, \quad \delta y \quad ep\partial a \quad x = 3.$$

6-variant

$$S = \sum_{k=1}^{10} \prod_{\ell=1}^5 \frac{\ell n(k^2 + \ell)}{4,5^{k+\ell}}$$

7-variant

$$S = a^{\sqrt{x}} \cdot \sum_{i=1}^5 \sum_{j=1}^6 \frac{ctg ij}{5ij}, \quad \delta y \quad ep\partial a \quad a = 3; \quad x = 3.$$

8-variant

$$P = \prod_{i=1}^4 \ell n i + \prod_{j=1}^5 \sin j$$

9-variant

$$S = \ell n a \cdot \sum_{i=1}^5 \frac{e^{i-1}}{5i} + \sum_{j=1}^3 \ell n j, \quad \delta y \quad ep\partial a \quad a = 3,65.$$

10-variant

$$P = e^{\sqrt[3]{x-3}} \prod_{t=1}^6 \prod_{q=1}^7 \frac{t+5}{\ell n q}, \quad \delta y \quad ep\partial a \quad x = 6,59.$$

11-variant

$$P = \prod_{k=1}^5 \prod_{l=1}^8 \frac{\arccos(k+l)}{\sqrt{k^4 + l^4}}$$

12-variant

$$S = \sum_{k=1}^9 \sum_{\ell=1}^3 \frac{2,5^{k+\ell}}{\sqrt[3]{k^\ell}},$$

13-variant

$$S = \sum_{n=1}^5 \sum_{m=1}^6 \frac{\sin(n+m^{\sqrt{x}})}{\sqrt{m+\frac{x}{n}}}, \quad \delta y \quad ep\partial a \quad x = 18,61.$$

14-variant

$$Q = \ell n^2 a \sum_{k=1}^7 \prod_{\ell=1}^5 \frac{e^k - e^{-\ell}}{\sqrt{k+\ell^2}}, \quad \delta y \quad ep\partial a \quad a = 10,89.$$

15-variant

$$W = (b^2 - 4,15) \sum_{k=1}^6 \prod_{\ell=1}^3 \frac{tg^2(k+\ell)}{k^3 + \ell}, \quad \delta y \quad ep\partial a \quad b = 3,09.$$

16-variant

$$Z = \sum_{i=1}^9 \prod_{j=1}^{10} \frac{\sin(i^2 + j)}{2^{i+j}}$$

17-variant

$$P = 2a \sum_{i=1}^5 ctg i + 5b \prod_{j=1}^5 tg j, \quad \delta y \quad ep\partial a \quad a = 1,5; \quad b = 5.$$

18-variant

$$A = \sum_{k=1}^4 \prod_{\ell=1}^5 \frac{\ell n^3(k+\ell)}{5a}, \quad \delta y \quad ep\partial a \quad a = 3,5.$$

19-variant $Q = e^{2x+1} \prod_{i=1}^9 \frac{i+1}{5} + e^{\ell n x} \sum_{j=1}^5 \frac{j-1}{5}$, *by epda* $x = 4,5$.

20-variant $S = \sum_{k=1}^7 \prod_{t=1}^3 \frac{\ell n^3 a^t}{2,61^{k+t}}$, *by epda* $a = 10,719$.

21-variant $Y = \frac{a+b}{5} \prod_{j=1}^6 \frac{j+1}{2} + \sum_{i=1}^3 \ell n i$, *by epda* $a = -5$; $b = 3,5$.

22-variant $P = \prod_{k=1}^3 \prod_{j=1}^4 \frac{\arctg(k + \pi \cdot j)}{k^2 + 5}$

23-variant $S = \sum_{n=1}^6 \prod_{m=1}^4 \frac{\ell n(n^2 + m)}{4^{\sqrt{n+m+0,6}}}$

24-variant $Z = t g \frac{2a}{3} \sum_{n=1}^3 \prod_{m=1}^5 \frac{\sqrt[3]{n+m^2}}{\sqrt{|n-m|}}$, *by epda* $a = 0,3$.

25-variant $W = z^2 \sum_{n=1}^5 \prod_{\ell=1}^4 \frac{t g^2(k + \ell)}{2,5^{n+\ell}}$, *by epda* $z = 5$.

2-ilova

«Blits-so'rov» savollari savol-javob vaqtida to'ldiriladi.

№	Savollar	Javoblar
1.	Takrorlanish algoritmi qanday tashkil etiladi?	
2.	Murakkab tsikllarda For .. To .. Do operatori qanday qo'llanadi?	
3.	Murakkab tsikllarda Repeat...Until operatorida shart qachon tekshiriladi?	
4.	Murakkab tsikllarda While...Do operatori qanday amalga oshiriladi	

Baholash mezonlari va ko'rsatkichlari (ball) 3- ilova

№	F.I.SH.	Nima uchun (1,0)	1-2-savol (1,0)	3-savol (1,0)	4-misol (1,0)	2-misol (1,0)	Jami ballar (5,0)
1							
2							
3							
4							
5							

8-AMALIY MASHG'ULOT:Paskalda funksiya va protseduralar.

Amaliy mashg'ulotni olib borish texnologiyasi

Vaqt –2 soat	Talabalar soni: 25-30 nafar
O'quvmashg'ulotshakli	Nazariy bilimlarni mustahkamlash va amaliy ko'nikmalarni shakllantirishamaliy mashg'uloti
O'quv mashg'ulotirejasi	1. Paskalda proceduralar. 2. Paskalda funksiyalar. 3. Paskalda qism dastur-funksiya
O'quv mashg'ulotining maqsadi: Amaliy mashg'ulot davomida nazariy bilimlarga asoslangan amaliy ko'nikmalarni shakllantirish.	
O'quv faoliyatining natijalari:	
Pedagogik vazifalar:	Talaba:
Formal va joriy o'zgaruvchilar haqida tushuncha beradi.	Formal va joriy o'zgaruvchilar tushunchalariga ega bo'ladilar.
Pascal dasaturlash tilida proseduralar tushunchasini tushuntirib beradi.	Pascal dasaturlash tilida proseduralar yaratishni o'rganadilar.
Funksiyaning berilishi va funksiyaning nomi to'g'risida ma'lumot beradi va misollar ishlaydi.	Funksiyaning berilishi va funksiyaning nomi to'g'risida ma'lumotga ega bo'ladilar.
Qism dastur proseduralarining berilishi va proseduralarning nomi tushuntirad. Misollar ishlanadi.	Qism dastur proseduralarining tadbirini o'rganadilar
O'qitish uslubi va texnikasi	Birgalikda o'rganamiz, o'zaro nazorat, munozara.
O'qitish vositalari	Ma'ruza matni, mustaqil ishlar to'plami, o'quv qo'llanmasi, proektor, doska, tarqatma materiallar, slaydlar.
O'qitish shakli	Guruhda, individual, juftlikda o'qitish.
O'qitish sharoitlari	Komp'yuter, proektor, doska bilan ta'minlangan, guruhda dars o'tishga moslashtirilgan auditoriya

AMALIY MASHG'ULOTNING TEXNOLOGIK KARTASI

Bosqichlar,vaqti	Faoliyatmazmuni	
	O'qituvchi	Talaba
1-bosqich. Kirish (15 min)	1.1.Mavzuni, uning maqsadi, kutiladigan o'quv natijalarini e'lon qiladi. Mashg'ulot yakka, juftlikda, hamkorlikda ishlash texnologiyasini qo'llagan holda o'tishni ma'lum qiladi. Baholash turlari, mezonlarini e'lon qiladi. 1.2.Mavzu bo'yicha talabalar nazariy bilimlarini umumlashtiradi.	3.1. Mavzuni yozadi va «Nilufar guli» jadvalini to'ldiradi. 3.2. Savollarga javob beradi, eshitadi.
2-bosqich asosiy (55 min)	2.1. Mavzuga doir masala olinadi. Talabalarni o'ylashga undaydi. Ularning fikri tinglanib qo'yilgan masalaning blok sxemasi tuziladi. So'ng Turbo Pascal muhitida dastur matni kiritilib natijasi olinadi 2.2. Talabalarga har biriga yakka tartibda bajarish uchun mustaqil ish topshiriladi. Har bir talaba daftariga mustaqil ishlar to'plamidan o'z variantidagi misollarni yozib oladi. 2.3.Vazifani bajarishda o'quv materiallari (ma'ruza matni, o'quv qo'llanma)laridan foydalanish mumkinligini eslatadi. 2.4.O'zaro nazorat, muhokama amalga oshirilishini eslatadi. Qatorlar oralab talabalarining misol ishlash va o'zaro muhokamasini kuzatadi.	2.1.Tinglaydi, o'ylaydi, savollarga javob beradi, yozib oladi. 2.2.O'z topshirig'ini bajaradi, zarur holda o'qituvchidan yordam so'raydi. 2.3..Tinglaydi, o'ylaydi, savollarga javob beradi, yozib oladi. 2.5.O'z topshirig'ini bajaradi, zarur holda o'qituvchidan yordam so'raydi.
3-bosqich Yakuniy (10 min)	3.1. Mashg'ulotni yakunlaydi talabalarni baholaydi (4-ilova) va faol ishtirokchilarni rag'batlantiradi. 3.2. Mustaqil ish sifatida misol va mashqlar to'plamidagi misollarni ishlab kelishni topshiradi.	3.1. Eshitadilar. 3.2Topshiriqni oladilar.

8.1-AMALIY ISH.

MAQSAD: Ma'ruzada olingan bilimlarni mustaxkamlash, dasturlash ko'nikmalarini shakllantirish, yechimlarning turli variantlarini aniqlash. Qism dasturlar bilan ishlash ko'nikmasini shakllantirish

TOPSHIRIQLAR: Berilgan variantlarga ko'ra dasturlar tuzish va ularni Turbo-Pascal dasturlash muhitida bajarish hamda himoyalash.

NAZARIY QISM: Dasturlash jarayonida murakkab dasturlarning bir necha joyida bir xil vazifani bajaruvchi operatorlar guruhini qo'llashga to'g'ri keladi va dasturda bir-biriga aynan o'xshash bir necha qism dasturlar vujudga keladi. Kompyuter o'tirasini va dasturtuzuvchi vaqtini tejash maqsadida ushbu qismdagi o'xshash peratorlardan bir marta asosiy dastur dasturdan ajratib yoziladi va unga asosiy dastur bajarilishi jarayonida murojaat qilinadi.

Dasturning ixtiyoriy qismidan murojaat qilib, bir necha bor ishlatilishi mumkin bo'lan buday operatorlar guruhida qism dastur deb ataladi va u asosiy dastur bilan bir butunni tashkil qiladi.

Qism dasturlarni ishlatish dasturning hajmini kichraytiradi va uning ko'rinishini , o'qilishini hamda xatolar sonini kamaytiradi.

Protsedura quyidagicha e'lon qilinadi:

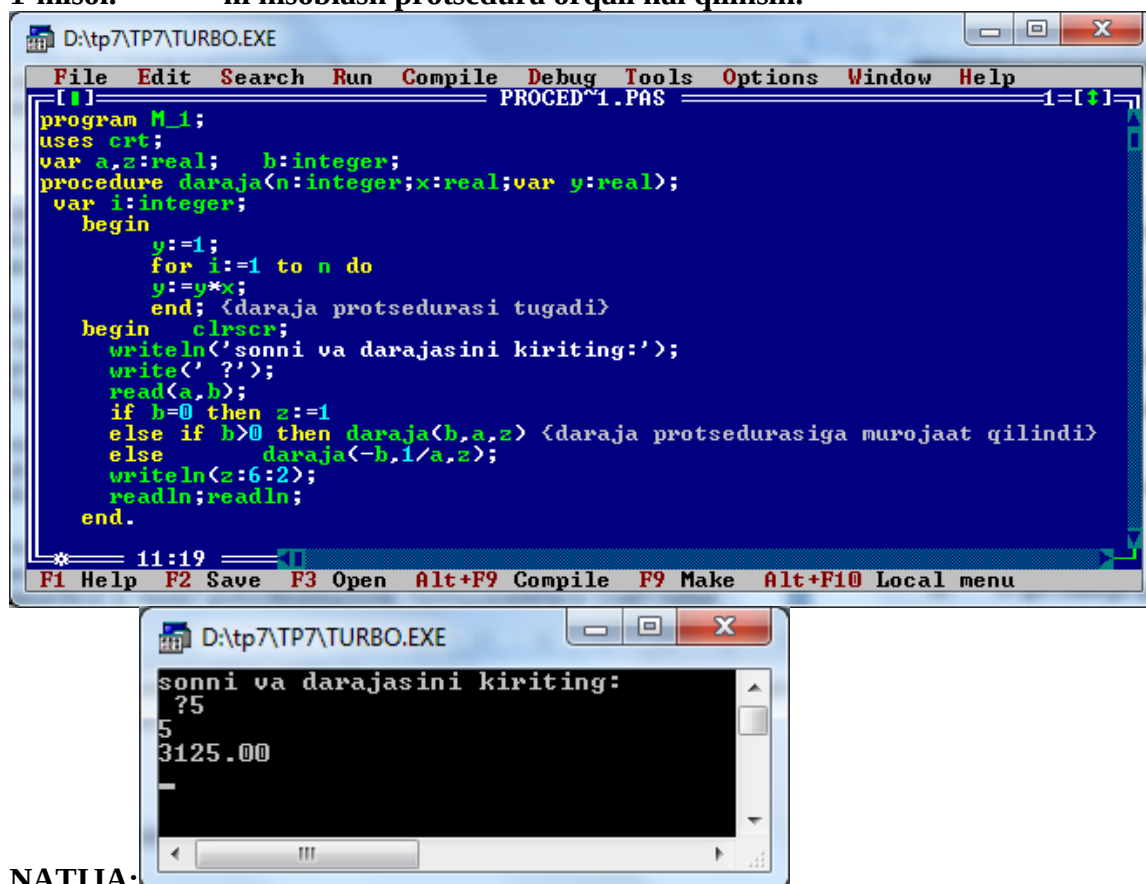
Procedure<nom>(<sohta parametrlar ro'yhati>);

Bu yerda PROCEDURE-xizmatchi so'z; <nom>-protseduraning nomi; <soxta parametrlar ro'yxati>-turlari ko'rsatilgan erilganlarni va natijalarni ifodalaydigan nomlar turlari sanab o'tiladigan satr.

Amaliy ishni o'tkazish tartibi.

I. O'qituvchi quyidagi masalani o'quchilarga tushintirib beradi.

1-misol. $y = x^n$ ni hisoblash protsedura orqali hal qilinsin.



NATIJA:

2-MISOL. 4 ta sonning eng kattasini topish protsedura orqali hal qilinsin.

The top screenshot shows the Turbo Pascal 7.0 IDE with the following code in 'FUNCSIYA.PAS':

```

program TUTTA_SONNING_KATTASINI_TOPISH;
var a,b,c,d:integer;
    mah,mcd,max:integer;
procedure max1(x,y:integer;var z:integer);
begin
    if x>y then z:=x else z:=y;
end; {max1 protsedurasi tugadi}
begin
    writeln('4 ta butun sonni kiriting: ');
    readln(a,b,c,d);
    max1(a,b,mah); {protseduraga murojaat}
    max1(c,d,mcd);
    max1(mah,mcd,max);
    writeln('Eng kattasi:',max);
    readln;
end.

```

The bottom screenshot shows the program's execution output:

```

Turbo Pascal Version 7.0 Copyright (c) 198
4 ta butun sonni kiriting:
12
-45
34
-67
Eng kattasi:34

```

NATIJA:

II. O'quvchilarga quyidagi masalalarni dasturini tuzib, natijalar olish taklif etiladi.

1. * dan tashkil topgan satrni chop etuvchi satrni dasturini tuzing. * lar soni protsedura parametri bo'lsin.
2. Parallelopipedni hajmini, asos yuzasini hisoblash protsedurasini tuzing. Protsedura parametrlari: bo'yi, eni, balandligi.
3. Agar M1, M2, M3 X(30), Y(15), Z(20) massivlarining maksimum elementlari bo'lsa, $K=(M1+M2+M3)/3$ ni hisoblash dasturini tuzing
4. To'rtburchakning burchaklari koordinatalari berilgan. To'rtburchak perimetri va yuzasini topish dasturini tuzing.
5. A(15), V(10), S(20) massivlariberilgan. Massiv elementlarini o'sish tartibida chop etuvchi dastur tuzing.
6. Agarm1 vam2 sonlarX(20) vaX(35) massivlarning eng kichik elementlari bo'lsa, $Z=(m1+m2)/2$ nihisoblashdasturinituzing.
7. X(5,5) va Y(7,7) massivlarining bosh dioganallarida joylashdan elementlar yig'indisini topish dasturini tuzing.
8. A(m,n) massivdagi eng ko'p ketme-ket keluvchi bir xil qiymatli sonlar ketma-ketligini aniqlovchi dastur tuzing.
9. Faqat toq satrlaridan iborat bo'lgan massivn elementlarining:
 1. yig'indisini topish.
 2. O'rtacha arifmetigini topish.
 3. Eng kattasini topish.
 4. Eng kichigini topish dasturini tuzing.
10. Uchburcakni berilgan 3 ta tomoniga ko'ra perimetrini va yuzasini topish dasturi tuzilsin. Dastur foydalanuvchi tomonidan kiritilgan qiymatlarni tekshirishi kerak va 'Kiritish xatosi' a'lumotini chop etsin.

III. Amaliy ish himoyasi.

8.2-AMALIY ISH.

PASCALDA QISM DASTUR-FUNKSIYA.

MAQSAD: Ma'ruzada olingan bilimlarni mustaxkamlash, dasturlash ko'nikmalarini shakllantirish, yechimlarning turli variantlarini aniqlash. Qism dasturlar bilan ishlash ko'nikmasini shakllantirish.

TOPSHIRIQ: Talabalarning ma'ruza va amaliy mashg'ulotlarda olgan bilimlarini mustaxkamlash, dasturlash ko'nikmalarini shakllantirish, yechimlarning turli variantlarini aniqlash.

NAZARIY MA'LUMOT: funksiya bu shunday qism dasturki, bunda bitta natija chiqadi va u funksiya qism dasturning nomida hosil bo'ladi. Shuning uchun funksiya qism dasturi protsedura qism dasturining xususiy holi bo'lib, undan farqi, birinchidan, funksiya qism dasturining natijasi faqat bitta qiymatdan iborat bo'ladi. Protsedura qism dasturida natija esa bitta yoki bir nechta bo'lishi mumkin. Funksiya qism dasturining natijasi asosiy dasturga funksiya natijasi sifatida beriladi. Protsedura qism dasturi natijalari esa uning parametrlari qiymati sifatida hosil bo'ladi.

Funksiya qism dasturining ko'rinishi quyidagicha:

Function<nom>(<soxta parametrlar ro'yxati>):<tur> ;

Bu yerda function-xizmatchi so'z, <nom>-funksiya nomi,<soxta parametrlar ro'yxati>-kiritiladigan soxta parametrlar ro'yxati sana o'tiladi, <tur>-funksiya natijasi turi. Bunda natija funksiya nomida hosil bo'ladi. Bundan tashqari, funksiyaning parametrsiz ko'rinishini ham yozish mumkin;

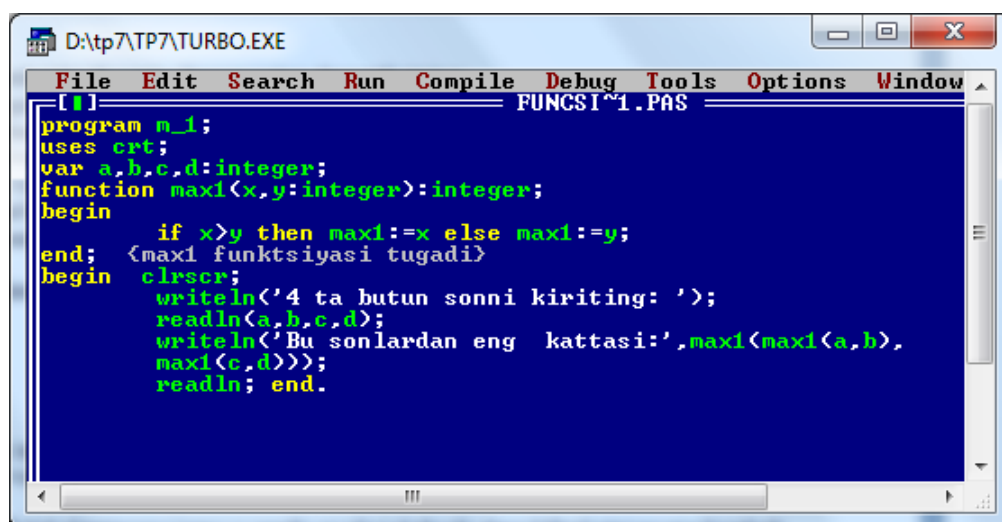
Function <nom>:<tur>;

Har ikkala holda ham funksiya natijasi funksiya qism dasturi tanasida funksiya nomiga hech bo'laganda bir marta o'zlashtirilishi shart;

Amaliy ishni o'tkazish tartibi.

