

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ
Ш. ЕСЕНОВ АТЫНДАҒЫ КАСПИЙ МЕМЛЕКЕТТІК ТЕХНОЛОГИЯЛАР ЖӘНЕ
ИНЖИНИРИНГ УНИВЕРСИТЕТІ

Г.Г.Бегаришева

Р.М.Байназарова

Г.М.Тасмухамбетова

Алгоритмдік тілдерде бағдарламалау

(Оқу құралы 050704-Есептеуіш техникасы және бағдарламалық қамтамасыз ету)

Ақтау, 2011

УДК 004.43(075)

ББК 32.973-018.1я7

А 39

Пікір жазғандар: д.т.н. Ускенбаева Р.К.

д.т.н. Туркпенбаева Б.Ж.

А39 Алгоритмдік тілдерде бағдарламалау: оқу құралы /Құраст.: Г.Г.Бегаришева, Р.М.Байназарова, Г.М.Тасмухамбетова - Ақтау, 2011. – 108 бет.

ISBN 978-601-7276-97-3

Оқу құралында «Алгоритмдік тілдерде бағдарламалау» пәнінің негізгі түсініктері, алгоритм түсінігі, оның қасиеттері, түрлері, алгоритмді бағдарламалау, Паскаль тілі, оның негізгі түсініктері, тіл операторлары, мүмкіндіктері қарастырылған.

Оқу құралы ЖОО-ң «Есептеуіш техникасы және бағдарламалық қамтамасыз ету» мамандығына оқытылатын «Алгоритмдік тілдерде бағдарламалау» курсының типтік бағдарламасы мен мемлекеттік стандартқа сәйкес келеді.

УДК 004.43(075)

ББК 32.973-018.1я7

А 39

Ш.Есенов атындағы Каспий мемлекеттік технологиялар және инжиниринг университетінің Оқу-әдістемелік Кеңесінің шешімі бойынша баспаға ұсынылған.

ISBN 978-601-7276-97-3

© Ш.Есенов атындағы КМТЖИУ, 2011

КІРІСПЕ.

ЭЕМ-дер пайда болғаннан кейін оларға сәйкес келетін бағдарлама керек болды. ЭЕМ әр түрлі жаңа есептерді шешуіне байланысты бағдарламалық қамсыздандыру да дами түсті.

Бір мәселені компьютерде шешу үшін оның алгоритмін жасап, бұл алгоритмдерді компьютер түсінетін нұсқаулар және заң-ережелер негізінде жазуымыз керек болады. ЭЕМ-де есептерді шешу үшін қарапайым бағдарламаларды құрып білудің өзі компьютерлік сауаттылықтың бір бөлігі болып саналады.

Бағдарлама дегеніміз – бұл компьютер орындайтын командалар жиынтығы. Басқаша айтқанда, алгоритмді машина түсінетін тілде жазу.

ЭЕМ көмегімен есептерді шешу үшін бағдарламалық тіл қажет болды. Машиналық кодпен бағдарламалау деп нақты алгоритмді машиналық бұйрық тілінде сипаттауды айтады. Программистер машиналық операциялардың барлық кодтарын, олардың қолдану ерекшеліктерін және оларды сақтайтын жадының нақты ұяшықтарының адресін білу керек. Мұндай бағдарламаларды құру кезінде өте қиын болды. Сондықтан, программистер өздерінің жұмыстарын жеңілдету үшін машиналық кодтарға символдық аттарды аударатын тіл ойлап тапты. Бір сөзге сәйкес келетін екілік кодты есте сақтағаннан көрі, сол сөздің өзін есте сақтаған оңай. Символдық белгілермен жұмыс істейтін бағдарламаларды **ассемблер** деп атайды. Мұндай машиналық бағдарламалық тілдер төменгі деңгейдегі тілдерге жатады.

1955 ж. ең бірінші жоғарғы деңгейдегі тіл пайда болды. Мұндай тілде жазылған бағдарламалар жеке машиналар бұйрықтарының жиынтығынан емес, **операторлар** деп аталатын ірі элементтерден тұрады. Осындай тілдер алгоритмді сипаттауға бағытталған. Сондықтан, олар **алгоритмдік тілдер** деп те аталады. ЭЕМ тек машиналық тілді түсінетін болғандықтан алгоритмдік тілде жазылған бағдарлама орындалар алдында **транслятор** деп аталатын арнайы бағдарлама арқылы машиналық тілге аударылады. **Translator** ағылшын тілінен аударғанда “аудармашы” дегенді білдіреді. Трансляторда бағдарламадағы алгоритмдік тілдің барлық ережелері және оны машиналық тілге аударатын құрылғы енгізілген. Транслятордың екі түрі бар:

- 1) Интерпретатор (Interpretation) – бағдарламалау тілінде берілген бағдарламаны аударатын отырып орындау әдісі.
- 2) Компилятор (Compile-собрать) – бағдарламаның барлық мәтіндері аударылып, жинақталғаннан кейін орындалатын әдіс.