I. O'qituvchi quyidagi masalani o'quchilarga tushintirib beradi.

3-MISOL.4 ta sonning eng kattasini topish funksiya orqali hal qilinsin.



```
program m_1;
uses crt;
var a,b,c,d:integer;
function max1(x,y:integer):integer;
begin
    if x>y then max1:=x else max1:=y;
end; {max1 funktsiyasi tugadi}
begin clrscr;
    writeln('4 ta butun sonni kiriting: ');
    readln(a,b,c,d);
    writeln('Bu sonlardan eng kattasi:',max1(max1(a,b),
    max1(c,d)));
    readln; end.
```

NATIJA:

```

D:\tp7\TP7\TURBO.EXE
4 ta butun sonni kiriting:
23
-5
64
-52
Bu sonlardan eng kattasi:64

```

4-MISOL. Ikkita registrdan tashkil topgan zanjirdagi qarshilikni hisoblash dasturi tuzilsin. Funksiya parametrlari bo'lib ulanish turi (ketma-ket yoki parallel) turi qabul qilinsin.

```

D:\tp7\TP7\TURBO.EXE
File Edit Search Run Compile Debug Tools Options Window Help
[ ] FUNCSI~2.PAS 1=[ ]
program zanjir;
uses crt;
var
  q1,q2:real;
  tur:integer;
function qq(r1,r2:real;t:integer):real;
begin
  if t=1 then qq:=r1+r2;
  if t=2 then qq:=r1*r2/(r1+r2);
end;
begin
  clrscr;
  write('Qarshiliklarni kiriting:');
  readln(q1,q2);
  write('Ketma-ket ulanish uchun 1/Parallel ulanish uchun 2:');
  readln(tur);
  writeln('Ushbu zanjidagi qarshilik:',qq(q1,q2,tur):6:2,' OM');
  readln;
end.
17:60
F1 Help F2 Save F3 Open Alt+F9 Compile F9 Make Alt+F10 Local menu

D:\tp7\TP7\TURBO.EXE
Qarshiliklarni kiriting:123
234
Ketma-ket ulanish uchun 1/Parallel ulanish uchun 2:2
Ushbu zanjidagi qarshilik: 80.62 OM

```

- II. O'quvchilarga quyidagi masalalarni dasturini tuzib, natijalar olish taklif etiladi.
1. Silindrni hajmini hisoblash funksiyasi dasturini tuzing. Funksiya parametrlari bu_ silindr asosi radiusi va balandligi.
 2. Ikkita sonni solishtirib, mos xolda $>$, $<$, yoki $=$ korinishlarida natijani ko'rsatuvchi funksiya dasturini tuzing.
 3. Ikkita qarshilikdan iborat zanjirdagi umumiy qarshilikni topish funksiyasi dasturini tuzing. Funksiya parametrlari bo'lib ulanish turi ko'rsatilsin.
 4. a^b fuksiyani hisoblash funksiyasi dasturini tuzing.
 5. Berlgan sonni N foizini hisoblash funksiyasi dasturini tuzing.
 6. Bankka qo'yilgan puldan ma'lum vaqtdan keyin olinadigan foydani hisoblash funksiyasi dasturini tuzing. Fuksiya paramtrlari: Boshang'ich summa, yillik (oylik) foiz, muddati.
 7. $n!$ ni hisolash fuksiyasi dasturini tuzing.
 8. n natural son va $n > 2$. Quyiagi yig'indini hisoblash dasturi tuzilsin

$$S=1+(1+2)+(1+2+3)+\dots+(1+2+3+\dots+n)$$

9. Chiziqli tenglamani ildizini topish dasturini tuzing. Funksiya parametrlari: a, b koeffitsiyentlar.

10. A(n) masivni shunday hosil qilinki, $a(1)=1!$, $a(2)=2!$... $a(n)=n!$

III. Amaliy ish himoyasi.

Baholash mezonlari va ko'rsatkichlari (ball)4-ilova.

№	F.I.SH.	Nima uchun (1,0)	1-misol (1,0)	2-misol (1,0)	3-misol (1,0)	4-misol (1,0)	Jami ballar (5,0)
1.							
2.							
3.							
4.							
5.							

9-AMALIY MASHG'ULOT:Paskalda massivlar va satriy kattaliklar bilan ishlash.

AMALIY MASHG'ULOTNI OLIB BORISH TEXNOLOGIYASI

Vahti –2 soat	Talabalar soni: 25-30 nafar
O'quvmashg'ulotishakli	Nazariy bilimlarni mustahkamlash va amaliy ko'nikmalarni shakllantirishamaliy mashg'uloti
O'quv mashg'ulotirejasi	1. Bir o'lchovli massivlar bilan ishlash 2. Ikki o'lchovli massivlar bilan ishlash 3. Satriy kattaliklar bilan ishlash
O'quv mashg'ulotining maqsadi: Amaliy mashg'ulot davomida nazariy bilimlarga asoslangan amaliy ko'nikmalarni shakllantirish.	
O'quv faoliyatining natijalari:	
Pedagogik vazifalar: - amaliy topshiriqlar asosida nazariy bilimlarni mustahkamlash; -individual topshiriqlar asosida mustaqil amaliy ko'nikmalarga ega bo'lishga o'rgatish; -juftlikda ishlash, o'z xulosalarini asoslash ko'nikmalarini rivojlantirish	Talaba: - Turbo Pascal dasturlash muhitida massivlar va satriy kattaliklarga doir dasturlarni kiritib, natija olish ko'nikmasiga ega bo'ladilar.
O'qitish uslubi va texnikasi	Birgalikda o'rganamiz, o'zaro nazorat, munozara.
O'qitish vositalari	Ma'ruza matni, mustaqil ishlar to'plami, o'quv qo'llanmasi, proektor, doska, tarqatma materiallar, slaydlar.
O'qitish shakli	Guruhda, individual, juftlikda o'qitish.
O'qitish sharoitlari	Komp'yuter, proektor, doska bilan ta'minlangan, guruhda dars o'tishga moslashtirilgan auditoriya

Mavzu: № 7

Paskalda massivlar va satriy kattaliklar bilan ishlash

AMALIY MASHG'ULOTNING TEXNOLOGIK KARTASI

Bosqichlar,vaqti	Faoliyatmazmuni	
	o'qituvchi	talaba
1-bosqich. Kirish (15 min)	1.1.Mavzuni, uning maqsadi, kutiladigan o'quv natijalarini e'lon qiladi. Mashg'ulot yakka, juftlikda, hamkorlikda ishlash texnologiyasini qo'llagan holda o'tishni ma'lum qiladi. Baholash turlari, mezonlarini e'lon qiladi. 1.2.Mavzu bo'yicha talabalar nazariy bilimlarini umimlashtiradi.	3.3. Mavzuni yozadi va «Nilufar guli» jadvalini to'ldiradi. 3.4. Savollarga javob beradi, eshitadi.
2-bosqich	2.1. Mavzuga doir masala olinadi. Talabalarni o'ylashga undaydi.	2.1.Tinglaydi, o'ylaydi,

asosiy (55 min)	Ularining fikri tinglanib qo'yilgan masalaning blok sxemasi tuziladi. So'ng Turbo Pascal muhitida dastur matni kiritilib natijasi olinadi 2.2. Talabalarga har biriga yakka tartibda bajarish uchun mustaqil ish topshiriladi. Har bir talaba daftariga mustaqil ishlar to'plamidan o'z variantidagi misollarni yozib oladi. 2.3. Vazifani bajarishda o'quv materiallari (ma'ruza matni, o'quv qo'llanma)laridan foydalanish mumkinligini eslatadi. 2.4. O'zaro nazorat, muhokama amalga oshirilishini eslatadi. Qatorlar oralab talabalarning misol ishlash va o'zaro muhokamasini kuzatadi.	savollarga javob beradi, yozib oladi. 2.2. O'z topshirig'ini bajaradi, zarur holda o'qituvchidan yordam so'raydi. 2.3. Tinglaydi, o'ylaydi, savollarga javob beradi, yozib oladi. 2.5. O'z topshirig'ini bajaradi, zarur holda o'qituvchidan yordam so'raydi.
3-bosqich YAkuniy (10 min)	3.1. Mashg'ulotni yakunlaydi talabalarni baholaydi (4-ilova) va faol ishtirokchilarni rag'batlantiradi. 3.2. Mustaqil ish sifatida misol va mashqlar to'plamidagi misollarni ishlab kelishni topshiradi.	3.1. Eshitadilar. 3.2. Topshiriqni oladilar.

9.1- AMALIY ISH.

BIR O'LCHOVLI MASSIV BILAN ISHLASH.

MAQSAD: Ma'ruzada olingan bilimlarni mustaxkamlash, Massiv bilan ishlash ko'nikmalarini shakllantirish.

TOPSHIRIQLAR: Berilgan variantlarga ko'ra dasturlar tuzish va ularni Turbo-Pascal dasturlash muhitida bajarish hamda himoyalash.

NAZARIY MA'LUMOT: Massiv o'zida bir xil tipdagi qiymatlarni saqlovchi kattalik xisoblanadi. Massiv bilan ishlash uchun quyidagilarni amalga oshirish kerak:

Massiv o'zgaruvchilar yoki ozgarmaslar bo'limida e'lon qilinishi mumkin:

Massivni kiritish, chiqarish, elementlari bilan ishlash uchun qaytarilish operatorlaridan foydalanish kerak bo'ladi. (while, repeat, for).

Agar dasturda massiv ishtirok etsa uni quyidagicha e'lon qilinadi:

var

massiv_nomi:array[boshlang'ich indeks..oxirgi indeks] of [qiymatlar turi];

Bu yerda massiv_nomi uchun pascal tili qonuniyati bo'yicha ixtiyoriy identifikator tanlanadi, Array-massivni e'lon qiluvchi so'z, Boshlang'ich va oxirgi indekslar – butun sonlar bo'lishi shart. Misol uchun yanvar oyidagi temperatura qiymatlarini o'zida saqlovchi massivni e'lon qilish uchun:

var

Yanvar:array[1..31] of real;

Bu holda biz bo'sh massiv e'lon qildik. Masalani hal qilish davomida bu massini to'ldirish, qiymatlari ustida amallar bajarish kerak bo'ladi, buning uchun massivning har bir elementiga murojaat qilish kerak bo'ladi. Quyida massivni to'ldirish ko'rsatilgan:

For i:=1 to 31 do

Begin

Write(I, '-elementni kiriting:');

Readln(a[i]);

End;

Amaliy ishni bajarish tartibi:

I. O'quvchilar o'qituvchi rahbarligida nazariy ma'lumotlar va mavzu yuzasidan yaratilgan dastur bilan tanishtiriladi (10-15 daq).

1-masala. Foydalanuvchi tomonidan kiritilgan son massivda necha marta qatnashishini aniqlash dasturi tuzilsin. (Massivda biror sonni qidirish).

```

Program m_1;
Uses crt; Const      N=7;
var  a:array[1..n] of integer;
      s,q,i:integer;
begin  clrscr;
       writeln(n,' ta butun sonni kiriting: ');
       for i:=1 to n do
       begin
           write(i,'-son:');
           readln(a[i]);
       end;
       write('Qidirilayotgn sonni kiriting:');
       readln(q);
       s:=0;
       for i:=1 to n do
       if a[i]=q then s:=s+1;
       if s<>0 then writeln(q,' soni ', s,' marta qatnashdi' )
       else writeln('Bunday element massivda mavjud emas');
       readln;
       end.

```

```

7 ta butun sonni kiriting:
1-son:1
2-son:2
3-son:1
4-son:3
5-son:5
6-son:4
7-son:4
Qidirilayotgn sonni kiriting:4
4 soni 2 marta qatnashdi

```

NATIJA:

II.O'quvchilarga quyidagi masalalarni dasturini tuzib, natijalar olish taklif etiladi.

1. Massivning o'rtacha arifmetigini aniqlash dasturi tuzilsin.
 2. Massiv elementlarini o'sish tartibida saralash dasturini tuzing.
 3. Massiv elementlari ichidan eng kattasini topish dasturi tusilsin.
 4. Massiv elementlari o'sish tartibida joylashganini aniqlash dasturi tuzilsin.
 5. N ta elementli massiv berilgan. Eng katta va eng kichik elementlar o'rnini almashtirish dasturini tuzing.
 6. N ta elementdan iborat massiv elementlarini teskari tartibda chop etuvchiu dasturni tuzing.
 7. N ta elementdan iborat massiv elementlarinining faqat toq sonlarini chop euvchi dastur tuzing.
 8. $a[1..10]$ massivni tasodifiy xolda 0 va 1 sonlari bilan to'ldiring va 0 ni 1 soniga, 1 ni 0 soniga almashtiruvchi dastur tuzing.
 9. Shunday massiv tuzingki, unda dastlabki 100 ta 13 va 17 ga bo'linadigan sonlar joylashsin.
 10. $xy+x+y=1000$ tenlamaning barcha ildizlarini topish dasturini tuzing.
- III. Amaliy ish himoyasi.

9.2 -AMALIY ISH.

IKKI O'LCHOVLI MASSIV BILAN ISHLASH.

MAQSAD: Ma'ruzada olingan bilimlarni mustaxkamlash, dasturlash ko'nikmalarini shakllantirish, yechimlarning turli variantlarini aniqlash. Massiv elementlari ustida amallar bajarish.

TOPSHIRIQLAR: Berilgan variantlarga ko'ra dasturlar tuzish va ularni Turbo-Pascal dasturlash muhitida bajarish hamda himoyalash.

NAZARIY MA'LUMOT: Ikki o'lchovli massiv bilan ishlash uchun ikkita o'lcham bilan ishlashga to'g'ri keladi, satr nomeri va ustun nomeri. Misol uchun beshinchi satr, yettinchi ustunda 8 soni oylasgan bo'lsa, «pascal» tilida quyidagicha ko'rinishga ega bo'ladi:

A[5,7]:=8

«pascal» tilida ikki o'lchovli massiv quyidagicha e'lon qilinadi:

Var

massiv_nomi:array

[Birinchi_satr_nomeri..

Oxirgi_satr_nomeri,

Birinchi_ustun_nomeri..

Oxirgi_ustun_nomeri] of massiv_elementlari_tipi;

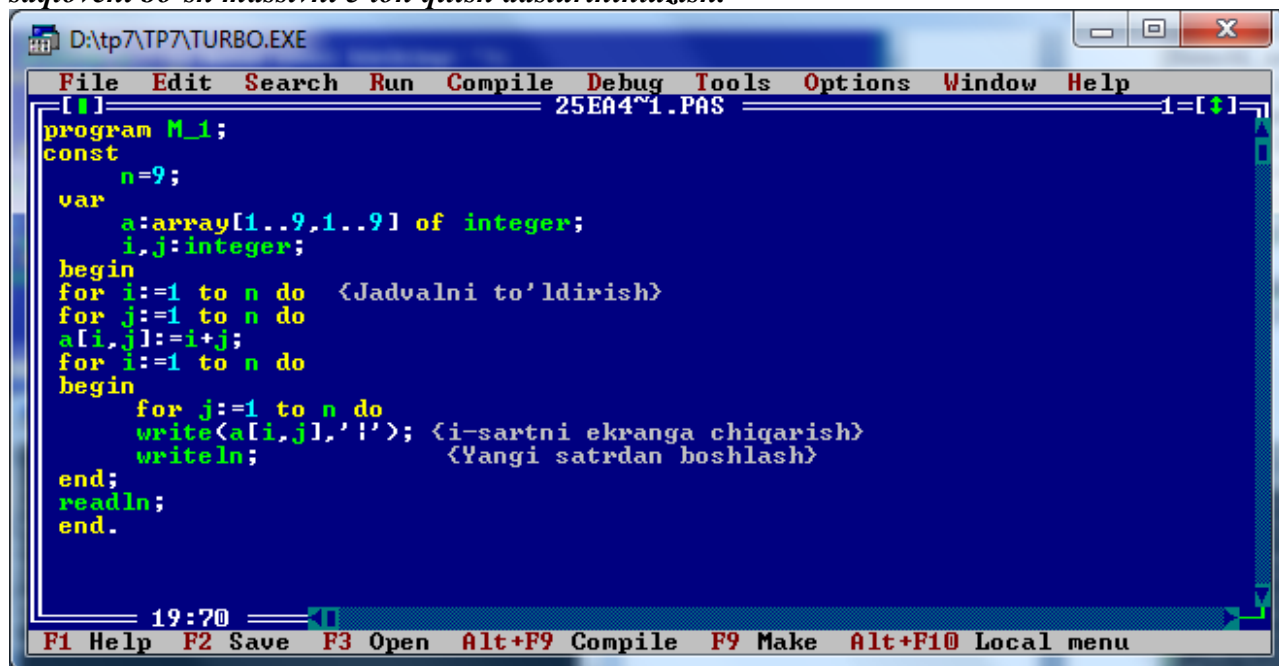
Kompyuter o'z hotirasida 9 ta satr, 9 ta ustundan iborat jadvalda butun solarni saqllovchi bo'sh massivni e'lon qilish uchun:

Var a:array[1..9,1..9] of integer;

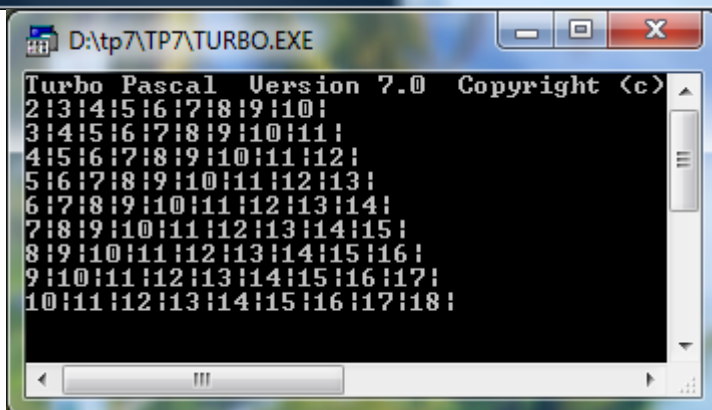
Amaliy ishini bajarish tartibi:

I. O'quvchilar o'qituvchi rahbarligida nazariy ma'lumotlar va mavzu yuzasidan yaratilgan dastur bilan tanishtiriladi (10-15 daq).

1-masala. Kompyuter o'z hotirasida 9 ta satr, 9 ta ustundan iborat jadvalda butun solarni saqllovchi bo'sh massivni e'lon qilish dasturini tuzish.



```
program M_1;
const
  n=9;
var
  a:array[1..9,1..9] of integer;
  i,j:integer;
begin
  for i:=1 to n do {Jadvalni to'ldirish}
    for j:=1 to n do
      a[i,j]:=i+j;
  for i:=1 to n do
    begin
      for j:=1 to n do
        write(a[i,j], ' '); {i-sartni ekranga chiqarish}
      writeln; {Yangi satrdan boshlash}
    end;
  readln;
end.
```



```
Turbo Pascal Version 7.0 Copyright (c)
2 3 4 5 6 7 8 9 10
3 4 5 6 7 8 9 10 11
4 5 6 7 8 9 10 11 12
5 6 7 8 9 10 11 12 13
6 7 8 9 10 11 12 13 14
7 8 9 10 11 12 13 14 15
8 9 10 11 12 13 14 15 16
9 10 11 12 13 14 15 16 17
10 11 12 13 14 15 16 17 18
```

NATIJA:

II. O'quvchilarga quyidagi masalalarni dasturini tuzib, natijalar olish taklif etiladi.

1. Ikki o'lchovli massivning diagonal joylashgan elementlari yig'indisini topish dasturi tuzilsin.
2. Ikki o'lchovli massining o'rta arimetigini hisoblash dasturi tuzilsin.
3. Ikki o'lchovli massivda elementlar yig'indisini topish.
4. Ikki o'lchovli massivning eng kichik elementini va joylashgan o'rni topish dasturi tuzilsin.
5. $A(m,n)$ massivning elementlari yig'indisi eng katta va eng kichik bo'lgan satrlarini topuvchi dastur tuzing.
6. $A(5,5)$ massivning eng katta elementi indeksini topish dasturini tuzing.
7. $A(m,n)$ massivning 3 ga bo'linadigan sonlari yig'indisini topish dasturini tuzing.
8. Pifagor jadvalini chop etuvchi dastur tuzing.
9. M kvadrat massivning bosh diagonalida joylashgan elementlar o'rtacha arifmetigini topish dasturini tuzing.
10. 5×10 kvadrat massiv berilgan. Faqat musbat elementlardan tashkil topgan satrni chop etuvchi dasturini tuzing. Agar bunday satr bo'lmasa 0 ni chop etsin.

III. Amaliy ish himoyasi.

9.3-AMALIY ISH. SATRIY KATTALIKLAR.

MAQSAD: Ma'ruzada olingan bilimlarni mustaxkamlash, dasturlash ko'nikmalarini shakllantirish.

TOPSHIRIQ: berilgan variantlarga ko'ra dasturlar tuzish va ularni Turbo-Pascal dasturlash muhitida bajarish hamda himoya qilish.

NAZARIY MA'LUMOT: Satrlar ASCII kodining belgilarining apostroflar (' ') ichiga olgan ketma-ketligi iborat. Masalan 'mart', '123', va h.k. Satr uzunligi deb unda qatnashayotgan belgilar soniga aytiladi. Bunda chegaralovchi apostrof belgilar hisobga olinmaydi. Masalan, ' ' satrning uzunligi 1 ga teng, '' uzunligi esa 0 ga teng. Satriy o'zgaruvchilar quyidagicha e'lon qilinadi:

Var

O'zgaruvchi_Nomi:string;

yoki

var

O'zgaruvchi_Nomi:string[m];

Bu yerda M – ushbu satrga ajratilgan bayt hisobi. Satrning har bir belgisi hotirada 1 bayt joy egallaydi. Agar satr uzunligi ko'rsatilmagan bo'lsa, ushbu satr 255 belgidan iborat bo'lish mumkin.

«pascal» tilida matnli ma'lumotlar bilan ishlash uchun bir satr funksiya va protseduralar ishlab chiqilgan:

Copy(S:string; start,len:integer):string;

Bu yerda S satrning start belgisidan boshlab len uzunlikdagi satrni belgilaydi (aniqlaydi).

1-MASALA.

```

Program satriy_1;
uses crt;
Var s,s1:string;
Begin
  clrscr;
  S:='algoritm';
  S1:=copy(s,5,4);(s1=ritm)
  Writeln(s1);
  Readln;
End.

```

NATIJA:

```

ritm

```

ekranda ritm so'zi chop etiladi.

Delete (var s:string;start,len:integer);

S satrning start simvolidan boshlab len usunlikdagi satrini o'chiradi.

S:='algoritm';

Delete(s,1,4);

Natija Ritm

Delete(s,3,1); ritm satrini aniqlaydi.

Insert(subs:string;var s:string; start:integer) ;

S satrga start o'rindan boshlab subs satrini qo'shadi.

S:='alm';

Subs:='gorit';

Insert(subs,s,3);

Pos(subs,s:string);byte;

Subs satri s satriga tegishli ekanligini aniqlaydi.

Misol.

S:='algoritm';

Subs:='ritm';

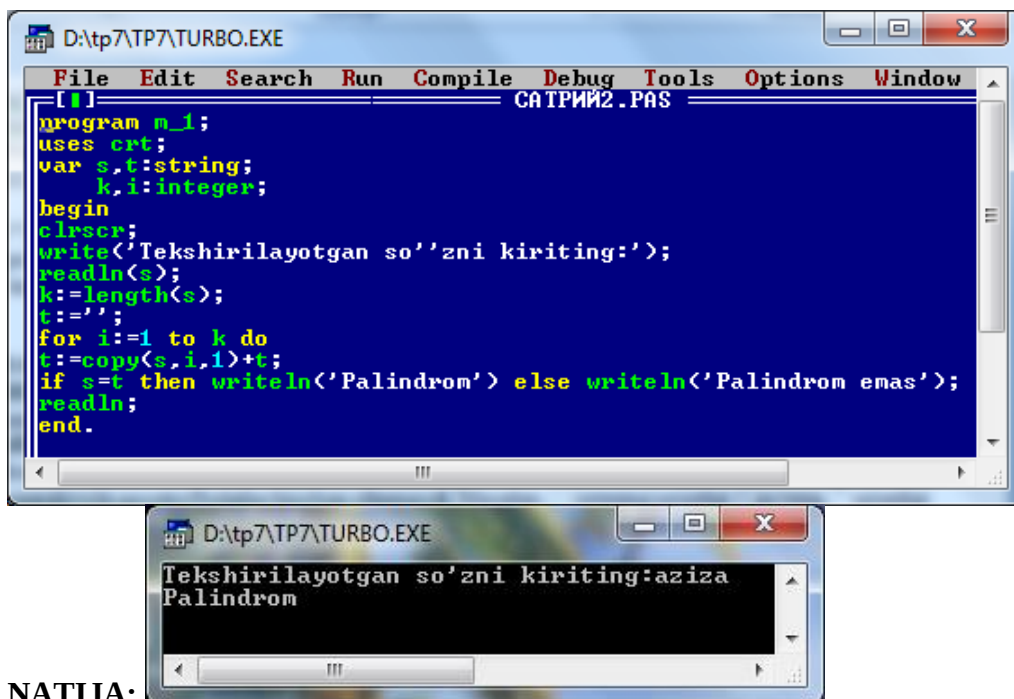
X:=pos(subs,s);

If x<>0 then writeln(subs,' satri ',s,' satrining ',x,' o'rnidan boshlanadi');

Amaliy ishni o'tkazish tartibi.

I. O'qituvchi quyidagi masalani o'quvchilarga tushintirib beradi.

2-MASALA.*Berilgan so'zni Polindrom yoki Polindrom emasligini tekshruvchi dasturni tuzing.*



NATIJA:

Ushbu masalada berilgan so'ni polindromligini topish uchun shu so'zni har bir harfini aniqlash uchun qaytarilish jarayoni ichida `copy(s,i,1)` funksiyasidan foydalaniladi va uni teskarida yig'ib bo'rish uchun `copy(s,i,1)+t` amali bajariladi.

Buni biz katak so'zi misolida ko'rib chiqamiz:

Qaytarilish jarayoni bajarilganda `copy(s,i,1)+t` buyrug'i

k
ak
tak
atak
katak

so'zni hosil qiladi va `if s=t then writeln('Palindrom')` buyrug'i bajarilgandan so'ng ekranda Palindrom javobi paydo bo'ladi.

II .O'quvchilarga quyidagi masalalarni dasturini tuzib, natijalar olish taklif etiladi.

1. Biror so'zda qatnashuvchi belgilar sonini aniqlash va ustun ko'rinishida chop etuvch dastur tuzing

2. Berilgan literli kattalikdagi "a" va "b" harflar soni yig'indisini hisoblovchi dastur tuzing.

3. Berilgan literli kattalikdagi hamma "a" harflarini "b" harfga almashtiruvchi dastur tuzing..

4. Berilgan matndagi har bir harfni ikkilantiruvchi dastur tuzing

5. Qirqib olish va birlashtirish yordamida "sobit" so'zining qismlaridan "obit", "bit", "it" so'zlarini tuzing.

6. NS so'zi KS so'zida necha marta uchrashini aniqlash dasturi tuzilsin.

7. "bol" so'zi berilgan so'zning qismligini aniqlash dasturi tuzilsin. Javob "ha" yoki "yo'q" bo'lishi kerak. Misol uchun "futbol" va "bolalar" so'zlari uchun "ha", "kitob" so'zi uchun "yo'q".

8. Berilgan gapda uchraydigan xar bir so'zning «dan» qo'shimchasini «ning» qo'shimchasi bilan almashtiring.

9. Satrda uchraydigan barcha '+' belgilarisoininianiqlashva '-' belgisiga o'zgartirish dasturini tuzing.

10. Berilgan gap nechta so'zdan iborat ekanligini topish dasturini tuzing.

III. Amaliy ish himoyasi.

4-ilova.Baholash mezonlari va ko'rsatkichlari (ball)

№	F.I.SH.	Nima uchun (1,0)	1-misol (1,0)	2-misol (1,0)	3-misol (1,0)	4-misol (1,0)	Jami ballar (5,0)
---	---------	------------------	---------------	---------------	---------------	---------------	-------------------

1.							
2.							
3.							
4.							
5.							

10-AMALIY MASHG'ULOT: Paskal dasturlash tilining grafik imkoniyatlari.

AMALIY MASHG'ULOTNI OLIB BORISH TEXNOLOGIYASI

Vaqt – 2 soat	Talabalar soni: 25-30 nafar
O'quvmashg'ulotishakli	Nazariy bilimlarni mustahkamlash va amaliy ko'nikmalarni shakllantirish amaliy mashg'uloti
O'quv mashg'ulotirejasi	1. Ekranni ish rejimlari, ekranni grafik holatga o'tkazish 2. Ranglar. Nuqtalar. Chiziqlar. 3. Chizmalarni to'ldirish. Matnlar bilan ishlash. 4. Asosiy grafik protseduralar va funksiyalar.
O'quv mashg'ulotining maqsadi: Amaliy mashg'ulot davomida nazariy bilimlarga asoslangan amaliy ko'nikmalarni shakllantirish.	
O'quv faoliyatining natijalari:	
Pedagogik vazifalar: - amaliy topshiriqlar asosida nazariy bilimlarni mustahkamlash; - individual topshiriqlar asosida mustaqil amaliy ko'nikmalarga ega bo'lishga o'rgatish; - juftlikda ishlash, o'z xulosalarini asoslash ko'nikmalarini rivojlantirish	Talaba: - gapning mulohaza bo'lish, bo'lmasligi, rost yoki yolg'onligini aniqlay oladi;
O'qitish uslubi va texnikasi	Birgalikda o'rganamiz, o'zaro nazorat, munozara.
O'qitish vositalari	Ma'ruza matni, mustaqil ishlar to'plami, o'quv qo'llanmasi, proektor, doska, tarqatma materiallar, slaydlar.
O'qitish shakli	Guruhda, individual, juftlikda o'qitish.
O'qitish sharoitlari	Komp'yuter, proektor, doska bilan ta'minlangan, guruhda dars o'tishga moslashtirilgan auditoriya

AMALIY MASHG'ULOTNING TEXNOLOGIK KARTASI

Bosqichlar, vaqti	Faoliyat mazmuni	
	o'qituvchi	talaba
1-bosqich. Kirish (15 min)	1.1. Mavzuni, uning maqsadi, kutiladigan o'quv natijalarini e'lon qiladi. Mashg'ulot yakka, juftlikda, hamkorlikda ishlash texnologiyasini qo'llagan holda o'tishni ma'lum qiladi. Baholash turlari, mezonlarini e'lon qiladi. 1.2. Mavzu bo'yicha talabalar nazariy bilimlarini umumlashtiradi.	3.5. Mavzuni yozadi va «Nilufar guli» jadvalini to'ldiradi. 3.6. Savollarga javob beradi, eshitadi.
2-bosqich asosiy (55 min)	2.1. Mavzuga doir masala olinadi. Talabalarni o'ylashga undaydi. Ularning fikri tinglanib qo'yilgan masalaning blok sxemasi tuziladi. So'ng Turbo Pascal muhitida dastur matni kiritilib natijasi olinadi 2.2. Talabalarga har biriga yakka tartibda bajarish uchun mustaqil ish topshiriladi. Har bir talaba daftariga mustaqil ishlar to'plamidan o'z variantidagi misollarni yozib oladi. 2.3. Vazifani bajarishda o'quv materiallari (ma'ruza matni, o'quv qo'llanma)laridan foydalanish mumkinligini eslatadi. 2.4. O'zaro nazorat, muhokama amalga oshirilishini eslatadi. Qatorlar oralab talabalarining misol ishlash va o'zaro muhokamasini kuzatadi.	2.1. Tinglaydi, o'ylaydi, savollarga javob beradi, yozib oladi. 2.2. O'z topshirig'ini bajaradi, zarur holda o'qituvchidan yordam so'raydi. 2.3. Tinglaydi, o'ylaydi, savollarga javob beradi, yozib oladi. 2.5. O'z topshirig'ini bajaradi, zarur holda o'qituvchidan yordam so'raydi.
3-bosqich	3.1. Mashg'ulotni yakunlaydi talabalarni baholaydi (4-ilova) va	3.1. Eshitadilar.

Yakuniy (10 min)	faol ishtirokchilarni rag'batlantiradi. 3.2. Mustaqil ish sifatida misol va mashqlar to'plamidagi misollarni ishlab kelishni topshiradi.	3.2Topshiriqni oladilar.
---------------------	--	--------------------------

10-AMALIY ISH.

MAQSAD: Ma'ruzada olingan bilimlarni mustaxkamlash, dasturlash ko'nikmalarini shakllantirish.

TOPSHIRIQ: berilgan variantlarga ko'ra dasturlar tuzish va ularni Turbo-Pascal dasturlash muhitida bajarish hamda himoya qilish.

NAZARIY MA'LUMOT:

Oddiy matnli holatdan grafik holatga o'tish uchun GRAPH modulining InitGraph prosedurasi ishlatiladi.

InitGraph(Gd,Gm,Path); bu erda Gd - drayver nomeri; Gm - rejim nomeri; Path - kerakli drayverga yo'l ko'rsatadi.

Quyidagi dastur grafik holatni ishga tushurib, yana uni yopadi.

Program Gra;

uses Graph; Var Gd,Gm: Integer;

Begin

Gd:=Detect; {avtomatik ravishda drayverni tanlaydi}

InitGraph(Gd,Gm,'c:/Tp'); {grafik holatga o'tish}

Readln; {Enter tugmasini bosishni kutish}

CloseGraph; {gafik holatni yopadi}

End.

PutPixel(x,y,color); -prosedura ekranda koordinatasi (x,y) bo'lgan nuqtani beradi. Color parametri shu nuqtaga rang beradi. Misol: PutPixel(100,120,Red); bu erda ekranning (100,120) nuqta qizil rangda tasvirlanadi. Red=4 nuqtaga qizil rang beradi.

GetPixel(x,y); -funksiya nuqta rangini aniqlaydi. Misol: Col:=GetPixel(50,80); bu erda Col parametriga nuqta rangi yuboriladi.

Line(x1,y1,x2,y2); -prosedura koordinatalari (x1,y1) va (x2,y2) bo'lgan kesmani chizadi.

Circle(x,y,Radius); - markazi (x,y) nuqta bo'lgan va radiusi Radius bo'lgan aylana chizadi.

Arc(x, y, a, b, r); – markazi x va y nuqta bo'lgan yoy chizadi. a va b yoyning boshlang'ich va oxirgi burchaklarini aniqlaydi. R yoy radiusi hisoblanadi.

Rectangle(x1,y1,x2,y2); -prosedura to'g'ri to'rtburchak chizadi. (x1,y1) -yuqori chap burchak koordinatasi; (x2,y2) -pastki o'ng burchak koordinatasi.

SetColor(Color); -prosedura rasmga rang beradi. Color rang nomeri.

Bar(x1,y1,x2,y2); -rangli yoki shtrixlangan to'g'ri to'rtburchak chizish;

Bar3d(x1,y1,x2,y2,depth,top); -rangli yoki shtrixlangan paralelopeped chizish;

FileEllipse(x,y,xradius,yradius); -rangli yoki shtrixlangan ellips chizish;

SetFillStyle(Style,Color); -shtrix va rang berish. Bu erda Style shtrix, Color rang tanlovchi o'zgarmas parametr.

OutTextXY(x,y,TextString); -(x,y) koordinatali nuqtadan TextString nomli matn satrini kiritadi.

**MISOL 1. 14 XIL RANGDA
KESMALARNI CHIZISH DASTURINI TUZING
(QORA VA OQ RANGLARDAN TASHQARI)**

```
Program M_1;
Uses Graph;
Var Gd, gm, c, x, y, i: integer;
Begin Gd:=detect; InitGraph(gd, gm, ' ');
Setbkcolor(white);
ClearDevice; x:=0;
for c:=0 to 14 do
begin x:=x+35; Setcolor(c);
Line(x, 0, x, 400);
end;
```

**MISOL 2. QUYIDAGI
DASTUR TO'G'RI TO'RTBURCHAK
CHIZADI.**

```
Program M_2;
Uses Graph;
Var gd, gm: Integer;
Begin
gd:=detect;
InitGraph(gd, gm, '');
Rectangle(30, 30, 120, 120);
```

II. O'quvchilarga quyidagi masalalarni dasturini tuzib, natijalar olish taklif etiladi.

1. Strofoida taschirinoxosil qiling. $X=a(t^2-1)(t^2+1)$, $Y=at(t^2-1)(t^2+1)$, $a>0$, $t \in (-\infty, +\infty)$
2. Nikomed konxoidi tasvirinoxosil qiling. $X=(a+1)\cos t$, $u=a \lg t + 1\sin t$, $1>0$, $a>0$, $1<a \in (-\pi/2, \pi/2)$.
3. Nikomed konxoidi tasvirini xosil qiling. $X=(a+1)\cos t$, $U=a \lg t + 1\sin t$, $1>0$, $a>0$, $1<a \in (-\pi/2, \pi/2)$.
4. Quyidagi nuqtalar tasvirini xosil qiling. $X=a\cos 3t \sin t$, $Y=a \sin 3t \cos t$, $a>0$, $t \in [0, 2\pi]$.
5. Quyidagi nuqtalar tasvirini xosil qiling. $X=a\cos 3t \sin 5t$, $Y=a \sin 3t \cos 5t$, $a>0$, $t \in [0, 2\pi]$.
6. Quyidagi nuqtalar tasvirini xosil qiling. $X=\cos t(a \cos 5t + b \sin 3t)$; $Y=\sin t(a \cos 5t + b \sin 3t)$, $a>0$, $b>0$, $t \in [0, 2\pi]$.
7. Berilgan xamma topshiriqlardagi xarakat biror bir tugma bosilguncha davom etishi lozim. Aylan bo'ylab xarakatlarni tasvirlarni xosil qilishda qutb koordinatalar sistemasidan foydalanishni tavsiya qilinadi va strelkasi bo'yicha aylan bo'ylab o'zgarmas burchak tezligida xarakat qilayotgan nuqta tasvirini xosil qiling.
8. Ekranda soat miliga qarshi yo'nalishda aylana bo'ylab o'zgarmas burchak tezligida xarakat qilayotgan "O" xarfi tasvirini yasang.
9. O'zining o'rta nuqtasi atrofida soat strelkasi bo'yicha aylanayotgan kesma tasvirini xosil qiling.
10. O'zining o'rta nuqtasi atrofida soat strelkasiga qarshi yo'nalishda aylanayotgan kesma tasvirini xosil qiling.
11. O'zining uchlaridan biri atrofida soat strelkasi bo'yicha aylanayotgan kesma tasvirini xosil qiling.
12. Ekranning pastidan yuqorisiga va aksincha gorizontal yo'nalishda xarakat qilayotgan nuqta tasvirini xosil qiling.
13. Ekranning diogonali bo'yicha to'xtovsiz xarakat qilayotgan nuqta tasvirini xosil qiling. Nuqta ekranning yuqori burchagiga etgandan keyin xarakatini boshlang'ich xolatidan boshlasin.

IV. Amaliy ish himoyasi.

1-ilova. Baholash mezonlari va ko'rsatkichlari (ball)

№	F.I.SH.	Nima uchun (1,0)	1-misol (1,0)	2-misol (1,0)	1-savol (1,0)	2-savol (1,0)	Jami ballar (5,0)
1.							
2.							
3.							
4.							
5.							

Joriy nazorat uchun umumiy savollar

1. Model va uning turlari
2. Iqtisodiy matematik modellar
3. Algoritm tushunchasi
4. Algoritm deyilganda nima tushuniladi? Algoritm so'zi qayerdan kelib chiqadi?
5. Algoritmning qanday xossalari bilasiz? Algoritmni yozishning qanday usullarini bilasiz?
6. Blok-sxema nima? Blok-sxemaning asosiy elementlarini yozib bering.
7. Al-Xorazmiyning hayoti va ijodi haqida nimalarni bilasiz?
8. Masalani echish algoritmini qanday usullar bilan ifodalash mumkin?
9. Algoritmga qo'yiladigan talablarni sanab o'ting.
10. Algoritm nazariyasiga doir tarixiy ma'lumotlar.
11. Algoritm tushunchasini formallashtirish.
12. Masala echimi algoritmini tuzishda qanday qoidalarga amal qilish lozim?
13. Chiziqli algoritm deb nimaga aytiladi?
14. Tarmoqlanuvchi algoritmni ta'riflang va unga hayotiy masalalar keltiring.
15. Takrorlanuvchi algoritmning vazifasini tushuntiring.
16. Algoritm ijrosini tekshirish.
17. Algoritmni ifodalashda blok-sxemalardan foydalanishning afzalliklarini yozib bering.
18. Blok-sxema elementlarini chizib ko'rsating va ularning vazifalarini yozing
19. Algoritmning asosiy xossalari.
20. Algoritmni tasvirlash usullari.
21. Algoritmning asosiy turlari.
22. Ro'yxat nima?
23. Tyuring mashinasi va EHMlar.
24. Rekursiya tushunchasi.
25. Rekkurent algoritm.
26. Rekursiv funksiya tushunchasi.
27. Primitiv rekursiya operatori.
28. Minimizatsiya operatori.
29. Kompyuter dasturlari.
30. Dastur va dasturlash tillari.
31. Dasturlash tillari tarixi.
32. Dasturlash tillarining sintaktik jihatdan turlari.
33. Komp'yuterda masala echish bosqichlari.
34. Yuqori darajali dasturlash tillari.
35. Pascal dasturlash tili muhiti.
36. Pascal dasturlash tili alifbosi.
37. Pascal dasturlash tili tuzilishi.
38. Mantiqiy amallarni rostlik jadvali yordamida izohlang.
39. Identifikator haqida ma'lumot bering.
40. Pascal tilida dasturlar umumiy holda qanday tuzilishga ega?
41. Pascalda dastur qanday qismlardan tashkil topgan bo'ladi?
42. Masalani EHMda echish bosqichlari.
43. O'zgarmas miqdorlar.
44. O'zgaruvchi miqdorlar.