Интерпретатор мен компилятордың айырмашылығын қарапайым мысалмен қарастырайық. Мысалы, дәріханада латын тіліндегі рецепт бойынша дәрі дайындау керек. Оның екі жолы бар. Бірінші жолы, рецепті өз тілімізге аударып, содан кейін ғана дәрі дайындайды (к). Екінші жолы, рецепті өз тілімізге аударатын отырып, дәрі дайындау (и). 1-ші жағдайда өз тіліміздегі аудармамен дәрі аламыз, ал 2-ші жағдайда өз тіліміздегі аудармасыз дәріні бірден аламыз. Егер, дәріні тағыда дайындау керек болса, онда рецепті тағы да аудару керек болады.

Интерпретатор күрделі емес трансляторлар қолданылатын тілдерде (Бейсик), компилятор қолданылуы қиын, мүмкін емес тілдерге (ЛИСП) қолданылады.

Ең алғашқы алгоритмдік тіл 1955 ж. пайда болды. Ол Фортран тілі. Ол ғылыми-техникалық және инженерлік есептерді шешуде қолданылды. Фортранды (Formula translator ағылшын сөзінің алғашқы буындарынан құрылған – формулалар аудармасы) Джон Бэкусаның басқаруымен IBM фирмасының қызметкерлері құрды. Көп жылдар өте келе Фортранның көптеген нұсқалары пайда болды. Осы тілмен көптеген тілдердің кітапханасы жазылды.

1960 ж. Алгол (Algoritm language – алгоритмдік тіл) пайда болды. Бұл тіл де ғылыми жұмыстарға арналды және соңғы тілдердегі блоктық құрылым түсінігі енгізілді.

IBM фирмасының қолдауымен Кобол (Cobol ағылшынның Common business oriented language – жұмысқа бағытталып жалпы қабылданған тіл) тілі пайда болды. Ол экономикалық есептерді шешуге, ақпараттарды өңдеуге арналған.

1961 ж. американдық профессор Джон Маккарти “ЛИСП” (List processing language – тізім өңдеу тілі) бағдарламалау тілін ойлап тапты.

1965 ж. Бейсик тілі пайда болды.

1967-1968 жж. PL/1 (Programming language – бағдарламалық бағытталған әмбебап тіл) бағдарламалау тілі IBM фирмасында пайда болды. Бұл тіл қуатты, қиын тіл. Ол жоғарғы оқу орындарында, ғылыми зерттеу орталықтарында қолданылды.

1970 ж. Никлаус Вирт Цюрих политехникалық университетінде Паскаль тілін ойлап тапты. Бұл тіл бағдарламаларды үйренуге жақсы бағытталған және қазіргі заманғы ЭЕМ-де әртүрлі есептерді тиімді шешуге мүмкіндік береді.

1973 ж. Франциядағы Люммин университетінің Колмероз және басқа да ғалымдары “Пролог” (Prolog-programmation en logique – логикалық бағдарламалау) бағдарламалау тілін ойлап тапты.

1980 ж. Ада бағдарламалау тілі пайда болды. Бұл тіл ағылшын поэты Лорд Байронның қызы, есептеуіш техника тарихындағы алғашқы программист Ада Лавлейс атына берілді. Ада тілі американың қорғаныс министрінің сұранысы бойынша әмбебап бағдарламалау тілі ретінде Францияда құрылды. Ол ең жаңа, қуатты бағдарламалау тілі болды. Оны Паскаль мен Алгол-68 тілдерінің ортақ қасиеттерінен құрған және ол қосымша қасиеттерге де ие.

Кейбір салаларда есептерді тиімді шешетін арнайы тілдер бар.

1. АЛГОРИТМ ТҮСІНІГІ.

Барлық есептеу машиналарының негізгі ерекшелігі оның жұмысына басқарудың бағдарламалық принципі енгізілген. Бұл ЭЕМ нәтиже беру үшін жеңіл есептермен бірге күрделі есептерге де қолданушы орындалуын немесе бұйрықтар тізімін қолдану керек.

Сонымен ЭЕМ-де есептерді шешу үшін алдын-ала әрекеттерді анықтау (алгоритмдеу) керек. Алгоритм күнделікті өмірде де кездеседі. Мысалы, шай қайнату. Шәйнектің қақпағын ашу, су құю, қақпағын жабу, газ жағу, шәйнекті газге қою.

Алгоритм – ақпараттану пәнінің негізгі ұғымдарының бірі. Қоғам өмірінің қай саласында болмасын пайдалана білу үшін алгоритм ұғымын меңгеру керек.

«Алгоритм» сөзі мағынасы жағынан нұсқау, бұйрық, ереже, тәртіп, заң, жоба сөздеріне синоним болып келеді.

Алдын – ала анықталған мақсатқа жету үшін, есепті шешу үшін орындаушыға қажетті амалдардың тізбегін анық көрсететін ережелер жүйесін **алгоритм** деп атайды.

«Алгоритм» сөзі IX ғ-да өмір сүрген ұлы ғалым Мұхамед Әл-Хорезмидің атынан бұрмаланып шыққан.

Алгоритм түсінігі есепті шешу әдісі түсінігімен тығыз байланысты. Әдіс деп қатаң негізделген есепті шешу тәсілі мен оны қолдануға болатын берілген мәліметтер бойынша есептер тобын анықтау мақсатында құрылған тәсілді зерттеуді айтады.

Ал, алгоритм есепті шешу мен тәжірибеде қолдану әдісін сипаттау болып табылады. Ол әдісті зерттеу нәтижесі бойынша құрылады.

1.1. Алгоритмді жазу тәсілдері

Алгоритм әрқашанда орындаушы үшін жазылады. Ол адам, ЭЕМ және т.б. құрылғы болуы мүмкін. Алгоритмді сипаттау орындаушыға жазылған әрекеттер түсінікті болуы үшін соның тілінде жүзеге асырылады.

Алгоритмдегі бұйрықтарды, нұсқаулардың берілу түріне қарай алгоритмді жазу әдістерін ажыратуға болады. Орындаушының өзіне тән біліміне байланысты арнайы белгілер, сөздер, іс-қимылдар, схемалар арқылы алгоритмдерді жазудың тәсілдерін ұйымдастыруға болады.

Алгоритмді жасау барысында оны 3 тәсілмен сипаттауға болады:

- 1) Табиғи тілде
- 2) Блок схема түрінде
- 3) Арнаулы тілде

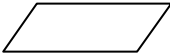
Орындаушы адам болатын жағдайда алгоритм көбінесе сөзбен жазылады. Сөзбен жазылған алгоритмдер ретпен орналасқан сөйлемдерден (нұсқаулардан) тұрады.

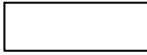
Қарапайым тілде әрбір адамға түсінікті етіп жазу *табиғи тілде жазу* деп аталады.

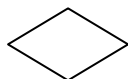
Блок схема деп әр түрлі геометриялық фигуралар арқылы бағдарламада орындалатын әрекеттер мен олардың орындалу ретін көрсететін графикалық схеманы айтады.

Блок схема алгоритмді сипаттаудың графикалық тәсілі. Алгоритмнің әр пункті геометриялық фигура – блоктың ішінде бейнеледі. Орындалатын іс-әрекеттердің түріне қарай оларға әртүрлі геометриялық фигуралар сәйкес келеді. Геометриялық фигуралар арасындағы байланыс жолдары бағыттау сызықтары арқылы көрсетіледі.

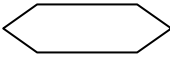
1.  – алгоритмнің басы мен соңы (эллипс)

2.  – енгізу/шығару блогы

3.  – орындалу блогы



4. — шартты тексеру блогы

5.  — цикл

6.  — байланыстыру сызықтары

Арнаулы тілде жазу деп ЭЕМ түсінетін тілде (бағдарлама) жазуды айтады.

1.2. Алгоритм түрлері

Алгоритмнің 3 түрі бар:

- 1) Сызықтық
- 2) Тармақталушы
- 3) Циклдық

Амалдардың бірінен кейін бірі орындалуын **сызықтық алгоритм** деп атайды.

Есепте шешуде барлық уақытта сызықтық алгоритм түрінде өрнектеуге болмайды. Амалдар тізбегін таңдау қандай да бір шартқа байланысты болса, онда ондай алгоритмді **тармақталушы алгоритм** деп атайды.

Кейбір амалдар бірнеше рет қайталанатын болса, онда ондай алгоритмді **циклдық алгоритм** деп атайды.

Төменде сызықтық, тармақталушы және циклдық алгоритмнің түрлеріне есептер шығару. Есептер табиғи және блок схема түрінде жазылады.