45. Jadval ko‘rinishidagi miqdorlar.
46. Pascal tilida standart funksiyalar va protseduralar.
47. Pascal tilida algebraik ifodalar.
48. O‘zlashtirish operatori.
49. Ma’lumotlarni ekranga chiqarish operatori.
50. Ma’lumotlarni xotiraga kiritish operatori.
51. Matn holatida ekran bilan ishlash.
52. Chiziqli dasturlar tuzish.
53. O‘tish operatorlari.
54. Tarmoqlanish operatorlari.
55. Tanlash operatorlari.
56. Tarmoqlanish operatori nima uchun qo‘llanilishini izohlab bering?
57. Kompilyatsiya va interpretatsiya qilinuvchi tillar.

Oraliq nazorat uchun umumiy savollar

1. Takrorlash operatorlari nima uchun qo‘llaniladi? Misoli bilan ko‘rsating.
2. Parametrli takrorlash operatorining necha xil turlari mavjud? Misoli bilan ko‘rsating.
3. Parametrli takrorlash operatorini tushuntirib bering. Misoli bilan ko‘rsating.
4. Parametrli takrorlash operatorida takrorlash tanasi deb nimaga aytiladi? Misoli bilan ko‘rsating.
5. Break va Continue proseduralari qo‘llanilishini tushuntiring? Misoli bilan ko‘rsating.
6. Takrorlash parametrlari qanday qiymat qabul qiladi? Misoli bilan ko‘rsating.
7. Takrorlashning takrorlanishlar soni qanday aniqlanadi? Misoli bilan ko‘rsating.
8. Repeat operatori takrorlash tanasida qanday operatorlar bo‘lishi mumkin? Misoli bilan ko‘rsating.
9. Repeat operatori necha marta takrorlanadi? Misoli bilan ko‘rsating.
10. Repeat operatoridan qanday chiqiladi? Misoli bilan ko‘rsating.
11. Repeat operatori ishlashini tushuntirib bering. Misoli bilan ko‘rsating.
12. While operatori takrorlash tanasida qanday operatorlar bo‘lishi mumkin? Misoli bilan ko‘rsating.
13. While operatoridan qanday chiqiladi? Misoli bilan ko‘rsating.
14. While operatori ishlashini tushuntirib bering. Misoli bilan ko‘rsating.
15. Massivlar nima uchun qo‘llaniladi?
16. Massivlarning necha xil turlari mavjud?
17. Bir o‘lchovli massivni tushuntirib bering. Misoli bilan ko‘rsating.
18. Ikki o‘lchovli massivni tushuntirib bering. Misoli bilan ko‘rsating.
19. Massiv elementlarini tartiblashning necha xil usullari mavjud?
20. Massiv qanday e‘lon qilinadi? Misoli bilan ko‘rsating.
21. Massivni tartiblashni tanlash usuli qanday? Misoli bilan ko‘rsating.
22. Massivni tartiblashni almashtirish usuli qanday? Misoli bilan ko‘rsating.
23. Satrli ifodalar.
24. Concat funksiyaning vazifasi nimadan iborat? Misoli bilan ko‘rsating.
25. Concat funksiyasining o‘rniga boshqa amaldan foydalanish mumkinmi? Mumkin bo‘lsa tushuntirib bering.
26. Qaysi funksiya satr uzunligini aniqlaydi? Ta’rifi va misoli bilan ko‘rsating.
27. Length funksiyasi nima uchun qo‘llaniladi? Misoli bilan ko‘rsating.
28. Pos funksiyasi qanday vazifani bajaradi? Misoli bilan ko‘rsating.
29. Pos funksiyasining qiymati qachon nolga teng bo‘ladi? Misoli bilan ko‘rsating.
30. Copy funksiyasi nima uchun qo‘llaniladi? Misoli bilan ko‘rsating.
31. Copy funksiyasining qiymati sonli bo‘lishi mumkinmi? Misoli bilan ko‘rsating.

32. Delete protsedurasi nima uchun qo'llaniladi? Misoli bilan ko'rsating.
33. Insert protsedurasi nima uchun qo'llaniladi? Misoli bilan ko'rsating.
34. STR protsedurasi nima uchun qo'llaniladi? Misoli bilan ko'rsating.
35. VAL protsedurasi nima uchun qo'llaniladi? Misoli bilan ko'rsating.
36. Paskalda berilgan satrning qismini o'chiradigan qanday protsedura bor? Misoli bilan ko'rsating.
37. Ord va Chr funksiyalarning vazifasini aytib bering. Misoli bilan ko'rsating.
38. Satrli o'zgaruvchilarning "massiv xususiyati" nimani anglatadi? Misoli bilan ko'rsating.
39. Qismdasturlar nima uchun ishlatiladi? Misoli bilan ko'rsating.
40. Qismdasturlar katta dasturlarni yozishda qanday rol o'ynaydi? Misoli bilan ko'rsating.
41. Qismdasturni ta'rifini bering. Misoli bilan ko'rsating.
42. Protsedura tavsifi. Protsedura operatori. Misoli bilan ko'rsating.
43. Funksiya tavsifi va undan foydalanish. Misoli bilan ko'rsating.
44. Funksiya qanday e'lon qilinadi? Misoli bilan ko'rsating.
45. Funksiyaning protseduradan asosiy farqlari? Misoli bilan ko'rsating.
46. Turbo Paskalda protsedura va funksiya tuzilmasini tavsiflang. Misoli bilan ko'rsating.
47. Parametrli protsedura deb qanday protseduralarga yatiladi? Misoli bilan ko'rsating.
48. Parametrli protseduralarning nechta xil turi mavjud? Misoli bilan ko'rsating.
49. Qanday hollarda protseduradan, qaysi hollarda funksiyadan foydalanish mumkin? Misoli bilan ko'rsating.
50. Protsedura matematik yoki mantiqiy ifoda operandi bo'lishi mumkinmi? Misoli bilan ko'rsating.
51. Qismdasturning qaysi parametrlari formal, qaysilari faktik deb ataladi? Misoli bilan ko'rsating.
52. Bir oyqlama protseduralarni tushuntirib bering. Misoli bilan ko'rsating.
53. Ikki oyqlama protseduralarni tushuntirib bering. Misoli bilan ko'rsating.
54. Qanday parametrlar o'zgaruvchan parametrlar deyiladi? Misoli bilan ko'rsating.
55. Qanday parametrlar qiymat parametrlar deyiladi? Misoli bilan ko'rsating.
56. Formal va faktik parametrlar. Misoli bilan ko'rsating.
57. Parametr-qiymatlar. Misoli bilan ko'rsating.
58. Parametr-o'zgaruvchilar. Misoli bilan ko'rsating.
59. Parametr-konstantalar. Misoli bilan ko'rsating.
60. Tipsiz parametrlar. Misoli bilan ko'rsating.
61. Massivlar va ochiq tipdagi satrlar. Misoli bilan ko'rsating.
62. Parametr – protseduralar va parametr – funksiyalar. Misoli bilan ko'rsating.
63. Qismdasturlar buyruqlari. Misoli bilan ko'rsating.
64. Formal parametrlar nomlari faktik parametrlar nomi bilan ustma-ust tushishi mumkinmi? Misoli bilan ko'rsating.
65. Formal parametrni ta'riflashning qanday usullari sizga ma'lum? Misoli bilan ko'rsating.
66. Dasturda qaysi obyektlar lokal, qaysilari global bo'ladi? Misoli bilan ko'rsating.
67. Lokal o'zgaruvchi nomi global o'zgaruvchi nomi bilan ustma-ust tushishi mumkinmi? Misoli bilan ko'rsating.
68. Xotiraning qaysi modellari dasturlarda ishlatilishi mumkin? Misoli bilan ko'rsating.
69. Xotiraning qanday modeli asosiy dasturning kodi ixtiyoriy segmentidan qismdasturni chaqirishga imkon beradi? Misoli bilan ko'rsating.
70. Qaysi buyruqlar Paskalda qismdasturlarni yozishda foydalanishi mumkin? Misoli bilan ko'rsating.
71. Qaysi hollarda u yoki bu buyruq foydalanishi mumkin? Misoli bilan ko'rsating.
72. Fizik fayl deb qanday fayllarga aytiladi? Misoli bilan ko'rsating.
73. Mantiqiy fayl deb qanday fayllarga aytiladi? Misoli bilan ko'rsating.
74. Fizik va mantiqiy fayllar qanday bog'laniladi? Misoli bilan ko'rsating.
75. Fayllar nima uchun va qanday ochiladi? Misoli bilan ko'rsating.

76. Fayl qanday yopiladi? Misoli bilan ko'rsating.
77. Fayl ko'rsatkichi deb nimaga aytiladi? Misoli bilan ko'rsating.
78. Matnli fayllar qanday e'lon qilinadi? Misoli bilan ko'rsating.
79. Matnli fayllarni toifali fayllardan qanday farqi bor? Misoli bilan ko'rsating.
80. Append va Rewrite protseduralari farqi? Misoli bilan ko'rsating.
81. Assign protsedurasining vazifasi tushuntirib bering? Misoli bilan ko'rsating.
82. Rewrite protsedurasining vazifasi tushuntirib bering? Misoli bilan ko'rsating.
83. Reset protsedurasining vazifasi tushuntirib bering? Misoli bilan ko'rsating.
84. Close protsedurasining vazifasi tushuntirib bering? Misoli bilan ko'rsating.
85. Eof funksiyasining vazifasi tushuntirib bering? Misoli bilan ko'rsating.
86. Read protsedurasining vazifasi tushuntirib bering? Misoli bilan ko'rsating.
87. Write protsedurasining vazifasi tushuntirib bering? Misoli bilan ko'rsating.
88. Matnli fayllar bilan ishlovchi protsedura va funksiyalarni tushuntirib bering? Misoli bilan ko'rsating.
89. Toifalanmagan fayllar bilan ishlovchi protsedura va funksiyalar? Misoli bilan ko'rsating.
90. Modul nima? Misoli bilan ko'rsating.
91. Modullar nima uchun ishlatiladi? Misoli bilan ko'rsating.
92. Standart modullarni sanab bering. Misoli bilan ko'rsating.
93. Foydalanuvchi modullari qanday hosil qilinadi? Misoli bilan ko'rsating.
94. Modulning programmadan farqi nimada? Misoli bilan ko'rsating.
95. Modul sarlavhasi haqida yozing? Misoli bilan ko'rsating.
96. Interfeys bo'limi haqida yozing? Misoli bilan ko'rsating.
97. Realizatsion bo'limi haqida yozing? Misoli bilan ko'rsating.
98. Initsializatsiya bo'limi haqida yozing? Misoli bilan ko'rsating.
99. Crt moduli nima maqsadda qo'llaniladi? Misoli bilan ko'rsating.
100. Crt moduli funksiya va protseduralaridan foydalanish uchun nima qilish lozim? Misoli bilan ko'rsating.
101. Kompyuter ekrani koordinatalar sistemasini tushuntirib bering. Misoli bilan ko'rsating.
102. Ekran foniga necha hil rang tanlash mumkin? Misoli bilan ko'rsating.
103. Mant rangiga necha hil rang tanlash mumkin? Misoli bilan ko'rsating.
104. Crt moduli funksiya va protseduralarini sanab bering. Misoli bilan ko'rsating.
105. Rangni boshqaruvchi protseduralar. Misoli bilan ko'rsating.
106. Ekran bilan ishlovchi protseduralar. Misoli bilan ko'rsating.
107. Ekran bilan ishlovchi funksiyalar. Misoli bilan ko'rsating.
108. Ovozni boshqaruvchi protseduralar. Misoli bilan ko'rsating.
109. Grafik rejim qanday o'rnatiladi? Misoli bilan ko'rsating.
110. Grafika bilan ishlovchi fayl drayverlari qanday kengaytmali bo'ladi? Misoli bilan ko'rsating.
111. Siz ishlayotgan kompyuterda drayverlar qaysi katalogda saqlangan? Misoli bilan ko'rsating.
112. Qachon drayver joriy katalogdan izlaniladi? Misoli bilan ko'rsating.
113. GraphResult funksiyasini tushuntirib bering? Misoli bilan ko'rsating.
114. Grafika bilan ishlovchi protsedura va funksiyalarni sanab bering? Misoli bilan ko'rsating.
115. Chiziqning joriy rangi nima? Misoli bilan ko'rsating.
116. Joriy holat nima? Misoli bilan ko'rsating.
117. Koordinata ko'rsatkichini o'rnatish protsedura va funksiyalari? Misoli bilan ko'rsating.
118. SetLineStyle protsedurasini tushuntirib bering? Misoli bilan ko'rsating.
119. SetFillStyle protsedurasini tushuntirib bering? Misoli bilan ko'rsating.
120. Grafik rejimda matnlar bilan ishlovchi protseduralar? Misoli bilan ko'rsating.

Yakuniy nazorat uchun umumiy savollar

Variant-1

1. Takrorlash operatorlari nima uchun qo'llaniladi?
2. Dasturda qaysi obyektlar lokal, qaysilari global bo'ladi? Misoli bilan ko'rsating.
3. K va N ($N > 0$) butun sonlari berilgan. K sonini N marta chiqaruvchi dastur tuzilsin (for).
4. N ta elementdan tashkil topgan massiv berilgan. Massiv elementlari orasidan o'ng qo'shnisidan katta bo'lganlar sonini chiqaruvchi programma tuzilsin (array).
5. S satr berilgan. Agar S faylning mumkin bo'lgan nomi bo'lsa, u holda shu nomdagi bo'sh fayl hosil qilinsin va TRUE chop qilinsin. Agar S nomdagi faylni yaratish mumkin bo'lmasa u holda FALSE chop qilinsin (file).

Variant-2

1. Parametrli takrorlash operatorini tushuntirib bering.
2. Lokal o'zgaruvchi nomi global o'zgaruvchi nomi bilan ustma-ust tushishi mumkinmi? Misoli bilan ko'rsating.
3. A va B butun sonlari berilgan ($A < B$). A va B sonlar orasidagi barcha butun sonlarni (A va B ni ham) chiqaruvchi va chiqarilgan sonlar sonini chiqaruvchi dastur tuzilsin. (A va B ham chiqarilsin), (for).
4. N ta elementdan tashkil topgan massiv berilgan. Agar massiv 1 dan n gacha bo'lgan sonlarning o'rin almashtirishidan hosil bo'lgan bo'lsa, nol chiqarsin. Aks holda birinchi uchragan nojoiz kiritilgan element indeksini chiqaruvchi programma tuzilsin (array).
5. Fayl nomi va N butun soni berilgan ($N > 1$). Berilgan nomdagi fayl hosil qilinsin va unga N ta birinchi musbat juft sonlari chop qilinsin (2,4,...) (file).

Variant-3

1. Break va Continue proseduralari qo'llanilishini tushuntiring?
2. Xotiraning qaysi modellari dasturlarda ishlatilishi mumkin? Misoli bilan ko'rsating.
3. A va B butun sonlari berilgan ($A < B$). A va B sonlar orasidagi barcha butun sonlarni (A va B kirmaydi) kamayish tartibida chiqaruvchi va chiqarilgan sonlar sonini chiqaruvchi dastur tuzilsin (for).
4. N ta elementdan tashkil topgan massiv berilgan. Massivda eng ko'p qatnashgan bir xil qiymatli element sonini chiqaruvchi programma tuzilsin (array).
5. Fayl nomi va A va D haqiqiy sonlar berilgan. Shu nomdagi fayl hosil qilinsin va unga A boshlang'ich hadi va D farqiga ega bo'lgan arifmetik progressiyaning birinchi 10 ta hadi yozilsin (file). $A, A+D, A+2*D, A+3*D, \dots$

Variant-4

1. Repeat operatori ishlashini tushintirib bering.
2. Xotiraning qanday modeli asosiy dasturning kodi ixtiyoriy segmentidan qismdasturni chaqirishga imkon beradi? Misoli bilan ko'rsating.
3. Bir kg konfet narxi – haqiqiy son berilgan. 1, 2, ..., 10 kg konfet narxini chiqaruvchi dastur tuzilsin (for).
4. n ta elementdan tashkil topgan massiv berilgan. Massivda qatnashgan sonlardan faqat bittadan chiqaruvchi programma tuzilsin (array). Berilgan massiv: 7 4 2 3 1 4 2 4 7; Natija: 7 4 2 3 1 5
5. 4 ta faylning nomi berilgan. Shu fayllarning nechitasi joriy katalogda joylashgani aniqlansin (file).

Variant-5

1. While operatori ishlashini tushintirib bering.

2. Qaysi buyruqlar Pascalda qismdasturlarni yozishda foydalanishi mumkin? Misoli bilan ko'rsating.

3. Bir kg konfet narxi – haqiqiy son berilgan. 0.1, 0.2, ..., 0.9, 1 kg konfet narxini chiqaruvchi dastur tuzilsin (for).

4. N ta elementdan tashkil topgan massiv va R butun soni berilgan. Massiv elementlari orasidan R soniga yig'indisi eng yaqin 2 ta elementni chiqaruvchi programma tuzilsin (array).

5. Butun sonlar fayli berilgan. Shu fayl tarkibiga kiruvchi elementlar soni aniqlansin. Agar bunday fayl mavjud bo'lmasa u holda -1 chop qilinsin (file).

Variant-6

1. Bir o'lchovli massivni tushuntirib bering.

2. Qaysi hollarda u yoki bu buyruq foydalanishi mumkin? Misoli bilan ko'rsating.

3. Bir kg konfet narxi – haqiqiy son berilgan. 1.2, 1.4, ..., 2 kg konfet narxini chiqaruvchi dastur tuzilsin (for).

4. N ta elementdan tashkil topgan massiv berilgan. Massiv elementlari orasidan bir-biriga eng yaqin qo'shnilar indeksini chiqaruvchi programma tuzilsin (ayirmasining moduli eng kichik bo'lgan elementlari indeksi) (array).

5. Manfiy bo'lmagan butun sonlardan iborat fayl va K soni berilgan (K butun). Faylning K-elementi chop qilinsin (elementlar 1 dan boshlab nomerlanadi). Agar bunday element mavjud bo'lmasa, (-1) chop qilinsin (file).

Variant-7

1. Ikki o'lchovli massivni tushuntirib bering.

2. Fizik fayl deb qanday fayllarga aytiladi? Misoli bilan ko'rsating.

3. A va B butun sonlari berilgan ($A < B$). A dan B gacha bo'lgan barcha butun sonlar (A va B ham kiradi) yig'indisini chiqaruvchi dastur tuzilsin (for).

4. n ta elementdan tashkil topgan massiv berilgan. Massiv elementlari orasida aniq 2 ta bir xil qiymatli element bor. Shu bir xil qiymatli elementlar indeksini chiqaruvchi programma tuzilsin (array).

5. Elementlari soni 4 dan ko'p bo'lgan butun sonlardan iborat fayl berilgan. Ushbu faylning birinchi, ikkinchi, oxirgi va oxirgidan bitta oldingi elementlari chop qilinsin (file).

Variant-8

1. Satrli ifodalar.

2. Mantiqiy fayl deb qanday fayllarga aytiladi? Misoli bilan ko'rsating.

3. A va B butun sonlari berilgan ($A < B$). A dan B gacha bo'lgan barcha butun sonlar (A va B ham kiradi) ko'paytmasini chiqaruvchi dastur tuzilsin (for).

4. n ta elementdan tashkil topgan massiv elementlari tartiblangan holda (o'sish yoki kamayish) berilgan. Massivning har hil qiymatli elementlari sonini chiqaruvchi programma tuzilsin (array).

5. Ikkita haqiqiy sonlar fayli berilgan. Shu fayllarning birinchisi bo'sh bo'lmagani ma'lum va ikkinchisi joriy katalogda mavjud emas. Mavjud bo'lmagan fayl yaratilsin va mavjud bo'lgan faylning birinchi va oxirgi elementi shu faylga yozilsin (tartibi buzilmagan holda) (file).

Variant-9

1. Concat funksiyaning vazifasi nimadan iborat?

2. Fizik va mantiqiy fayllar qanday bog'laniladi? Misoli bilan ko'rsating.

3. A va B butun sonlari berilgan ($A < B$). A dan B gacha bo'lgan barcha butun sonlar (A va B ham kiradi) kvadratlarining yig'indisini chiqaruvchi dastur tuzilsin (for).

4. n ta elementda tashkil topgan massiv R butun son berilgan. Massiv elementlari orasidan R soniga yig'indisi eng yaqin 2 ta qo'shni elementni chiqaruvchi programma tuzilsin (array).

5. Ikkita haqiqiy sonlar fayli berilgan. Shu fayllardan biri (birinchisi bo'lishi share mas) bo'sh bo'lmagani ma'lum va boshqasi esa joriy katalogda mavjud emas. Mavjud bo'lmagan fayl yaratilsin va mavjud bo'lgan faylning birinchi va oxirgi elementi shu faylga yozilsin (tartibi buzilmagan holda) (file).