Мысалы:

- 1) Герон формуласы бойынша үшбұрыштың ауданын табу. Үшбұрыштың 3 қабырғасы белгілі (3 қабырғаны есепті шешуші пернетақтадан енгізу қажет). Ауданын табу үшін бірінші үшбұрыштың мына формула бойынша периметрін табамыз: $p = \frac{a+b+c}{2}$. Периметрін тапқаннан кейін, ауданын есептейміз. Ол төмендегідей формуламен жүреді: $s = \sqrt{p(p-a)(p-b)(p-c)}$. Осы есепті табиғи тілде және блок схема түрінде сызу қажет.

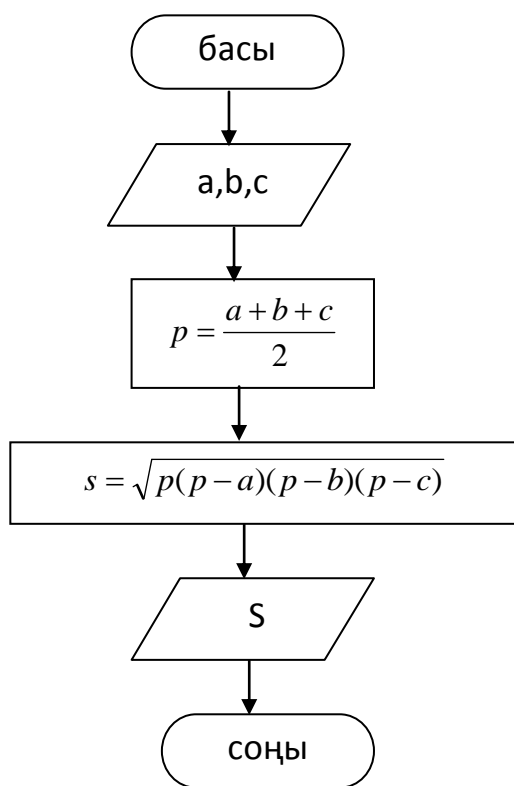
Жұмысты орындалу реті:

I. Табиғи тілде жазылуы:

- 1) алгоритмнің басы

- 2) a,b,c мәндерін енгізу
- 3) периметрдің формуласы бойынша P табу керек
- 4) ауданның формуласы бойынша S табу керек
- 5) ауданның мәнін баспаға шығару керек (периметрдің мәнін де баспаға шығаруға болады)
- 6) алгоритмнің соңы

II. Блок схема түрінде көрсету:



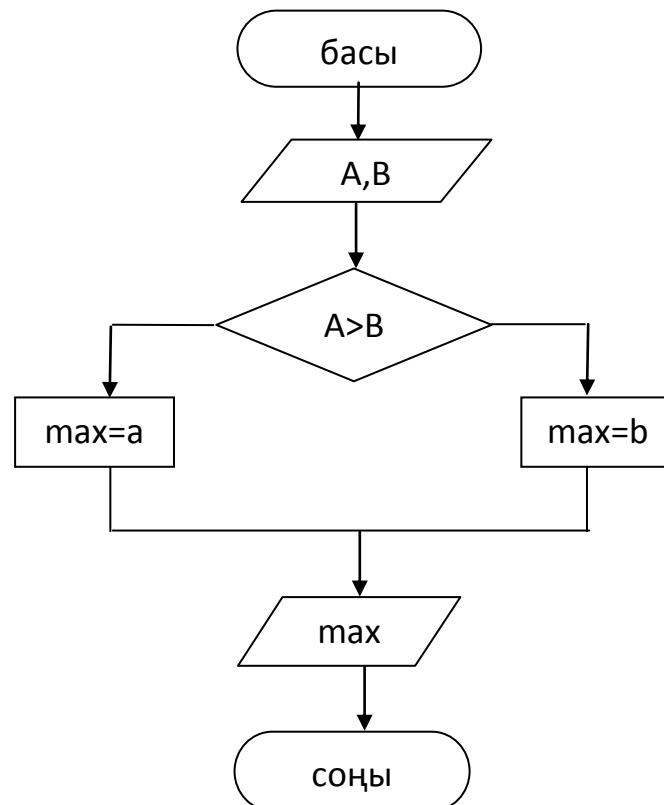
2) Екі санның үлкенін табу.