Variant-10

1. Fayllar nima uchun va qanday ochiladi?
2. Concat funksiyasining o'rniga boshqa amaldan foydalanish mumkinmi? Mumkin bo'lsa tushuntirib bering.
3. N ($N > 0$) butun son berilgan. Quyidagi yig'indini hisoblovchi dastur tuzilsin (for):
$$S = 1 + 1/2 + 1/3 + \dots + 1/N.$$
4. n ta elementda tashkil topgan massiv berilgan. Massiv elementlari orasidan yig'indisi eng katta bo'ladigan 2 ta qo'shni elementni chiqaruvchi programma tuzilsin (array).
5. Haqiqiy sonlar fayli berilgan. Berilgan fayl elementlarini teskari tartibda saqllovchi yangi fayl hosil qilinsin (file).

Variant-11

1. Qaysi funksiya satr uzunligini aniqlaydi? Ta'rifi va misoli bilan ko'rsating.
2. Fayl qanday yopiladi? Misoli bilan ko'rsating.
3. N ($N > 0$) butun son berilgan. Quyidagi yig'indini hisoblovchi dastur tuzilsin (for):
$$S = N^2 + (N+1)^2 + (N+2)^2 + \dots + (2N)^2.$$
4. n ta elementdan tashkil topgan massiv va R butun soni berilgan. Massiv elementlari orasidan R soniga eng yaqin sonni topuvchi programma tuzilsin (array). ($|a[k] - R|$ ayirma eng kichik bo'luvchi $a[k]$ topilsin).
5. Haqiqiy sonlar fayli berilgan. Ikkita yangi fayl hosil qilinsin. Ularning biriga boshlang'ich faylning toq nomerdagi elementlarini (1, 3,...), ikkinchisi esa – juft nomerdagi elementlarini (2, 4,...) saqllovchi dastur tuzilsin (file).

Variant-12

1. Length funksiyasi nima uchun qo'llaniladi? Misoli bilan ko'rsating.
2. Fayl ko'rsatkichi deb nimaga aytiladi? Misoli bilan ko'rsating.
3. N ($N > 0$) butun son berilgan. Quyidagi ko'paytmani hisoblovchi dastur tuzilsin (for):
$$P = 1.1 * 1.2 * 1.3 * \dots (N \text{ ta ko'paytuvchi}).$$
4. n ta elementdan tashkil topgan massiv berilgan. Massiv elementlari orasidan monoton oraliqlar sonini chiqaruvchi programma tuzilsin (array).
5. Butun sonlar fayli berilgan. Ikkita yangi fayl hosil qilinsin. Ularning birinchisi boshlang'ich faylning juft sonlarini, ikkinchisi esa toq sonlarini o'zida saqlasin. Agar boshlang'ich faylda juft yoki toq sonlar mavjud bo'lmasa u holda natijaviy fayl bo'sh holda qoldirilsin (file).

Variant-13

1. Pos funksiyasi qanday vazifani bajaradi? Misoli bilan ko'rsating.
2. Matnli fayllar qanday e'lon qilinadi? Misoli bilan ko'rsating.
3. N ($N > 0$) butun son berilgan. Quyidagi ifodani hisoblovchi dastur tuzilsin (for): $S = 1.1 - 1.2 + 1.3 - \dots$ (N ta qo'shiluvchi ishoralar almashib keladi. Shartli operatorlardan foydalanmang).
4. n ta elementdan tashkil topgan massiv berilgan. Massiv elementlari orasidan monoton okamayuvchi oraliqlar sonini chiqaruvchi programma tuzilsin (array).
5. Butun sonlar fayli berilgan. Ikkita yangi fayl hosil qilinsin. Ularning birinchisi boshlang'ich faylning musbat sonlaridan (teskari tartibda), ikkinchisi esa manfiy sonlaridan (teskari tartibda) iborat bo'lsin. Agar boshlang'ich faylning manfiy yoki musbat sonlari mavjud bo'lmasa u holda natijaviy fayl bo'sh holda qoldirilsin (file).

Variant-14

1. Pos funksiyasining qiymati qachon nolga teng bo'ladi? Misoli bilan ko'rsating.
2. Matnli fayllarni toifali fayllardan qanday farqi bor? Misoli bilan ko'rsating.
3. N ($N > 0$) butun son berilgan. Shu sonning kvadratini quyidagi formula asosida hisoblovchi dastur tuzilsin (for): $N^2 = 1 + 3 + 5 + \dots + (2N - 1)$. har bir qo'shiluvchidan keyin natijani ekranga chiqarib boring. Natijada ekranda 1 dan N gacha bo'lgan sonlarning kvadratlari chiqariladi.
4. n ta elementdan tashkil topgan massiv berilgan. Massiv elementlari orasidan monoton o'suvchi oraliqlar sonini chiqaruvchi programma tuzilsin (array).
5. Haqiqiy sonlar fayli berilgan. Shu fayl elementlarining o'rta arifmetigi topilsin (file).

Variant-15

1. Copy funksiyasi nima uchun qo'llaniladi? Misoli bilan ko'rsating.
2. Append va Rewrite protseduralari farqi? Misoli bilan ko'rsating.
3. A haqiqiy va N ($N > 0$) butun sonlari berilgan. A ning N - darajasini aniqlovchi dastur tuzilsin (for): $A^N = A * A * \dots * A$
4. n ta elementdan tashkil topgan massiv berilgan. Massiv lokal minimum yoki lokal maksimum bo'lmagan elementlari orasidan kattasini chiqaruvchi programma tuzilsin. Agar bunday element bo'lmasa nol chiqarilsin. Lokal minimum – o'ng va chap qo'shnisidan kichik bo'lgan element. Lokal maksimum – o'ng va chap qo'shnisidan katta bo'lgan element (array).
5. Haqiqiy sonlar fayli berilgan. Shu faylning juft nomerdagi elementlari yig'indisi hisoblansin (file).

Variant-16

1. Copy funksiyasining qiymati sonli bo'lishi mumkinmi? Misoli bilan ko'rsating.
2. Assign protsedurasining vazifasi tushuntirib bering? Misoli bilan ko'rsating.
3. A haqiqiy va N ($N > 0$) butun sonlari berilgan. Bitta ssikldan foydalanib A ning 1 dan N gacha bo'lgan barcha darajasini chiqaruvchi dastur tuzilsin (for).
4. n ta elementdan tashkil topgan massiv berilgan. Massiv lokal maksimumlari orasidan kichigini chiqaruvchi programma tuzilsin. Lokal maksimum – o'ng va chap qo'shnisidan katta bo'lgan element (array).
5. N va K butun musbat sonlar va fayl nomi berilgan. Yangi matnli fayl hosil qilinsin va unga N ta satr va har bir satr K ta "*" (yulduzcha) belgisidan iborat bo'lsin (text).

Variant-17

1. Delete protsedurasi nima uchun qo'llaniladi? Misoli bilan ko'rsating.
2. Rewrite protsedurasining vazifasi tushuntirib bering? Misoli bilan ko'rsating.
3. A haqiqiy va N ($N > 0$) butun sonlari berilgan. Bitta ssikldan foydalanib quyidagi A ning 1 dan N gacha bo'lgan barcha darajalarini chiqaruvchi va yig'indini hisoblash dasturi tuzilsin (for): $S = 1 + A + A^2 + \dots + A^N$.
4. n ta elementdan tashkil topgan massiv berilgan. Massiv lokal minimumlari orasidan kattasini chiqaruvchi programma tuzilsin. Lokal minimum – o'ng va chap qo'shnisidan kichik bo'lgan element (array).
5. N ($0 < N < 27$) butun son va fayl nomi berilgan. Berilgan nomdagi matnli fayl hosil qilinsin va unga: birinchi satri "a" kichik lotin harfi, ikkinchisiga "ab", uchinchisiga "abc" va h.k.satrlarni saqlovchi N ta satr yozilsin (text).

Variant-18

1. Insert protsedurasi nima uchun qo'llaniladi? Misoli bilan ko'rsating.
2. Reset protsedurasining vazifasi tushuntirib bering? Misoli bilan ko'rsating.
3. X haqiqiy va N ($N > 0$) butun sonlari berilgan. Bitta ssikldan foydalanib quyidagi X ning 1 dan N gacha bo'lgan barcha darajalarini chiqaruvchi va yig'indini hisoblovchi dastur tuzilsin (for): $S = 1 - X + X^2 - X^3 + \dots + (-1)^N X^N$. Shartli operatorlardan foydalanmang..

4. n ta elementdan tashkil topgan massiv berilgan. Massiv elementlari orasidan oxirgi lokal maksimum elementi indeksini chiqaruvchi programma tuzilsin. Lokal maksimum – o'ng va chap qo'shnisidan katta bo'lgan element (array).

5. N ($0 < N < 27$) butun son va fayl nomi berilgan. Berilgan nomdagi fayl hosil qilinsin va unga uzunligi N ga teng bo'lgan N ta satr quyidagicha yozilsin: K -nomerdagi satr ($K=1, \dots, N$) katta lotin harflarining boshlang'ich K ta harfini va undan o'ngda "*" belgisidan iborat bo'lsin. Masalan $N=4$ uchun fayl quyidagi satrlardan iborat bo'lishi kerak. "A***", "AB**", "ABC*", "ABCD" (text).

Variant-19

1. STR protsedurasi nima uchun qo'llaniladi? Misoli bilan ko'rsating.

2. Close protsedurasining vazifasi tushuntirib bering? Misoli bilan ko'rsating.

3. N ($N > 0$) butun soni berilgan. Birdan N gacha bo'lgan natural sonlari ko'paytmasini chiqaruvchi dastur tuzilsin (for): $N! = 1 * 2 * \dots * N$. 1 dan N gacha bo'lgan natural sonlari ko'paytmasi N faktorial deyiladi.

4. n ta elementdan tashkil topgan massiv berilgan. Massiv elementlari orasidan birinchi uchragan lokal minimum elementi indeksini chiqaruvchi programma tuzilsin. Lokal minimum – o'ng va chap qo'shnisidan kichik bo'lgan element (array).

5. Matnli fayl berilgan. Uning tarkibiga kiruvchi satrlar va belgilar soni (miqdori) chop qilinsin (satrning oxiri EOLN va EOF fayl oxirlari markerlari belgilarni sanayotganda hisobga olinmasin) (text).

Variant-20

1. VAL protsedurasi nima uchun qo'llaniladi? Misoli bilan ko'rsating.

2. Eof funksiyasining vazifasi tushuntirib bering? Misoli bilan ko'rsating.

3. N ($N > 0$) butun sonlari berilgan. Bitta ssikldan foydalanib quyidagi yig'indini hisoblash dasturi tuzilsin (for): $S = 1! + 2! + 3! + \dots + N!$ ($N!$ ifoda - N faktorial - 1 dan N gacha bo'lgan butun sonlari ko'paytmasini bildiradi: $N! = 1 * 2 * \dots * N$).

4. n ta elementdan tashkil topgan massiv berilgan. Massiv elementlari orasida, chap qo'shnisidan katta bo'lgan elementlarining indeksini kamayish tartibida chiqaruvchi va ularning sonini chiqaruvchi programma tuzilsin (array).

5. Satr va matnli fayl berilgan. S satr fayl oxiriga qo'shilsin (text).

Variant-21

1. Paskalda berilgan satrning qismini o'chiradigan qanday protsedura bor? Misoli bilan ko'rsating.

2. Read protsedurasining vazifasi tushuntirib bering? Misoli bilan ko'rsating.

3. N ($N > 0$) butun sonlari berilgan. Bitta ssikldan foydalanib quyidagi yig'indini hisoblash dasturi tuzilsin (for): $S = 1 + 1/1! + 1/2! + 1/3! + \dots + 1/N!$ ($N!$ ifoda - N faktorial - 1 dan N gacha bo'lgan butun sonlari ko'paytmasini bildiradi: $N! = 1 * 2 * \dots * N$). Olingan natija $e = \exp(1)$ o'zgarmaning taqribiy qiymatiga teng bo'ladi.

4. n ta elementdan tashkil topgan massiv berilgan. Massiv elementlari orasida, o'ng qo'shnisidan katta bo'lgan elementlarining indeksini o'sish tartibida chiqaruvchi va ularning sonini chiqaruvchi programma tuzilsin (array).

5. Ikkita matnli fayl berilgan. Birinchi fayl oxiriga ikkinchi fayl qo'shilsin (text).

Variant-22

1. Ord va Chr funksiyalarning vazifasini aytib bering. Misoli bilan ko'rsating.

2. Write protsedurasining vazifasi tushuntirib bering? Misoli bilan ko'rsating.

3. X haqiqiy va N ($N > 0$) butun sonlari berilgan. Quyidagi ifoda qiymatini hisoblash dasturi tuzilsin (for): $1 + X/1! + X^2/2! + X^3/3! + \dots + X^N/N!$ ($N! = 1 * 2 * \dots * N$). Olingan natija $e^X = \exp(X)$ ga taqribiy qiymatiga teng bo'ladi.

4. n ta elementdan tashkil topgan massiv berilgan. Massiv toq indeksli elementlari orasidan kattasini aniqlovchi programma tuzilsin (array). $\max(a_1, a_3, a_5, \dots)$.
5. S satr va matnli fayl berilgan. S satr fayl boshiga qo'shilsin (text).

Variant-23

1. Satrli o'zgaruvchilarning "massiv xususiyati" nimani anglatadi? Misoli bilan ko'rsating.
2. Matnli fayllar bilan ishlovchi protsedura va funksiyalarni tushuntirib bering? Misoli bilan ko'rsating.
3. X haqiqiy va N ($N > 0$) butun sonlari berilgan. Quyidagi ifoda qiymatini hisoblash dasturi tuzilsin (for): $X - X^3/3! + X^5/5! - \dots + (-1)^N X^{2N+1}/((2N+1)!) \quad (N! = 1 \cdot 2 \cdot \dots \cdot N)$. Olingan natija $\sin(X)$ ga taqribiy qiymatiga teng bo'ladi.
4. n ta elementdan tashkil topgan massiv berilgan. Massiv juft indeksli elementlari orasidan kichigini aniqlovchi programma tuzilsin (array). $\min(a_0, a_2, a_4, \dots)$.
5. Ikkita matnli fayl berilgan. Birinchi fayl boshiga ikkinchi fayl qo'shilsin (text).

Variant-24

1. Qismdasturlar nima uchun ishlatiladi? Misoli bilan ko'rsating.
2. Toifalanmagan fayllar bilan ishlovchi protsedura va funksiyalar? Misoli bilan ko'rsating.
3. X haqiqiy va N ($N > 0$) butun sonlari berilgan. Quyidagi ifoda qiymatini hisoblash dasturi tuzilsin (for): $1 - X^2/2! + X^4/4! - \dots + (-1)^N X^{2N}/(2N)! \quad (N! = 1 \cdot 2 \cdot \dots \cdot N)$. Olingan natija $\cos(X)$ ga taqribiy qiymatiga teng bo'ladi.
4. n ta elementdan tashkil topgan massiv berilgan. Massivda musbat va manfiy qiymatlar ketma-ket kelishini tekshiruvchi programma tuzilsin. Ketma-ketlik bajarilsa nol chiqarsin. Aks holda ketma-ketlikni buzgan birinchi element indeksi chiqarilsin (array).
5. K butun soni va matnli fayl berilgan. K -nomerdagi satrdan oldin bo'sh satr qo'shilsin. Agar bunday nomerli satr mavjud bo'lmasa, u holda fayl o'zgartirishsiz qoldirilsin (text).

Variant-25

1. Qismdasturlar katta dasturlarni yozishda qanday rol o'ynaydi? Misoli bilan ko'rsating.
2. Modul nima? Misoli bilan ko'rsating.
3. X ($|X| < 1$) haqiqiy va N ($N > 0$) butun sonlari berilgan. Quyidagi ifoda qiymatini hisoblash dasturi tuzilsin (for): $X - X^2/2 + X^3/3 - \dots + (-1)^{N-1} X^N/N$. Olingan natija $\ln(1+X)$ ga taqribiy qiymatiga teng bo'ladi.
4. n ta elementdan tashkil topgan massiv berilgan. Massivda juft va toq elementlar ketma-ket kelishini tekshiruvchi programma tuzilsin. Ketma-ketlik bajarilsa nol chiqarsin. Aks holda ketma-ketlikni buzgan birinchi element indeksi chiqarilsin (array).
5. K butun soni va matnli fayl berilgan. K -nomerdagi satrdan keyin bo'sh satr qo'shilsin. Agar bunday nomerli satr mavjud bo'lmasa, u holda fayl o'zgartirishsiz qoldirilsin (text).

Variant-26

1. Qismdasturni ta'rifini bering. Misoli bilan ko'rsating.
2. Modullar nima uchun ishlatiladi? Misoli bilan ko'rsating.
3. X ($|X| < 1$) haqiqiy va N ($N > 0$) butun sonlari berilgan. Quyidagi ifoda qiymatini hisoblash dasturi tuzilsin (for): $X - X^3/3 + X^5/5 - \dots + (-1)^{N-1} X^{2N-1}/(2N+1)$. Olingan natija $\arctg(X)$ ga taqribiy qiymatiga teng bo'ladi.
4. n ta elementdan tashkil topgan massiv berilgan. Massiv elementlari geometrik progressiyani tashkil qilsa, maxrajni aks holda nolni chiqaruvchi programma tuzilsin (array).
5. Matnli fayl berilgan. Undagi barcha bo'sh satrlar ikkilantirilsin (text).

Variant-27

1. Standart modullarni sanab bering. Misoli bilan ko'rsating.

2. Protsedura tavsifi. Protsedura operatori. Misoli bilan ko'rsating.
3. X ($|x| < 1$) xaqiqiy va N ($N > 0$) butun sonlari berilgan. Quyidagi ifoda qiymatini hisoblash dasturi tuzilsin (for): $X + 1 * X^3 / (2 * 3) + 1 * 3 * X^5 / (2 * 4 * 5) + \dots + 1 * 3 * \dots * (2N - 1) X^{2n+1} / (2 * 4 * \dots * (2N) * (2N + 1))$.
4. n ta elementdan tashkil topgan massiv berilgan. Massiv elementlari arifmetik progressiyani tashkil qilsa, ayirmani aks holda nolni chiqaruvchi programma tuzilsin (array).
5. S satr va matnli fayl berilgan. Fayldagi barcha bo'sh satrlar S satrga o'zgartirilsin (text).

Variant-28

1. Funksiya tavsifi va undan foydalanish. Misoli bilan ko'rsating.
2. Foydalanuvchi modullari qanday hosil qilinadi? Misoli bilan ko'rsating.
3. X ($|x| < 1$) xaqiqiy va N ($N > 0$) butun sonlari berilgan. Quyidagi ifoda qiymatini hisoblash dasturi tuzilsin (for): $1 + X/2 - 1 * X^2 / (2 * 4) + 1 * 3 * X^3 / (2 * 4 * 6) - \dots + (-1)^{N-1} * 1 * 3 * \dots * (2N - 3) X^{2n} / (2 * 4 * \dots * (2N))$. Olingan natija $\sqrt{1+x}$ ga taqribiy qiymatiga teng bo'ladi.
4. N ta elementdan tashkil topgan massiv va K, L butun sonlari berilgan. ($0 \leq K \leq L < N$). Massivning K va L indeksleri orasidagi elementlaridan tashqari elementlarining o'rta arifmetigini chiqaruvchi programma tuzilsin (array).
5. Bo'sh bo'lmagan matnli fayl berilgan. Undagi birinchi satr o'chirilsin (text).

Variant-29

1. Modulning programmadan farqi nimada? Misoli bilan ko'rsating.
2. Funksiya qanday e'lon qilinadi? Misoli bilan ko'rsating.
3. N butun soni va sonlar o'qida 2 ta A, B nuqta berilgan. (A va B haqiqiy son). $[A, B]$ kesmani teng N ta kesmaga bo'ling. $[A, B]$ kesmada ajratilgan barcha nuqtalarni chiqaring (for).
4. N ta elementdan tashkil topgan massiv va K, L butun sonlari berilgan. ($0 \leq K \leq L < N$). Massivning K va L indeksleri orasidagi elementlaridan tashqari elementlari yig'indisini chiqaruvchi programma tuzilsin (array).
5. Bo'sh bo'lmagan matnli fayl berilgan. Undagi oxirgi satr o'chirilsin (text).

Variant-30

1. Funksiyaning protseduradan asosiy farqlari? Misoli bilan ko'rsating.
2. Modul sarlavhasi haqida yozing? Misoli bilan ko'rsating.
3. N butun soni va sonlar o'qida 2 ta A, B nuqta berilgan. (A va B haqiqiy son). $[A, B]$ kesmani teng N ta kesmaga bo'ling. $[A, B]$ kesmada ajratilgan barcha nuqtalar uchun $F(X) = 1 - \sin(X)$ funksiya qiymatini hisoblang (for).
4. N ta elementdan tashkil topgan massiv va K, L butun sonlari berilgan. ($0 \leq K \leq L < N$). Massivning K va L indeksleri orasidagi elementlarining o'rta arifmetigini chiqaruvchi programma tuzilsin (array).
5. K butun soni va matnli fayl berilgan. Undagi K -nomerdagi satr o'chirilsin. Agar faylda bunday nomerdagi satr mavjud bo'lmasa, u holda fayl o'zgartirishsiz qoldirilsin (text).