Жұмысты орындалу реті:

I. Табиғи тілде жазылуы:

- 1) алгоритмнің басы
- 2) a,b мәндерін енгізу
- 3) $a > b$ шартын тексеру
- 4) егер шарт орындалса, онда $\max = a$ амалы, әйтпесе $\max = b$ амалы орындалады
- 5) нәтижеге \max мәні шығады
- 6) алгоритмнің соңы

II. Блок схема түрінде көрсету:



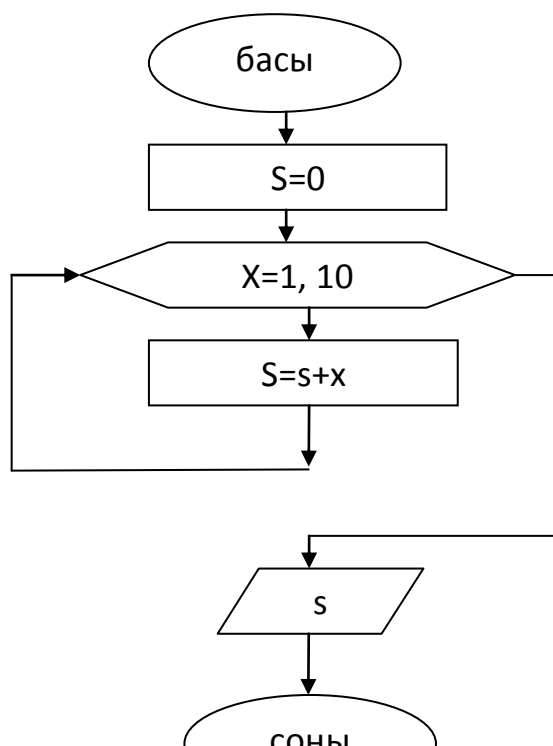
3) 1-ден 10-ға дейінгі сандардың қосындысын табу.

Жұмысты орындалу реті:

I. Табиғи тілде жазылуы:

- 1) алгоритмнің басы
- 2) қосындыны S деп алып, $S=0$ теңдігін жазу (ұандай санды 0-ге қосса да сол санның өзі шығады)
- 3) x -тің мәнін 1-ден 10-ға дейінгі сандарға теңестіру
- 4) $S=S+x$ амалы орындалады
- 5) нәтижеге S -тің мәні шығады
- 6) алгоритмнің соңы

II. Блок схема түрінде көрсету:



1.3. Алгоритмнің қасиеттері.

Алгоритмді кез келген басқа жазулардан мағыналық қасиеттері арқылы ажыратуға болады. Алгоритмнің қасиеттері түсініктілік, дискреттілік, анықтылық, нәтижелік және жалпыға бірдейлігі.

Алгоритмнің түсініктілігі деп – орындаушының бұйрықтар жүйе құрамына енетін іс-әрекеттерді орындау, тексеру туралы жазбалар мазмұнын айтады. Алгоритм ЭЕМ қабылдайтын және сол бойынша қажетті амалдарды орындай алатын нұсқаулар түрінде берілуі керек.

Алгоритмнің дискреттілігі деп – бұйрықтарға бөліп тізбектелген ретпен орындалуын айтады. Оның бір бұйрығының орындалуының соңы мен келесі бұйрықтың басына сілтеме нақты анықталады.

Әрбір бұйрықты орындағанда алгоритмнің орындалуы аяқталды ма немесе келесі қандай бұйрық орындалуы туралы дәл мәліметтің болуын *алгоритмнің анықтылығы* деп атайды.

Алгоритм шектеулі қадамдарды орындап болғаннан кейін нәтижеге алып келеді. Мұндай қасиетті *алгоритмнің нәтижелігі* деп атайды.

Алгоритм бұйрықтары жалпыға бірдей болады. Мұндай қасиетті *алгоритмнің жалпыға бірдейлігі* деп атайды.

2. ПАСКАЛЬ БАҒДАРЛАМАЛАУ ТІЛІ.

1964 ж. ақпаратты өңдейтін халықаралық IFIP федерациясына Алгол-60 бағдарламалау тілінің жаңа нұсқасын шығаруға көптеген елдердің ғалымдары жиналды. Осы ғалымдардың ішінде Стэнфордтық университеттегі «Информатика» кафедрасының доценті швецариялық Никлаус Вирт та болды. Вирт Швецарияға қайтып оралғаннан кейін жаңа бағдарламалау тілін ойлап тапты. 1968 ж. Н.Вирт Георих қаласындағы технологиялық институтында Паскаль тілінің ең бірінші нұсқасын шығарды. 2 жыл өткеннен кейін осы тілдің бірінші компиляторы пайда болды. Жаңа тіл пайда болғаннан кейін оны тұтынушылар және тілге қызығушылар көбейді. 1974 ж. Паскаль тілінің жаңа нұсқасы пайда болды. Ал келесі тілдің стандарттық операторлары шықты.

Вирт Паскаль тілін шығаруда 2 ережеге сүйенді:

- 1) бағдарламаны оқуда тіл жүйелі болу керек;
- 2) тілді есептеуіш машинада пайдалануда қарапайым және тиімді болу керек.