Variant-31

1. Interfeys bo'limi haqida yozing? Misoli bilan ko'rsating.
2. Turbo Pascalda protsedura va funksiya tuzilmasini tavsiflang. Misoli bilan ko'rsating.
3. N butun soni berilgan ($N > 0$). Quyidagi ketma-ketlikning dastlabki N ta hadini chiqaruvchi dastur tuzilsin (for).
 $A_0 = 2;$ $A_K = 2 + 1/A_{K-1};$ $K = 1, 2, \dots$
4. N ta elementdan tashkil topgan massiv va K, L butun sonlari berilgan. ($0 \leq K \leq L < N$). Massivning K va L indeksleri orasidagi elementlari yig'indisini chiqaruvchi programma tuzilsin (array).
5. Matnli fayl berilgan. Undagi barcha bo'sh satrlar o'chirilsin (text).

Variant-32

1. Parametrli protsedura deb qanday protseduralarga yatiladi? Misoli bilan ko'rsating.
2. Realizatsion bo'limi haqida yozing? Misoli bilan ko'rsating.
3. N butun soni berilgan ($N > 0$). Quyidagi ketma-ketlikning dastlabki N ta hadini chiqaruvchi dastur tuzilsin (for). $A_0 = 1$; $A_K = (A_{K-1} + 1)/K$; $K = 1, 2, \dots$
4. n ta elementdan tashkil topgan massiv berilgan. Massiv oxirgi elementidan kichkina va dastlabki elementidan kata bo'lgan oxirgi element indeksini chiqaruvchi programma tuzilsin. Agar bunday element bo'lmasa, nol chiqarsin (array). ($a[0] < a[k] < a[n-1]$).
5. Ikkita mantli fayl berilgan. Birinchi faylning har bir satridan ikkinchi fayldagi mos satrlar qo'shilsin. Agar ikkinchi fayl birinchi fayldan kalta bo'lsa, u holda qolgan satrlar o'zgartirishsiz qoldirilsin (text).

Variant-33

1. Initsializatsiya bo'limi haqida yozing? Misoli bilan ko'rsating.
2. Parametrli protseduralarning nechta xil turi mavjud? Misoli bilan ko'rsating.
3. N butun soni berilgan ($N > 1$). Fibonachchi ketma-ketlikning dastlabki N ta hadini chiqaruvchi dastur tuzilsin (for). $F_1 = 1$; $F_2 = 1$; $F_K = F_{K-2} + F_{K-1}$; $K = 3, 4, \dots$
4. n ta elementdan tashkil topgan massiv berilgan. Massiv oxirgi elementidan kichkina bo'lgan birinchi elementni chiqaruvchi programma tuzilsin. Agar bunday element bo'lmasa, nol chiqarsin (array).
5. K butun soni va matnli fayl berilgan. Faylning har bir satridan birinchi k ta belgi o'chirilsin (agar satr uzunligi K dan kichik bo'lsa, u holda satrning hamma belgilari o'chirilsin) (text).

Variant-34

1. Qanday hollarda protseduradan, qaysi hollarda funksiyadan foydalanish mumkin? Misoli bilan ko'rsating.
2. Crt moduli nima maqsadda qo'llaniladi? Misoli bilan ko'rsating.
3. N butun soni berilgan ($N > 1$). Quyidagi ketma-ketlikning dastlabki N ta hadini chiqaruvchi dastur tuzilsin (for). $A_1 = 1$; $A_2 = 2$; $A_K = (A_{K-2} + 2 * A_{K-1})/3$; $K = 3, 4, \dots$
4. n ta elementdan tashkil topgan massiv berilgan. Massiv elementlari quyidagicha chiqaruvchi dastur tuzilsin (array). $A[0], A[1], A[n-1], A[n-2], A[3], A[4], A[n-3], A[n-4], \dots$
5. Matnli fayl berilgan. Fayldagi hamma katta lotin harflari kichik harflarga va aksincha, barcha kichik lotin harflari katta harflarga almashtirilsin (text).

Variant-35

1. Crt moduli funksiya va protseduralaridan foydalanish uchun nima qilish lozim? Misoli bilan ko'rsating.
2. Protседura matematik yoki mantiqiy ifoda operandi bo'lishi mumkinmi? Misoli bilan ko'rsating.
3. N butun soni berilgan ($N > 2$). Quyidagi ketma-ketlikning dastlabki N ta hadini chiqaruvchi dastur tuzilsin (for). $A_1 = 1$; $A_2 = 2$; $A_3 = 3$; $A_K = A_{K-1} + A_{K-2} * 2 * A_{K-3}$; $K = 4, 5, \dots$
4. n ta elementdan tashkil topgan massiv berilgan. Massiv elementlari quyidagicha chiqaruvchi dastur tuzilsin (array). $A[0], A[n-1], A[1], A[n-2], A[2], A[n-3], \dots$
5. Matnli fayl berilgan. Undagi barcha ketma-ket kelgan probellar bitta probelga almashtirilsin (text).

Variant-36

1. Qismdasturning qaysi parametrlari formal, qaysilari faktik deb ataladi? Misoli bilan ko'rsating.
2. Kompyuter ekrani koordinatalar sistemasini tushuntirib bering. Misoli bilan ko'rsating.
3. N va K burun sonlari berilgan. Quyidagi yig'indini chiqaruvchi dastur tuzilsin (for). $1^K + 2^K + \dots + N^K$.
4. n ta elementdan tashkil topgan massiv berilgan (n juft son). Dastlab massiv elementlari orasidan toq indeksli elementlarni o'sish tartibida keyin juft indeksli elementlarni kamayish tartibida chiqaruvchi dastur tuzilsin. A[1], A[3], A[5],...A[6], A[4], A[2], A[0]. Shart operatori ishlatilmasin (array).
5. Ixtiyoriy sonning 3-darajasini hisoblovchi dastur tuzing (proc).

Variant-37

1. Bir oyqlama protseduralarni tushuntirib bering. Misoli bilan ko'rsating.
2. Ekran foniga necha hil rang tanlash mumkin? Misoli bilan ko'rsating.
3. N va K burun sonlari berilgan. Quyidagi yig'indini chiqaruvchi dastur tuzilsin (for). $1^1 + 2^2 + \dots + N^N$
4. n ta elementdan tashkil topgan massiv berilgan. Dastlab massiv elementlari orasidan juft indeksli elementlarni keyin toq indeksli elementlarni chiqaruvchi dastur tuzilsin. A[0], A[2], A[4],...A[1], A[3], A[5], Shart operatori ishlatilmasin (array).
5. Ixtiyoriy sonning 2,3,4- darajasini hisoblovchi dastur tuzing. Bu protsedura besh marta ixtiyoriy sonning 2,3,4-darajalarini hisoblagandan so'ng tuxtasin (proc).

Variant-38

1. Ikki oyqlama protseduralarni tushuntirib bering. Misoli bilan ko'rsating.
2. Mant rangiga necha hil rang tanlash mumkin? Misoli bilan ko'rsating.
3. N va K burun sonlari berilgan. Quyidagi yig'indini chiqaruvchi dastur tuzilsin (for). $1^N + 2^{N-1} + \dots + N^1$
4. n ta elementdan tashkil topgan massiv berilgan (n toq son). Massiv elementlari orasidan quyidagilarni chiqaruvchi dastur tuzilsin. A[n-1], A[n-3], ..., A[1]. Shart operatori ishlatilmasin (array).
5. 2 ta sonning o'rta arifmetigi va geometrigini hisoblovchi MEAN nomli prosedura hosil qiling. MEAN prosedura orqali A,B,C,D sonlaridan (A,B), (A,C), (A,D) juftliklarining o'rta arifmetigi va geometrigini hisoblovchi dastur tuzilsin (proc).

Variant-39

1. Qanday parametrlar o'zgaruvchan parametrlar deyiladi? Misoli bilan ko'rsating.
2. Crt moduli funksiya va protseduralarini sanab bering. Misoli bilan ko'rsating.
3. A va B butun soni berilgan (A<B). A va B sonlari orasidagi barcha butun sonlarni chiqaruvchi dastur tuzilsin. Bunda har bir son o'zining qiymaticha chiqarilsin. Ya'ni 3 soni 3 marta chiqarilsin (for).
4. n ta elementdan tashkil topgan massiv berilgan (n juft son). Massiv elementlari orasidan quyidagilarni chiqaruvchi dastur tuzilsin. A[0], A[2], A[4], ... Shart operatori ishlatilmasin (array).
5. Teng tomonli 3 burchakning yuzasi va perimetrini hisoblovchi Triangle nomli protsedura hosil qiling, Triangle protsedurasini orqali 3 ta teng tomonli uchburchakning perimetri va yuzini hisoblovchi dastur tuzilsin (proc).

Variant-40

1. Qanday parametrlar qiymat parametrlar deyiladi? Misoli bilan ko'rsating.
2. Rangni boshqaruvchi protseduralar. Misoli bilan ko'rsating.
3. A va B butun soni berilgan (A<B). A va B sonlari orasidagi barcha butun sonlarni chiqaruvchi dastur tuzilsin. Bunda A soni 1 marta, (A+1) soni 2 marta chiqariladi va xokazo (for).

4. n ta elementdan tashkil topgan massiv va K butun soni berilgan ($1 \leq K < n$). Massiv elementlari orasidan indeksi K ga karralilarini chiqaruvchi dastur tuzilsin. $A_k, A_{2k}, A_{3k}, \dots$ Shart operatori ishlatilmasin (array).

5. Natural sonning raqamlari soni va raqamlari yig'indisi hisoblovchi DigitCountSum nomli protsedura hosil qiling. Bu protsedura orqali a, b, c sonlarining raqamlari soni va raqamlari yig'indisi hisoblovchi dastur tuzilsin(proc).

Variant-41

1. Ekran bilan ishlovchi protseduralar. Misoli bilan ko'rsating.
2. Formal va faktik parametrlar. Misoli bilan ko'rsating.
3. A, B, C musbat butun sonlari berilgan. $A \times B$ to'rtburchak ichida tomoni C bo'lgan kvadratdan nechitasi sig'ishini aniqlovchi dastur tuzilsin. Ko'paytirish va bo'lish amallarini ishlatmang (while).

4. n ta elementdan tashkil topgan massiv berilgan. Dastlab massiv elementlari orasidan juftlarini indeksleri o'sish tartibida chiqaruvchi, keyin massiv elementlari orasidan toqlarini indeksleri kamayish tartibida chiqaruvchi dastur tuzilsin (array).

5. Butun musbat sonning raqamlarini teskari tartibda chiqaruvchi InvertDigit nomli protsedura hosil qiling. Bu protsedura orqali a, b, c sonlarining raqamlari teskari tartibda chiqaruvchi dastur tuzilsin (proc).

Variant-42

1. Parametr-qiymatlar. Misoli bilan ko'rsating.
2. Ekran bilan ishlovchi funksiyalar. Misoli bilan ko'rsating.
3. Fibonachchi soni bo'lgan N butun soni berilgan ($N > 1$). N soni Fibonachchi ketma – ketligining nechanchi xadi ekanini chiqaruvchi dastur tuzilsin (while).

4. n ta elementdan tashkil topgan massiv berilgan. Uning elementlari orasidan juftlarini indeksleri kamayish tartibida chiqaruvchi va ularning sonini chiqaruvchi dastur tuzilsin (array).

5. Kiritilgan K butun musbat sonining o'ng tarafiga (oxiriga) R raqamini ($1 \leq R \leq 9$) qo'shuvchi AddRightDigit nomli protsedura hosil qiling (proc).

Variant-43

1. Parametr-o'zgaruvchilar. Misoli bilan ko'rsating.
2. Ovozni boshqaruvchi protseduralar. Misoli bilan ko'rsating.
3. Fibonachchi soni bo'lgan N butun soni berilgan ($N > 1$). N sonidan bir oldingi va bir keying Fibonachchi sonlarini chiqaruvchi dastur tuzilsin (while).

4. n ta elementdan tashkil topgan massiv berilgan. Uning elementlari orasidan toqlarini indeksleri o'sish tartibida chiqaruvchi va ularning sonini chiqaruvchi dastur tuzilsin (array).

5. Haqiqiy sonning ishorasini aniqlovchi ishora nomli funksiya hosil qiling. Funksiya argumenti noldan kichik bo'lsa -1; noldan katta bo'lsa 1; nolga teng bo'lsa 0 qiymat qaytarsin. Haqiqiy a va b sonlari uchun ishora(a) + ishora(b) ifodasi hisoblansin (func).

Variant-44

1. Parametr-konstantalar. Misoli bilan ko'rsating.
2. Grafik rejim qanday o'rnatiladi? Misoli bilan ko'rsating.
3. N butun soni berilgan ($N > 1$). N sonidan kata bo'lgan birinchi Fibonachchi sonini aniqlovchi dastur tuzilsin (while).

4. n ta elementdan tashkil topgan massiv berilgan. Uning elementlari teskari tartibda chiqaruvchi dastur tuzilsin (array).

5. Kvadrat tenglamaning ildizlar sonini aniqlovchi funksiya hosil qiling. $A \cdot x^2 + B \cdot x + C = 0$ ko'rinishdagi tenglama kvadrat tenglama deyiladi (A noldan farqli son) (func).

Variant-45

1. Grafika bilan ishlovchi fayl drayverlari qanday kengaytmali bo'lad? Misoli bilan ko'rsating.
2. Tipsiz parametrlar. Misoli bilan ko'rsating.
3. N butun soni berilgan ($N > 1$). N sonini Fibonachchi sonlari orasida bor – yo'qligini aniqlovchi dastur tuzilsin (while).
4. n natural soni va A, B butun sonlari berilgan ($n > 2$). $a[0] = A$; $a[1] = B$; boshqa elementlari o'zidan oldingi barcha elementlari yig'indisiga teng bo'lgan massivni hosil qiling va elementlarini chiqaring (array).
5. Doiraning yuzini hisoblovchi funksiya hosil qiling. Bu funksiya yordamida 3 ta doira yuzini hisoblang. Doiraning yuzi $S = \pi R^2$ orqali hisoblanadi. $\pi = 3.1415$ ni o'zgarmas deb qabul qiling (func).

Variant-46

1. Massivlar va ochiq tipdagi satrlar. Misoli bilan ko'rsating.
2. Qachon drayver joriy katalogdan izlaniladi? Misoli bilan ko'rsating.
3. A va B butun musbat sonlari berilgan. Berilgan sonlarning eng katta umumiy bo'luvchisini aniqlovchi dastur tuzilsin (while).
4. n natural soni berilgan. Dastlabki n ta Fibonachchi sonlaridan tashkil topgan massivni hosil qiling va elementlarini chiqaring (array). $F_0 = 1$; $F_1 = 1$; $F[k] = F[k-1] + F[k-2]$; $k = 2, 3, 4, \dots$
5. Markazi bir nuqtada bo'lgan, R1 va R2 radiusga ega 2 ta aylananing ustma-ust tushmaydigan (kesishmaydigan) qismining yuzasini topuvchi RingS nomli funksiya hosil qiling. Doiraning yuzini hisoblash formulasidan foydalaning. $S = \pi R^2$. $\pi = 3.1415$ ni o'zgarmas deb qabul qiling (func).

Variant-47

1. GraphResult funksiyasini tushuntirib bering? Misoli bilan ko'rsating.
2. Parametr – protseduralar va parametr – funksiyalar. Misoli bilan ko'rsating.
3. N butun soni berilgan ($N > 0$). N sonini tub yoki tub emasligini aniqlovchi dastur tuzilsin (while).
4. n natural soni va geometrik progressiyaning dastlabki hadi A va maxraji D berilgan. Geometrik progressiyaning dastlabki n ta hadidan tashkil topgan massivni hosil qiling va elementlarini chiqaring (array). $A_i = A_{i-1} * D$
5. To'g'ri burchakli uchburchakning katetlari A va B berilganda, uning perimetrini hisoblovchi TriangleP nomli funksiya hosil qiling (func).

Variant-48

1. Qismdasturlar buyruqlari. Misoli bilan ko'rsating.
2. Grafika bilan ishlovchi protsedura va funksiyalarni sanab bering? Misoli bilan ko'rsating.
3. N butun soni berilgan ($N > 0$). Bo'lib butun va qoldiq qismlarini aniqlash orqali, berilgan son raqamlarining orasida toq raqamlar bor – yo'qligini aniqlovchi dastur tuzilsin (while).
4. n natural soni va arifmetik progressiyaning dastlabki hadi A va ayirmasi D berilgan. Arifmetik progressiyaning dastlabki n ta hadidan tashkil topgan massivni hosil qiling va elementlarini chiqaring (array). $A_i = A_{i-1} + D$
5. A va B sonlari orasidagi sonlar yig'indisini hisoblovchi SumRange(A,B) nomli funksiya hosil qiling. Agar $A > B$ bo'lsa, funksiya 0 qiymat qaytaradi. Bu funksiya orqali A dan B gacha va B dan C gacha bo'lgan sonlar yig'indisini hisoblang. A, B, C butun sonlar (func).

Variant-49

1. Chiziqning joriy rangi nima? Misoli bilan ko'rsating.
2. Formal parametrlar nomlari faktik parametrlar nomi bilan ustma-ust tushishi mumkinmi? Misoli bilan ko'rsating.
3. N butun soni berilgan ($N > 0$). Bo'lib butun va qoldiq qismlarini aniqlash orqali, berilgan son raqamlarining orasida 2 raqami bor – yo'qligini aniqlovchi dastur tuzilsin (while).

4. n natural soni berilgan. 2 sonining dastlabki n ta darajasidan tashkil topgan massivni hosil qiling va elementlarini chiqaring (array). (1, 2, 4, 8,)

5. Arifmetik amallarni bajaruvchi Calc(A, B, Op) funksiyasini hosil qiling. A va B haqiqiy sonlar. Op o'zgaruvchi orqali bajariladigan arifmetik amal aniqlanadi. 1-ayirish, 2-ko'paytirish, 3-bo'lish, boshqalari qo'shish. Shu funksiya orqali A va B sonlari uchun N1, N2, N3, N4 amallari bajarilsin (N1-N4 butun sonlar) (func).

Variant-50

1. Grafik rejimda matnlar bilan ishlovchi protseduralar?
 2. Formal parametrni ta'riflashning qanday usullari sizga ma'lum? Misoli bilan ko'rsating.
 3. N butun soni berilgan ($N > 0$). Bo'lib butun va qoldiq qismlarini aniqlash orqali, berilgan son raqamlarini yig'indisi va raqamlari sonini chiqaruvchi dastur tuzilsin (while).

4. n natural soni berilgan. Dastlabki n ta toq sondan tashkil topgan massivni hosil qiling va elementlarini chiqaring (array).

5. X va Y butun sonlari berilgan (X va Y noldan farqli). (X , Y) nuqta qaysi chorakda ekanini aniqlovchi Quarter nomli funksiya hosil qiling. Bu funksiya orqali 4 ta nuqtaning choragini aniqlang (func).

Baholash mezon

Baholashni 5 baholik shkaladan 100 ballik shkalaga o'tkazish

JADVALI

5 baholik shkala	100 ballik shkala	5 baholik shkala	100 ballik shkala	5 baholik shkala	100 ballik shkala
5,00 — 4,96	100	4,30 — 4,26	86	3,60 — 3,56	72
4,95 — 4,91	99	4,25 — 4,21	85	3,55 — 3,51	71
4,90 — 4,86	98	4,20 — 4,16	84	3,50 — 3,46	70
4,85 — 4,81	97	4,15 — 4,11	83	3,45 — 3,41	69
4,80 — 4,76	96	4,10 — 4,06	82	3,40 — 3,36	68
4,75 — 4,71	95	4,05 — 4,01	81	3,35 — 3,31	67
4,70 — 4,66	94	4,00 — 3,96	80	3,30 — 3,26	66
4,65 — 4,61	93	3,95 — 3,91	79	3,25 — 3,21	65
4,60 — 4,56	92	3,90 — 3,86	78	3,20 — 3,16	64
4,55 — 4,51	91	3,85 — 3,81	77	3,15 — 3,11	63
4,50 — 4,46	90	3,80 — 3,76	76	3,10 — 3,06	62
4,45 — 4,41	89	3,75 — 3,71	75	3,05 — 3,01	61
4,40 — 4,36	88	3,70 — 3,66	74	3,00	60
4,35 — 4,31	87	3,65 — 3,61	73	3,0 dan kam	60 dan kam

Samarqand davlat universiteti
Amaliy matematika va informatika fakulteti
«Axborotlashtirish texnologiyalari» kafedrası

Ro'yxatga olindi:
№ _____
“ ____ ” _____ 2019 yil

Samarqand davlat universiteti
rektori:
_____ R.I.Xalmuradov
2019 yil “ ____ ” _____

ALGORITMLAR VA MA'LUMOTLAR STRUKTURASI
FAN DASTURI

Bilim sohasi: 100000 – Gumanitar
Ta'lim sohasi: 110000 – Pedagogika
Ta'lim yo'nalishi: 5110700 – Informatika o'qitish metodikasi

Samarqand – 2019

Fan dasturi Samarqand davlat universiteti amaliy matematika va informatika fakulteti kengashida korib chiqilgan va tavsiya qilingan (2019 yil “___” _____dagi “___” -sonli bayonnoma).