Тілдің компиляторы кез келген алгоритмді бірнеше құрылымдық түрде көрсетуге болады.

1985 ж. ТурбоПаскаль тілінің үшінші нұсқасы пайда болды. Жалпы білім оырндарында 1-ші бағдарламалау тілі болып Паскаль тілі қолданылады. ТурбоПаскальдың төртінші нұсқасына бұрынғыларға қарағанда өзгерістер енді. ТурбоПаскальдың 4-нұсқасына Модуль-2 тілінен UNIT-концепциясы енгізілді. Бұл ТурбоПаскальда үлкен бағдарламаларды шешуге мүмкіндіктер берді.

ТурбоПаскальдың 5-нұсқасы пайда болғаннан кейін кәсіби тұтынушылардың мүмкіндіктері де көбейді. ТурбоПаскальдың осы нұсқасында графикалық мүмкіндіктер болды. Бұл мүмкіндікті ТурбоПаскаль пакетінде қамсыздандырылған VGA (VideoGraphicsArray) графикалық адаптермен қолданды.

ТурбоПаскальдың 6-нұсқасы объектілердің жиынтығын пайдаланып, тұтынушының қолданбалы есептерін, теориялық концепциясын бағдарламалауды ұсынды. Жаңа мүмкіндіктерді тәжірибелік мысалдарда қолдану негізінде мәтіндік редактор құрылды. ТурбоПаскальдың 6-нұсқасында тұтынушының жұмысын жеңілдету үшін мәтіндік редактор ғана емес, тышқанды да қолданды.

1992 ж. Borland International фирмасы ТурбоПаскальдың 7-нұсқасын ойлап шығарды. Бұрынғы нұсқаларға қарағанда 7-нұсқасына көптеген өзгерістер енді. Біріншіден, мәтінді түстерге бояуға (тәжірибесіз тұтынушылар мәтін енгізу кезінде қатені жөндеп отыруға), екіншіден, күнделікті есептер шешуге (массивтерді шешуге) мүмкіндік берді.

2.1. Паскаль тілінің негізгі түсініктері.

Бағдарламалардың көпшілігі белгілі бір есепті шешу үшін жазылады. Ал есептің шешіміне ақпараттар мен мәліметтерді өңдеу арқылы жетуге болады. Сондықтан бағдарламалаушы ретінде сізге калай

- ақпараттарды бағдарламаға енгізу;
- ақпараттарды сақтау;
- берілгендерді өңдеудің дұрыс бұйрықтарын беру - амалдар;
- бағдарламадан берілгендерді алу (орындалулардың нәтижелері) – шығару керек екенін білу қажет.

2.1.1. Алфавит пен резервтелген сөздер.

Паскаль тілінің негізін құратын алфавит бар. Тілдің алфавитіне кіретіндер:

- 1) A-Z-ке дейінгі және a-z-ке дейінгі латын алфавитінің 26 әрпі;
- 2) 0-9 дейінгі араб цифрлары;
- 3) Арнайы таңбалар: + (қосу), - (азайту), * (көбейту), / (бөлу), . (нүкте), , (үтір), : (қос нүкте), ; (нүктелі үтір), @, #, \$;
- 4) Салыстыру амалдары: > (үлкен), < (кіші), >= (үлкен немесе тең), <= (кіші н/е тең), <> (тең емес);
- 5) Арифметикалық және логикалық функциялар: DIV, MOD, NOT, AND, OR, XOR, IN, SHL, SHR;
- 6) әртүрлі тыныс белгілер:
 - { } н/е (* *) – түсініктеме жазатын жақшалар;
 - [] – массив индекстерін және жиын элементтерін белгілеу;
 - () – айнымалылар тізімін белгілеу;
 - ' ' – апостроф;
 - := - меншіктеу белгісі.

Келесі сөздер Паскаль тілінде резервтелінген болып табылады:

absolute, and, assembler, asm, array, begin, case, const, constructor, destructor, div, do, downto, else, end, external, far, file, for, forward, function, goto, if, implementation, in, inline, interface, interrupt, label, mod, near, nil, not, object, of, or, packed, private, procedure, program, record, repeat, set, shl, shr, string, then, to, type, unit, until, uses, var, virtual, while, with, xor,

2.1.2. Өрнек, амалдар, операндалар

Өрнек – берілгендер элементтерімен орындалатын әрекеттердің орындалу тәртібін берді. Өрнек операндалар мен амалдардан тұрады.

Операнда деп әрекеттер орындалатын шама мен өрнектерді айтады.

Амалдар (операция) орындалуы қажет әрекеттерді (+, -, *, / және т.б.) анықтайды.