Fakultet dekani:

dots.A.Babayarov

Fan dasturi Samarqand davlat universitetida ishlab chiqildi.

Tuzuvchilar:

- | | | |
|--------------|---|---|
| Qobilov S.S. | - | SamDU “Axborotlashtirish gtexnologiyalari” kafedrası dotsenti. |
| Inatov A. | | SamDU “Axborotlashtirish gtexnologiyalari” kafedrası assistenti |

Taqrizchilar:

- | | | |
|--------------|---|---|
| Mahmudov Z. | - | TATU Samarqand filiali “Informatika” kafedrası dotsenti |
| Safarov T.S. | - | Samarqand tibbiyot instituti “Informatika” kafedrası professori |

Fanning dasturi Samarqand davlat universiteti oquv-uslubiy kengashining 2019 yil “___” _____dagi “___” - son majlis bayoni bilan maqullangan.

O‘quv uslubiy Kengash raisi:

prof.A.S.Soleev

I. O'quv fanning dolzarbligi va oliy kasbiy ta'limdagi o'rni

Ta'lim sohasidagi tub islohatlarning asosiy maqsadi jahon andozalari asosida bilimlar berish va raqobatlash kadrlar tayyorlashdir. SHuning uchun ta'lim tizimidagi 5110700 – Informatika o'qitish metodikasi yo'nalishida o'qitiladigan fanlar ham zamonaviy fanlardan hisoblanadi. Ushbu namunaviy dastur bugungi kunning zamonaviy bilimlari bilan yangilangan va qayta ishlangan dastur bo'lib, unda fanning nazariy va amaliy jihatlariga alohida e'tibor qaratilgan. Mazkur fan dasturi bakalavriat yo'nalishi: 5110700 – Informatika o'qitish metodikasi ta'lim yo'nalishida taxsil olayotgan talabalarning o'zlashtirishi lozim bo'lgan bilimlari va unga qo'yiladigan talablar asosida tuzilgan bo'lib, bo'lajak fan o'qituvchisi egallashi kerak bo'lgan bilimlar va ko'nikmalar majmuini o'z ichiga oladi:

- algoritmik tillar, algoritm va uning xossalari, algoritmni tasvirlash usullari, rekursiya va iteratsiya, algoritmning murakkabligi tushunchasi, algoritm turlari, samarali algoritmlar ishlab chiqishning asosiy usullari;

- steklar, navbatlar, daraxtlar, algoritmning tahlili;

- algoritmni saralash usullari, sodda saralash, tanlash usulida saralash, qo'shib saralash, almashish usulida saralash, saralashning Sheyker, Shella usullari, piramida usulida saralash, turnir usulida saralash;

- qidiruv usullari: binar qidiruv, Fibonachchi qidiruv, binar daraxt bo'yicha qidiruv, muvozanatlashtirilgan daraxt bo'yicha qidiruv, Rabin algoritmi, Boyer-Mur algoritmi, rekursiv algoritmlar;

- Paskal dasturlash tilli, dasturlash tilining alifbosi, kattaliklar va ularning tiplari, buruqlar tizimi va operatorlari, massivlar, chiziqli, tarmoqlanuvchi va takrorlanuvchi operatorlar, funktsiya va protseduralar, fayllar bilan ishlash, tilning grafik imkoniyatlari;

«Algoritmlar» fani insonlarda axborot muhitida ma'lum bir dunyoqarashni shakllantirishga xizmat qilishi bilan bir qatorda, uning axborot madaniyatni egallashida asosiy rol o'ynaydi. Bugungi «Axborot» asrida yoshlarning kompyuter savodxonligini oshiribgina qolmay, balki ma'lumotlar bazalari bilan ishlash imkoniyatlarini oshiradi. Umumiy o'rta ta'lim maktablari, akademik litsey va kasb – hunar kollejlarda «Informatika» mutahassislaridagi fanlarni o'qitish uchun kadrlarni tayyorlab beradi.

II. O'quv fanining maqsadi va vazifasi

Fanni o'qitishdan maqsad - bo'lajak informatika o'qituvchisining kasbiy sohasida egallashi lozim bo'lgan bilimlar va amalda qo'llash uchun ko'nikma va makalalarni shakllantirish va rivojlantirishdan iborat. Ushbu dasturda har bir kasb egasi uning faoliyat ko'rsatish turidan qat'iy nazar egallashi kerak bo'lgan tayanch nazariy va amaliy ma'lumotlarni o'z ichiga oladi.

Fanning o'qitishning vazifasi - algoritm tushunchasi va uning xossalari, algoritmni tasvirlash usullari, rekursiya va iteratsiya, algoritmning murakkabligi tushunchasi, algoritm turlari, samarali algoritmlar ishlab chiqishning asosiy usullari, algoritmik tillar bilan tanishtirish;

- algoritmik tillarning asosiy tushunchalari: steklar, navbatlar, daraxtlar, algoritm tahlili kabilar haqida ma'lumotlar berish.

- saralash algoritmni va ulardan foydalanish usullari haqida tasavvurlar hosil qilish.

borasidagi bilimlar va amalda qo'llash uchun ko'nikma va makalalarni shakllantirish va rivojlantirishdan iborat.

Fan bo'yicha talabalarning bilim, ko'nikma va malakalariga quydagi talablar qo'yiladi: Algoritmlar o'quv fanini o'zlashtirish jarayonida amalga oshiriladigan masalalar doirasida bakalavr:

- algoritm va uning xossalari, algoritmik tillar, steklar, navbatlar, daraxtlar, algoritmlar tahlili;

- algoritmlarni saralash usullari, sodda saralash, tanlash usulida saralash, qo‘shib saralash, almashish usulida saralash, Sheyker usulida saralash, Shell usulida saralash, piramida usulida saralash, turnir usulida saralash;

- qidiruv usullari: binar qidiruv, Fibonachchi qidiruv, binar daraxt bo‘yicha qidiruv, muvozanatlashtirilgan daraxt bo‘yicha qidiruv, Rabin algoritmi, Boyer-Mur algoritmi, rekursiv algoritmlar;

- Paskal dasturlash tilli, dasturlash tilining alifbosi, buruqlar tizimi va operatorlari, chiziqli, tarmoqlanuvchi va takrorlanuvchi algoritmlarga doir ***to‘g‘risida tasavvurga ega bo‘lishi***;

- algoritmlar, samarali algoritmlar ishlab chiqishning asosiy usullari, algoritmik tillar, steklar, navbatlar, daraxtlar, algoritmlar tahlili;

- algoritmlarni saralash usullari, sodda saralash, tanlash usulida saralash, qo‘shib saralash, almashish usulida saralash, Sheyker usulida saralash, Shell usulida saralash, piramida usulida saralash, turnir usulida saralash;

- qidiruv usullari: binar qidiruv, Fibonachchi qidiruv, binar daraxt bo‘yicha qidiruv, muvozanatlashtirilgan daraxt bo‘yicha qidiruv, Rabin algoritmi, Boyer-Mur algoritmi, rekursiv algoritmlar;

- Paskal dasturlash tilli, dasturlash tilining alifbosi, buruqlar tizimi va operatorlari, chiziqli, tarmoqlanuvchi va takrorlanuvchi algoritmlarni ***bilishi va ulardan foydalana olishi***;

-algoritmik tillarning asosiy tushunchalaridan foydalanib sodda va tadbqiqiy masalalar yechish ***ko‘nikmalariga ega bo‘lishi lozim***.

III. Asosiy nazariy qism (ma’ruza mashg‘ulotlari)

1-Modul. Algoritmlar tushunchasi va mohiyati

1-mavzu. Algoritm tushunchasi va ulardan foydalanish

Algoritm tushunchasi va ular haqida ma’lumotlar. Algoritmlar tarixi. Algoritm xossalari, turlari va uning berilish usullari.

2-mavzu. Algoritmlar samaradorligini baholash

Xotiraviy samara, vaqt samarasi. Algoritmlarning murakkablik darajasi. Algoritmlarning taqqoslash usullari.

3-mavzu. Tanlash va joylashtirish turkumidagi murakkablikga ega saralash algoritmlari

Saralash tushunchasi. Saralash algoritmlari. Tanlash va joylashtirish usulida saralash, O‘shib borish va kamayish tartibida saralash, qo‘shish usulida saralash, Joyida abstrakt qo‘shib saralash, Yuqoridan pastga qo‘shib saralash,

4-mavzu. Almashish usulida saralash, saralashning Sheyker usuli

Almashish usuli mohiyati, saralashning Sheyker, pufakcha va piramida usullari.

5-mavzu. Algoritmlar tahlili

Algoritmlarning baholash va ularning tahlili. Kiruvchi berilganlar sinfi. Xotira bo‘yicha murakkablik. Tahlil va maqsad.

6-mavzu. Algoritmlarni ishlab chiqish metodlari

Algoritmlarni ishlab chiqishning o‘sga xos jihatlari. Strukturaviy algoritmlar, Prosedurali algoritmlar metodi, Algoritmlar konstruksiyasi. Algoritmlar ekvivalent qayta ishlash. Toraytiruvchi o‘zgartirishlar. Formal usulni matematikaga bog‘liq bo‘lmagan muammoga qo‘llash.

7-mavzu. Rekursiya va rekursiv funksiyalar

Rekursiya tushunchasi. Funksiya va rekursiv funksiyalar. Rekursiv funksiyalar turlari. Oddiy va vositali rekursiya. Matematikada rekursiyning qo‘llanilishi. Fibonachchi funksiyasi.

8-mavzu. Qidiruv usullari: binar qidiruv, Fibonachchi qidiruv, binar daraxt bo‘yicha qidiruv

Qidiruv masalasi, Qidiruv usullari. Yozuvlarni oddiy ko'rib chiqish usuli, Ketma-ketlik usulida qidiruv, Binar qidiruv, Interpolyatsiya usulida qidiruv, Binar daraxt va Fibonachchi qidiruvlar, Muvozanatlashgan (Balansirlangan) daraxt bo'yicha qidiruv, Bor usulida qidiruv, h-Hashlash usulida qidiruv, Interval bo'yicha izlash.

9-mavzu. Rabin-Karp algoritmi

Rabin-Karp algoritmi haqida. Qidiruv algoritmlari. Rabin-Karp algoritmi xesh-funksiya. Algoritm ahamiyati.

10-mavzu. Graflar bilan ishlovchi sodda algoritmlar

Graflar nazariyasi, Graflarni tipik qo'llanilishi, Graflar terminologiyasi, Qism graf, orentirlangan va orentirlanmagan graflar, Graflar izomorfizmi.

11-mavzu. Paskal tili dasturlash tilining alifbosi, buyruqlar tizimi va operatorlari

Paskal tili dasturlash tilining alifbosi, Pascal dasturida asosiy fayllar va funksional tugmalar, O'zgarmas va o'zgaruvchi miqdorlar, Standart funksiyalar va algebraik ifodalar, O'zlashtirish va ma'lumotlarni ekranga chiqarish operatorlari, Ekran bilan ishlash operatorlari.

12-mavzu. Chiziqli, tarmoqlanuvchi va takrorlanuvchi dasturlar

Pascalda chiziqli, tarmoqlanuvchi va takrorlanuvchi dasturlar haqida ma'lumotlar, qiymat berish, tarmoqlanuvchi operatorlar, shartli va shartsiz o'tish operatorlari, siklik operatorlar.

2-Modul. Paskal dasturlash tilining imkoniyatlari

13-mavzu. Paskalda massivlar

Massivlar haqida tushuncha, bir o'lchovli va ikki o'lchovli massivlar, dinamik massivlar.

14-mavzu. Paskalda satriy kattaliklar bilan ishlash

Satriy kattaliklar, satriy kattaliklar bilan ishlovchi funksiya va proseduralar, satriy kattaliklardan foydalanishning o'ziga xos xususiyatlari.

15-mavzu. Qism dasturlar-funksiyalar

Qism dastur, qism dasturlar-funksiyalar, qismdastur-funksiya tuzilishi, formal, lokal va haqiqiy parametrlar, dasturda qism dasturlardan foydalanish.

16-mavzu. Qism dasturlar-proseduralar

Qism dastur, qism dasturlar-proseduralar, qismdastur-proseduralar tuzilishi, formal, lokal va haqiqiy parametrlar, dasturda qism dasturlardan foydalanish.

17-mavzu. Paskalda fayllar bilan ishlash

Paskalda fayllar bilan ishlash, tiplashtirilgan va tiplashtirilmagan fayllar, fayllar bilan ishlovchi standart funksiya va proseduralar, fayllarga yozish, fayllardan o'qish.

18-mavzu. Grafika moduli va ular bilan ishlash

Paskal dasturlash tilining grafik imkoniyatlari, grafik adapterlar, graph moduli funksiya va proseduralari.

19-mavzu. Paskal tilida sodda shakllar chizish

Paskalda sodda shakllar chizuvchi proseduralar, sodda shakllar chizishning o'ziga xos xususiyatlari.

IV. Amaliy mashg'ulotlar bo'yicha ko'rsatma va tavsiyalar

Fan bo'yicha rejalashtirilgan amaliy mashg'ulotlar davomida nazariy bilimlar mustahkamlanadi. Amaliy mashg'ulotlarda tadbiqlar keng qo'llanilayotgan asosiy matematik usullarni o'rgatishga e'tiborni kuchaytirish lozim. Ma'lum sababalariga ko'ra ma'ruzaga kirmagan va murakkab bo'lgan tushunchalarni amaliy mashg'ulotlarda ko'rib o'tish maqsadga muvofiqdir.

Amaliy mashg'ulotlarni o'tkazishda quyidagi didaktik taomillarga amal qilinadi:

amaliy mashg'ulotlarining maqsadini aniq belgilab olish;

o'qituvchining innovatsion pedagogik faoliyati bo'yicha bilimlarni chuqurlashtirish imkoniyatlariga talabalarda qiziqish uyg'otish;

talabada natijani mustaqil ravishda qo‘lga kiritish imkoniyatni ta’minlash;
talabani nazariy-metodik jihatdan tayyorlash;

amaliy mashg‘ulotlarda nafaqat aniq mavzu bo‘yicha bilimlarni yakunlash, balki talabalarni tarbiyalash manbai hamdir. Amaliy mashg‘ulotlarda talabalar muayyan masala bo‘yicha mavjud bo‘lgan yoki mustaqil tarzda kichik ishchi guruhlar yordamida hosil qilingan algoritmlarni muhokama qiladilar. Mazkur mavzularga oid test masalalar tuzib, ular asosida tuzilgan dasturlar majmuasini tuzadilar va kompyuterda olingan natijalarni birgalikda tahlil qiladilar. Amaliy mashg‘ulotlarning taxminiy tavsiya etiladigan mavzulari:

1. Algoritm tushunchasi va ulardan foydalanish
2. Algoritmlar samaradorligini baholash
3. Tanlash va joylashtirish turkumidagi murrakkablikga ega saralash algoritmlari.
4. Almashish usulida saralash, saralashning Sheyker usuli
5. Algoritmlar tahlili
6. Algoritmlarni ishlab chiqish metodlari
7. Rekursiya va rekursiv funksiyalar
8. Qidiruv usullari: binar qidiruv, Fibonachchi qidiruv, binar daraxt bo‘yicha qidiruv
9. Rabin-Karp algoritmi
10. Graflar bilan ishlovchi sodda algoritmlar
11. Paskal tili dasturlash tilining alifbosi, buruqlar tizimi va operatorlari.
12. Chiziqli, tarmoqlanuvchi va takrorlanuvchi dasturlar
13. Paskalda massivlar
14. Paskalda satriy kattaliklar bilan ishlash
15. Qism dasturlar-funksiyalar
16. Qism dasturlar-protseduralar
17. Qism dasturlar-funksiya va proseduralarga doir dasturlar tuzish
18. Paskalda fayllar bilan ishlash.
19. Grafika moduli va ular bilan ishlash.
20. Paskal tilida sodda shakllar chizish

Amaliy mashg‘ulotlarni tashkil etish bo‘yicha kafedra professor-o‘qituvchilari tomonidan ko‘rsatma va tavsiyalar ishlab chiqiladi. Unda talabalar asosiy ma’ruza mavzulari bo‘yicha olgan bilim va ko‘nikmalarini amaliy masalalarga dasturlar tuzish orqali bilimlarini yanada boyitadilar. Shuningdek, darslik va o‘quv qo‘llanmalar asosida talabalar bilimlarini mustaxkamlashga erishish, tarqatma materiallardan foydalanish, ilmiy maqolalar va tezislarni chop etish orqali talabalar bilimini oshirish, masalalarning dasturini tuzish, mavzular bo‘yicha ko‘rgazmali qurollar tayyorlash va boshqalar tavsiya etiladi.

V. Laboratoriya mashg‘ulotlar bo‘yicha ko‘rsatma va tavsiyalar

Laboratoriya mashg‘ulotlarida talabalar kompyuter yordamida grafik ma’lumotlarni hosil qilish va amalda uning natijalarini ko‘rib, ularni tahlil qiladi va xulosalar chiqaradilar. Laboratoriya mashg‘ulotlari uchun tavsiya etiladigan mavzular:

1. Algoritm tushunchasi va ulardan foydalanish
2. Algoritmlar samaradorligini baholash
3. Tanlash va joylashtirish turkumidagi murrakkablikga ega saralash algoritmlari.
4. Almashish usulida saralash, saralashning Sheyker usuli
5. Algoritmlar tahlili
6. Algoritmlarni ishlab chiqish metodlari
7. Rekursiya va rekursiv funksiyalar
8. Qidiruv usullari: binar qidiruv, Fibonachchi qidiruv, binar daraxt bo‘yicha qidiruv
9. Rabin-Karp algoritmi
10. Graflar bilan ishlovchi sodda algoritmlar
11. Paskal tili dasturlash tilining alifbosi, buruqlar tizimi va operatorlari.

12. Chiziqli, tarmoqlanuvchi va takrorlanuvchi dasturlar
13. Paskalda massivlar
14. Paskalda satriy kattaliklar bilan ishlash.
15. Qism dasturlar-funksiyalar
16. Qism dasturlar-protseduralar
17. Paskalda fayllar bilan ishlash.
18. Grafika moduli va ular bilan ishlash.
19. Paskal tilida sodda shakllar chizish

VI. Mustaqil ta'lim va mustaqil ishlar

Mustaqil ta'lim uchun tavsiya etiladigan mavzulari:

1. Algoritm tushunchasi va ulardan foydalanish
2. Algoritm va uning xossalari
3. Algoritm turlari
4. Algoritm nazariyasiga asos solgan olimlar
5. Tarmoqlanuvchi algoritmlar
6. Takrorlanuvchi algoritmlar
7. Algoritmlar samaradorligini baholash
8. Saralash usullari
9. Pufakcha usulida saralash
10. Tanlash va joylashtirish turkumidagi murakkablikga ega saralash algoritmlari.
11. Qo'shish usulida saralash
12. Piramida usida saralash
13. Almashish usulida saralash, saralashning Sheyker usuli
14. Algoritmlar tahlili
15. Eyler sikllari va zanjirlari. Eyler teoremasi.
16. Algoritmlarni ishlab chiqish metodlari
17. Evklid algoritmi
18. Rekursiya va rekursiv funksiyalar
19. Qidiruv usullari: binar qidiruv
20. Fibonachchi qidiruv
21. Binar daraxt bo'yicha qidiruv
22. Rabin-Karp algoritmi
23. Graflar bilan ishlovchi sodda algoritmlar
24. Paskal tili dasturlash tilining alifbosi, buruqlar tizimi va operatorlari.
25. Standart funksiyalar bilan ishlash
26. Tanlash operatorlari bilan ishlash
27. Chiziqli va tarmoqlanuvchi dasturlar
28. Takrorlanuvchi dasturlar
29. Paskalda bir o'lovli massivlar.
30. Paskalda ikki o'lovli massivlar.
31. Paskalda satriy kattaliklar bilan ishlash.
32. Qism dasturlar-funksiyalar yordamida dasturlar tuzish
33. Qism dasturlar-protseduralar yordamida dasturlar tuzish
34. Paskalda fayllar bilan ishlash.
35. Ma'lumotlarni fayllardan o'qish.
36. Ma'lumotlarni fayllarga yozish.
37. Grafika moduli va ular bilan ishlash.
38. Paskal tilida sodda geometrik shakllar chizish
39. Chiziqli ro'yhatlar
40. Tarmoqlar. Daraxtlar.
41. Izlash algoritmlari.