Мысалы: $x+y-10$

x , y , 10 - операндалар

$+$, $-$ – амалдар

Жақшалар амалдар болып табылмайды. Олар амалдардың орындалу тәртібін басқару үшін қойылады.

Turbo Pascal тілінде барлық амалдар арифметикалық ($+$, $-$, $*$, $/$, div , mod), салыстырмалы ($<$, $>$, $<=$, $>=$, $=$, $<>$), логикалық (or , xor , and , not) және жолдық болып бөлінеді.

Сонымен қатар барлық өрнектер де арифметикалық, салыстырмалы, логикалық және жолдық болып бөлінеді.

Арифметикалық деп тек арифметикалық амалдар қолданылатын арифметикалық типтегі операндалардан құрылған өрнекті айтады. Өрнек мәнінің типі операндалар мен амалдар типіне байланысты.

Салыстырмалы өрнек деп екі өрнектің салыстырмалы амалдармен байланысатын сөз тіркесін айтады. Салыстырылатын өрнектер кез келген скалярлық типтерге қатысты болуы мүмкін. Өрнек мәнінің типі барлық уақытта Boolean болады.

Логикалық деп логикалық амалдар көмегімен құрылған өрнекті айтады. Өрнек мәнінің типі барлық уақытта Boolean болады.

2.1.3. Идентификаторлар.

Идентификаторлар тұрақтылардың, типтердің, айнымалылардың, процедуралардың, модулдердің, бағдарламалардың және жазбалардағы өрістердің аттары ретінде пайдаланылады. Идентификаторлардың жүру аймағында оның мағынасы бір мәнді болуы қажет. Идентификаторлардың аты ретінде резервтелінген сөздерді пайдалануға тиым салынады. Паскальда идентификаторларды құрудың бірнеше ережелері бар. Олар:

- барлық идентификаторлар әріптен немесе сызу белгісінен басталуы қажет және оның құрамында бос орын болмауы керек. Символдар ретінде әріп, сызықша, сан болуы мүмкін. Басқа символдарды қолдануға болмайды;
- идентификаторлардың ұзындықтары әртүрлі болуы мүмкін;
- идентификаторлар үшін бас және кіші әріптердің айрмашылығы жоқ.

2.1.4. Turbo Pascal тіліндегі берілгендер типі.

Берілгендер типі – бұл мүмкін амалдардың біріккен жиынтығынан тұратын шамалар жиыны.

Әрбір тип өзінің мәндер диапазонына және өзінің сипаттамасы үшін резервтелген сөздерге ие.

Turbo Pascal тілінде дамыған стандартты типтер жүйесі бар. Сонымен қатар, бағдарламалаушы *Type* қызметші сөзін қолдана отырып, өзінің жеке типін құруға болады. Қолданушының жаңа тип форматының жазылуы төмендегідей болады:

Туре <тип аты>=<типті сипаттау>;

Turbo Pascal типтері **жай (скалярлы)** және **күрделі** болып бөлінеді. Типтер жүйесіндегі негізгі тип – *жай типтер*. Басқа типтер осы типтерден белгілі ереже бойынша құрылады. Стандартты скалярлық типке жататын айнымалылар *бүтін, нақты, символдық* және *бульдік* болып бөлінеді.

Бүтін тип – арифметикалық өрнектерде қолданыла алатын және жадының 1-ден 4-ке дейінгі байтын алатын берілгендер жиынтығы. Бүтін тип **Integer** стандартты атауымен белгіленеді. Бүтін типке жататын типтер тізімі 1-кестеде көрсетілген:

1-кесте

| Тип идентификаторы | Диапазон | Қажетті жады |
|--------------------|----------------------------------|--------------|
| Byte | [0..255] | 1 |
| Word | [0..65535] | 2 |
| Shortint | [-128..127] | 1 |
| Integer | [-32768..31767] | 2 |
| longint | [-2 147 483 648.. 2 147 483 647] | 4 |

Нақты тип – нақты санның ішкі ұсынылу форматын анықтайтын мәндердің соңғы санына ие. Бірақ, нақты санның мүмкін мәндері санының үлкендігі сондай, әр бірімен бүтін санды салыстыру мүмкін емес. Ішкі ұсынылуда нақты тип 4-тен 10-ға дейінгі байтқа, [2.9E-39; 1.7E+38] аралығындағы мәндерге ие. Нақты тип **Real** стандартты атауымен белгіленеді. Нақты типке жататын типтер тізімі 2-кестеде көрсетілген:

2-кесте

| Тип | Диапазон | Цифр мәндері | Байттық өлшем |
|----------|---------------------|--------------|---------------|
| Real | 2.9E-39 ..1.7E+38 | 11-12 | 6 |
| Single | 1.5E-45 .. 34E38 | 7-8 | 4 |
| Double | 5E-324 .. 1.7E308 | 15-16 | 8 |
| Extended | 1.6E-4951..1.1E4932 | 19-20 | 10 |

Char – ASCII кодтық кестесінің символдар жиынынан тұратын тип. Ішкі ұсынылуда ол 1 байтқа ие. Бұл типтің мәндер жиыны бекітілген және реттелген. Барлық символдар 0-ден 255-ке дейін нөмерленген. Символдық мәндер апострофқа (‘ ’) алынып жазылады.