42. Rekursiv funksiyalar.
43. Interpretatorlar va kompilyatorlar
44. Tanlash operatori bilan ishlash
45. Grahf moduli bilan ishlash.
46. Paskalda sodda shakll chizish
47. Paskalda kichik hajmdagi amaliy dasturlar yaratish
48. Paskalda amaliy dasturlar yaratish

VII. Asosiy va qo'shimcha o'quv adabiyotlar hamda axborot manbalari

Asosiy adabiyotlar:

1. M.Aripov, M.Muhammadiyev. Informatika, informasion texnologiyalar. Darslik. T.: TDYuI, 2005 y.
2. A. R. Azamatov, B. Boltayev. Algoritmash va dasturlash asoslari. O'quv qo'llanma. T. : "Cho'lpon", 2013 y.
3. Sh. I. Razzoqov, M. J. Yunusova. Dasturlash: Kasb-hunar kollejlari uchun o'quv qo'llanma. T. : "Ilm Ziyo", 2011y.
4. T. X. Holmatov, N. I. Tayloqov. Amaliy matematika, dasturlash va kompyuterning dasturiy ta'minoti. O'quv qo'llanma. T.: "Mehnat", 2000 y.
5. Sattorov A. Informatika va axborot texnologiyalari. Darslik. T. : , "O'qituvchi", 2011 y.
6. B.Mo'minov. Informatika. O'quv qo'llanma. T.: "Tafakkur-bo'stoni", 2014 y.

Qo'shimcha adabiyotlar

7. Мирзиёев Шавкат Миромонович. Эркин ва фаровон, демократик Ўзбекистон давлатини биргаликда барпо этамиз. Ўзбекистон Республикаси Президенти лавозимида киришиш тантанали маросимида бағишланган Олий Мажлис палаталарининг қўшма мажлисидаги нутқ / Ш.М. Мирзиёев. – Тошкент : Ўзбекистон, 2016. - 56 б.
8. Мирзиёев Шавкат Миромонович. Танқидий таҳлил, қатъий тартиб-интизом ва шахсий жавобгарлик – ҳар бир раҳбар фаолиятининг кундалик қондаси бўлиши керак. Мамлакатимизни 2016 йилда ижтимоий-иқтисодий ривожлантиришнинг асосий яқунлари ва 2017 йилга мўлжалланган иқтисодий дастурнинг энг муҳим устувор йўналишларига бағишланган Вазирлар Маҳкамасининг кенгайтирилган мажлисидаги маъруза, 2017 йил 14 январ / Ш.М. Мирзиёев. – Тошкент : Ўзбекистон, 2017. – 104 б.
9. Мирзиёев Шавкат Миромонович. Қонун устуворлиги ва инсон манфаатларини таъминлаш – юрт тараққиёти ва халқ фаровонлигининг гарови. Ўзбекистон Республикаси Конституцияси қабул қилинганининг 24 йиллигига бағишланган тантанали маросимдаги маъруза. 2016 йил 7 декабр /Ш.М.Мирзиёев. – Тошкент: "Ўзбекистон", 2017. – 48 б.
10. Ўзбекистон Республикаси Президентининг Фармони. Ўзбекистон республикасини янада ривожлантириш бўйича ҳаракатлар стратегияси тўғрисида. (*Ўзбекистон Республикаси қонун ҳужжатлари тўплами, 2017 й., 6-сон, 70-модда*)
11. П. Дарахвелидзе, Э. Марков. Программирование в Delphi7. Учебник. Санкт-Петербург, "БХВ-Петербург" 2003 г.
12. В. М. Пестиков, А. Н. Маслобоев. Turbo PASKAL 7. 0. Изучаем на примерах. Санкт-Петербург. : "БХВ-Петербург", 2004 г.
13. Фаронов В. В. Программирование на языке высокого уровня Delphi. Учебник. М. : "Питер", 2003 г.
14. В. Т. Безручко. Практикум по курсу информатики. М. : «Финансы и статистика», 2004 г.
15. Меняев Михаил Федорович. Информационные технология управления. Москва, «Издательский Омега», 2003 г.

16. В.Д.Колдаев. Основы алгоритмизации и программирования. Учебный пособый, Москва ИД “Форум”- ИНФРА-М 2006 г.
17. Thomas H. Cormen. Intruduction to algorithms. Third Edition. Massachusetts Institute of Technology. The MIT Press. London 2009. 1292-p.
18. Algorithms, Fourth Edition (Deluxe): Book and 24-Part Lecture Series 1st Edition , Addison-Wesley Professional , USA, 2015
19. Цой, Маргарита и др. Создание электронных учебников. Т. : “Ўзбекистон миллий энцеклопедияси”, 2007 г.

Internet saytlari

20. www.ziyonet.uz – Axborot ta’lim portali
21. www.edu.uz – Oliy va o’rta maxsus ta’lim vazirligi portali
22. www.tdpu.uz – Nizomiy nomidagi TDPU rasmiy sayti
23. [http:// corel.Deamiart.ru/](http://corel.Deamiart.ru/).
24. www.amazon.com
25. <http://www.ctc.msiu.ru/materials/Book1,2/index1.html>

XORIJIY MANBALAR

1. www.5ballow.ru, www.referat.ru – Algoritmlar va ma'lumotlar strukturasi, Informatika va dasturlash, Internet va ofis dasturlari bilan ishlashga doir referat va ma'ruzalar
2. www.download.ru – Dasturlar va drayverlarni olish va o'rnatish.
3. Scheinerman Edwand C++ for mathematicifns. AnIntroduction for Students and Frofessionals. Chapman&Hall//CRC, Taylor&Francis Group, LLC, Boca Raton, London, New York, 2006.
4. Stroustrup Djarne The C++ Programming Language (Fourt Edition). Amozon Product Page, 2012.
5. Virender Singh Learn C++ Programming Language :Become A Complete C++ Programmer, Amazon Digital Services, Inc. 2015.
6. Павловская Т.С., Шупак Ю.С. С++. Объектно-ориентированное программирование. Практикум.- СПб.: Питер, 2005 – 265 с.
7. Павловская Т.С. , Шупак Ю.С. С/С++. Структурное программирование. Практикум.- СПб.: Питер, 2002-240 с.
8. Архангельский А.Я. С++ Builder 6. Справочник пособие. Книга 1. Язык С++б –МюЖ Бином-Пресс, 2002, -544 с.
9. Культин Н.Б. С++ Builder в задачах и примерах. – СПб.:БХВ-Петербург, 2005. – 336 с.
10. Абрамов С.А., Гнезделова, Капустина Е.Н., и др. Задачи по программированию. - М.: Наука, 1988. Информатика. Базовый курс. 2-е издание. Под редакцией С.В.Симоновича. Учебник для ВУЗов. СПб.: Питер, 2004. – 640 с.: ил.
11. Павловская Т.А. С++. Программирование на языке высокого уровня – СПб.: Питер. 2005.- 461 с.
12. Глушаков С.В., Ковал А.В., Смирнов С.В. Язык программирования С++: Учебный курс.- Харьков: Фолио; М.: ООО «Издательство АСТ», 2001.-500с.
13. Б. Страуструп. Язык программирования С++. Специальное издание.-М.:ООО «Бином-Пресс», 2006.-1104 с.
14. Подбелский В.В. Язык СИ++.- М.; Финансы и статистика- 2003 562с.
15. Алексеев А.П. Информатика. 2001. М., СОЛОН-Р, 2001, 364 с.
16. Романчик В.С., Люлкин А.Е. Программирование в С++ Builder. Учебное пособие. Мн.: БГУ, 2007. –126 с..
17. Самарский А.А., Михайлов А.П. Компьютеры и жизнь. Математическое моделирование. - М., "Педагогика", 1987. -128с.
18. . Элементы Информатики / Под редакцией Л.Н. Королёва. - М., "Наука",1988. -318с.
19. Абрамов С.А., Зима Е.В. Начало информатики. - М., "Наука", 1989. -256с.
20. Задачи и упражнения по программированию./ под ред. А.Я. Савельева. - М., "Высшая школа", 1989. -110с.
21. Языки программирования Ада, Си. Паскаль. Сравнительный анализ и оценка. / Под редакцией А.Р.Фьюэра, Н.Джехани. - М., 1989
22. Есаян А.Р. и др. Информатика. - М., "Просвещение", 1991. -288с.
23. Касаткин В.Н. Информация. Алгоритмы. ЭВМ. - М., "Просвещение", 1991.

Atamalar

Abonent tizimi - foydalanuvchilarga axborot tarmoq'larining xizmatlarini ta'minlovchi axborot tizim.

Fayl adresi - faylning to'la nomi bo'lib, u fayl joylashgan diskning mantiq'iy nomi, unga yo'l va nomi orqamda kengaytmasini ko'rsatadi. *Masalan:* S: \kafedra\kitob\bob1.txt.

Avtomatlashtirilgan loyihalash tizimi - loyihalash jarayonini avtomatlashtirishga mo'ljallangan programmalar va q'urilmalar majmui.

Axborot tarmoq'lari arxitekturas - axboriy tizimlar birikmasini orqasil q'ilish san'ati.

Alifbo-raq'amli axborot-bitta tugmachani bosish bilan kompyuterga kiritiladigan belgilar majmui (raqamlar, raq'amlar, munosabat belgilari va b.).

Arifmetik ifoda-q'o'shish, ayirish, ko'paytirish, bo'lish, darajaga ko'tarish(^), ochuvchi (va yopuvchi) q'avslar yordamida yozilgan son, funksiya, indeksli va indeksiz o'zgauvchilar ketma-ketligidan iborat yozuv. Arifmetik ifoda natijasi son bo'ladi.

Bayt - kompyuter tomonidan yaxlit birlik sifatida q'abul q'ilinadigan 8 bitdan iborat axborot. U turli belgilarni kompyuterda tasvirlash, ularni bir-biridan farq'lash imko-niyatini beradi.

Bit - axborotning eng kichik biriligi bo'lib, u ikkilik sanoq' sistemasidagi raq'am. Bitning q'iymati 0 yoki 1 ga teng.

Almashuv buferi(Clipboard) - **Windows** operasion tizimi ishlash jarayonida ajratiladigan xotira sohasi. Ilova va orqaj-jatlar o'rtasida grafik va matnli axborotlarni almashishga mo'ljallangan xotira.

Bog'langan fayllar -birinning bosh kaliti-ikkinchisi bosh kalitining katta q'ismini tashkil q'iluvchi ikki ma'lumotlar fayli(DBF).

Bosh kalit - q'iymati ma'lumotlar faylidagi yozuvni bir q'iymatli aniq'lashga imkon beradigan **DBF** fayl yozuvlaridan iborat maydonlar majmui.

Axborotlar tuzilmasi - ma'lumotlarni tasvirlash va tashkil q'ilish to'larisidagi kelishuv.

Boshq'aruvchi belgi-kodlar jadvalidagi 32 dan kichik o'nli kodga mos keluvchi belgilar. Ulardan kompyuter q'urilmalarini boshq'arish va axborotlarni uzatish uchun foydalaniladi.

Videodisk- metall yoki plastmassadan ishlangan tekis disk.Uning satxiga lazer nuri yordamida o'q'iladigan axborot yoziladi.

Videoaxborot- turli tasvirlar(fotogarfiya, rasm va boshq'alar) yordamida uzatiladigan axborotlar.

Axboriy tizim-q'andaydir usulda tuzilgan ma'lumotlar, ularni saq'lash va q'ayta ishlashga mo'ljallangan q'urilma(programmalar majmui).

Axboriy resuslar- ilmiy nazariya,jarayon va orqadisalarni o'rganish borasidagi tadq'iq'otlar, kashfiyot, bino va mashinalar loyihalari, jamiyat va tabiat orqaidagi xabarlariga asoslangan davlatning ma'naviy potensiali.

Axboriy bank-axborotlarni saq'lash va q'ayta ishlashga imkon beruvchi axboriy, texnik, programmaviy, til va tashkiliy vositalar majmui.

Buyruq' - axboriy tizim tomonidan axborotni q'ayta ishlash jarayoniga bo'lik aniq' amal bajarilishini talab q'iluvchi son, so'z yoki jumla.

Bosish q'urilmasi (printer)- axborotni q'o'ozda akslantiruvchi q'urilma.

Amaliy programma - foydalanuvchi topshiri'ini bajaruvchi programma.

Amaliy jarayon-foydalanuvchi topshiri'i bo'yicha baja-riladigan axborotni q'ayta ishlash jarayoni.

Bayonnoma (protokol)- axboriy tizimlarni o'zaro faoliyatini ta'minlovchi masalalardan biri bajarilishini ifodalovchi q'oida berilgan hujjat.

Aloq'a tuguni - tizimlar o'rtasida uzatiladigan axborotlar aloq'asini ta'minlovchi apparat va programma ta'minoti.

Axborotni boshq'arish tili - xabarlarni izlash, ularni q'ayta ishlash va axboriy bazaga kirish bilan bo'lik amallarni aniq'lashga mo'ljallangan axborot izlash tili.

Boshq'arish bayonnomasi (tarmoq'lararo bayonnoma)- ma'lumotlararo kompyuter tarmoq'larida ma'lumotlarni almashishni ta'minlovchi kelishuv maq'ida hujjat.

Yost- kliyentga biron- bir xizmat ko'rsatuvchi kompyuter.

Axborotni siq'ish - ma'lum majmdagi axborotni saq'lash uchun undagi bitlar sonini kamaytirish jarayoni.

Telekommunikasiya - axborotni uzatishga telealoq'a vositalarini q'o'llash.

Virtual voq'yelik - axborot vositalarining eng dolzarb yo'nalishlaridan biri. Turli voq'yeliklarga taq'lidni tasvirlovchi programmalar.

Multimedia (multi-ko'p, media-vosita) -axborotlarni akslantirish ko'p axboriy vositalar (ovoz, rasm, fotografiya, musiq'a va boshq'alar)dan foydalanish.

Elektron nashr – maq'ola va kitoblarni kompyuterda tashkil q'ilish va undan foydalanish tizimi.

Kliyent (Client) – tarmoq'ning q'ayeridadir joylashgan servisdan foydalanuvchi nomidan xizmat oladigan amaliy programma.

Deytagramma - q'abul q'iluvchi kompyuterga oldindan xabar q'ilmasdan uzatilgan axborotlar paketi. Bu usul uncha katta bo'lmagan axborotlarni uzatishda foydalaniladi.

Axboriy madaniyat - insonni barcha axborot texnolo-giyalaridan kerakli tarzda foydalana olishi.

Oyna - ekranni to'rtburchak ushshiya bilan chegaralangan bo'lagi.

Piktogramma (ikonka)- programma yoki programmalar gu-ruxini ifodalovchi belgi.

Paintbrush - grafik muxarrir bo'lib, u relyasion ma'lumotlar ombori; ya'ni ikki o'lchovli jadvallar shaklida tashkil q'ilingan ma'lumotlar ombori.

Interfeys (interface)- o'zaro ta'sir, aloq'a, birlashish, kelishish vositalari. Programma tuzuvchi, foydalanuvchi, kompyuter operatori interfeyslari bir-biridan farq'lanadi. Interfeysning asosiy tushunchalari: menyu va muloq'ot oynasi.

Software - kompyuterning programma ta'minoti.

Hardware - kompyuterning q'urilmalar ta'minoti.

Menyu - tanlash mumkin bo'lgan obyekt (narsa) lar ro'yhati.

Ma'lum amalni bajarishga, vazifani bajarish yo'nalishini o'zgartiruvchi, tanlash maydoni turidagi bandlardan iborat. Asosiy, gorzontal, vertikal menyular bir-biridan farq'lanadi.

Muloq'ot oynasi – aniq' ma'lumotni kiritishga so'rov beriladigan joy.

Fayllarni arxivliash - diskdagi joyni tejash maq'sadida fayllar guruxini bitta arxiv faylga joylashtirish amali.

Vinchester – q'attiq' diskni programma tuzuvchilar tomonidan nomlanishi.

Ifoda - ma'lum mulohazani yozish shakli. Ifoda operand (tashkil q'iluvchilar)dan iborat bo'ladi. Ular bir-biri bilan ifodaning ma'nosini bildiruvchi maxsus belgilar yordamida birlashtiriladi. Amaliy informatikada arifmetik, shartli va mantiq'iy ifodalar farq'lanadi.

Egiluvchan disk - axborotlarni doimiy saq'lash uchun ishlatiladigan magnitli axborot to'plagich.

Grafik muxarrir - grafikli tasvirlarni yaratish va o'zgartirish uchun foydalaniladigan programma vositalari.

Tanlash maydonlari guruxi - menyuning bitta va faq'at bittasi tanlanadigan bandlari (*Masalan*, piktogramma, muloq'ot oynasi maydonlari). Tanlangan maydon bajariladigan vazifani mumkin bo'lgan uqollaridan birini amalga oshirishni ta'minlaydi.

Ma'lumot -obyektning muayyan xususiyatini belgilovchi ko'rsatkich bo'lib, u ko'rilayotgan obyektning aniq' bir nusxasi uchun ma'lum sonli, matnli yoki boshq'a q'iymat q'abul q'iladi.

Mantiq'iy disk - q'attiq' magnitli disk xotirasining bo'lagi bo'lib, bu bo'laklar S:, D:, E: va boshq'a lotin alifbosi uqarflari bilan belgilanadi.

Disk yuritgich (diskovod) - egiluvchan magnitli diskka xizmat q'iluvchi elektron-mexanik q'urilma.

Hujjat - amaliy programma yordamida q'ayta ishlanadigan obyekt.

Drayver(driver) - amaliy programma va tashq'i q'urilma yoki xotira o'rtasida joylashgan tizim programmasi. U aniq' bir vazifani bajaradi (*Masalan*, klaviatura, "sichq'oncha", printer, monitor drayverlari).

Ma'lumotlar ombori (MO)-q'aralayotgan somadagi obyektlar uqolatini va ular o'rtasidagi munosabatlarni aniq'lovchi nomlan-gan ma'lumotlar majmuidir. Relyasion, shajarasimon va to'rsi-mon tuzilishdagi MO farq'lanadi.

Ma'lumotlar omborini boshq'arish tizimi (MOBT) MO ni yaratish, yuritish va ko'p foydalanuvchilar tomonidan birgalikda q'o'llanishga mo'ljallangan til va programma vositalarining majmui.

Ma'lumotlar ombori tizimi (MOT)- MO bilan to'ldirilgan MOBT ma'lumotlar banki (MB). Markaz-lashtirilgan yivish va jamoa bo'lib foydalanishga mo'ljallangan ma'lumotlarning MO texnologiyasiga asoslangan programma, til, tashkiliy va texnik vositalari. Ma'lumotlar tuzilishini normallashtirish - shajarasimon daraxt shoxlari orq'ali bargni mos shoxga va tugunga joylashtirish jarayoni. Bosh kalit - MO dagi ixtiyoriy yozuvni tanlab olish imkonini beruvchi yozuv bo'lib, u obyektning bir nusxasidan boshq'asini farq'lay oladi.

Takrorlanuvchi gurux - obyektning har bir nusxasiga mos holda turli majmda mos keladigan axborot.

Mantiq'iy ma'lumot - "Haqiqat" yoki "Yo'lon"(1 yoki 0) qiymatlaridan birini q'abul qiluvchi ma'lumot.

Boshlan'ich yuklash - kompyuterni tokka ulaganda ishga tayyorlash.

Ilova oynasi - amaliy programma bajarilayotgan oyna.

Piksel - videoadapter tomonidan (monitor ekranida) yaratilgan tasvirning minimal elementi (**Picture Element - pel**).

Burash mintaq'asi - mujlat yoki ro'yhat oynasining pastki yoki o'ng q'ismida joylashadi. Oynaga si'magan mujlat bo'ylab ko'chib yurishga (skroling) xizmat qiladi (ko'rsatkich va bosq'ich bilan ta'minlangan).

Programma guruxi - mujlat oynasiga ochiladigan u yoki bu ma'noda bir xil turdagi programma bilan beriladi.

Punkt - shrift o'lchami (balandligi) ning birligi bo'lib, uning qiymati 1/72 dyuymga teng(1 dyuymq' 2,54 sm).

Kengaytma - fayl nomining bir q'ismi bo'lib, faylda ifodalangan axborot turini aniq'laydi.

Matn muxarriri - matnli fayl va matnli mujlatlarni kiritish va o'zgartirishga mo'ljallangan programma vositalari.

Relyasion yondoshuv - ixtiyoriy tuzilishdagi ma'lumotlarni sodda ikki o'lchovli jadval ko'rinishda tasvirlash.

Tarmoq' grafigi - tarmoq'ning aloq'a kanallari orq'ali vaq't birligi ichida o'tuvchi axborot majmi. U muhim ko'rsatkich bo'lib tarmoq'ning q'anchalik bandligini va islatilishini ko'rsatadi.

Elektron jadvallar - jadval ma'lumotlarini q'ayta ishlash uchun mo'ljallangan programma vositalari.

Matnli fayl - o'zgaruvchan uzunlikdagi q'atorlardan iborat fayl bo'lib, har bir satri kodlar jadvalidagi ixtiyoriy belgilar majmui.

Faylni chetlatish - diskda shu fayl joylashgan q'ismni boshq'a fayllar joylashishi uchun ochiq' deb e'lon qilish.