Символдық типтермен салыстыру амалдарын орындауға болады. ASCII кодтық кестесіндегі реттік нөмері үлкен символ үлкен болып табылады.

Ескерту! ASCII кодтық кестесінде бірінші басқару символдары, содан кейін арнайы символдар, цифрлар, латын әріптерінің бас әріптері одан кейін кіші әріптері, орыс әріптерінің бас әріптері одан кейін кіші әріптері орналасады.

Boolean – False (жалған) және True (ақиқат) екі элементтен тұратын логикалық мәндер диапазонын анықтайтын логикалық тип. Ішкі ұсынылуда ол 1 байтқа ие. Ол салыстырмалы амалдар мен логикалық амалдарда қолданылады. Берілгендер типімен салыстыру амалдарын қолдануға болады.

False<True

Күрделі типтер скалярлық айнымалылардың реттелген жиынтығынан анықталады және өзінің компоненттері типімен сипатталады. Турбо Паскальда күрделі типтер *жол*, *массив*, *жазу*, *жиын* және *файл* болып бөлінеді.

Жолдық тип – string

Массив – array

Жазу (жазба) – record

Жиын – set

Файл – file, text.

2.1.5. Стандартты функциялар.

Берілгендер типіне байланысты айнымалылар үшін төмендегідей стандартты функциялар анықталады:

ABS(x) – x-тің абсолют мәні

COS(x) – косинус x

SIN(x) – синус x

EXP(x) – e^x

LN(x) – натураль логарифм

FRAC(x) – x-тің бөлшек бөлігі

INT(x) – x-тің бүтін бөлігі

SQR(x) – x-тің квадраты

SQRT(x) – квадрат түбір

ROUND(x) – x мәнін бүтін мәнге дөңгелектейді

$\text{ARCCOS}(x) = \text{ARCTAN}(\text{SQRT}(1-\text{SQR}(x))/x)$
 $\text{ARCSIN}(x) = \text{ARCTAN}(x/\text{SQRT}(1-\text{SQR}(x)))$
 $\text{ARCTAN}(x)$ – арктангенс x

Арифметикалық өрнектерді Паскаль тілінде жазу.

Жұмысты орындалу реті

1. $\frac{a+b}{c+d} = (a+b)/(c+d)$
2. $\frac{3(a+b)^2}{cd} = 3*((a+b)+(a+b))/c*d$
3. $A\sqrt{B} = A*SQRT(B)$
4. $\frac{1}{2}\sin 35^\circ = 1/2*SIN(3.14*35/180)$
5. $a + \frac{b}{x} - \frac{2}{y} = a + b/x - 2/y$

2.2. Бағдарлама құрылымы

Паскаль тілінде бағдарламалар тақырыптан, блоктан және “.” (нүкте) белгісінен тұрады.

Бағдарламаның тақырыбы бағдарлама атынан және параметрлер тізімінен тұрады:

Program бағдарлама - аты;

Паскальда блок екі бөлімнен тұрады: сипаттау және орындалу. Әрбір блок процедураның немесе функцияның хабарлануының бөлігі не болмаса бағдарламаның немесе модульдің бөлігі болады. Хабарлау бөлімінде хабарланған барлық идентификаторлар мен белгілер блок үшін жергілікті болып табылады.

Орындалу бөлімі бағдарлама операторларын қамтитын құрамды оператордан тұрады.

Сипаттау бөліміне модульдерді іске қосу, белгілерді, тұрақтыларды, типтерді, айнымалыларды, процедуралар мен функцияларды сипаттау сияқты бөлімшелер енеді.

Паскаль тіліндегі бағдарламаның жалпы құрылымын келесі түрде көрсетуге болады:

Program бағдарламаның аты;

Uses модульдер бөлімі;

Label белгілерді сипаттау;

Const тұрақтыларды сипаттау;

Type мәліметтер типін сипаттау;

Var айнымалыларды сипаттау;

процедураларды (**Procedure**) және функцияларды (**Function**) сипаттау.


Begin

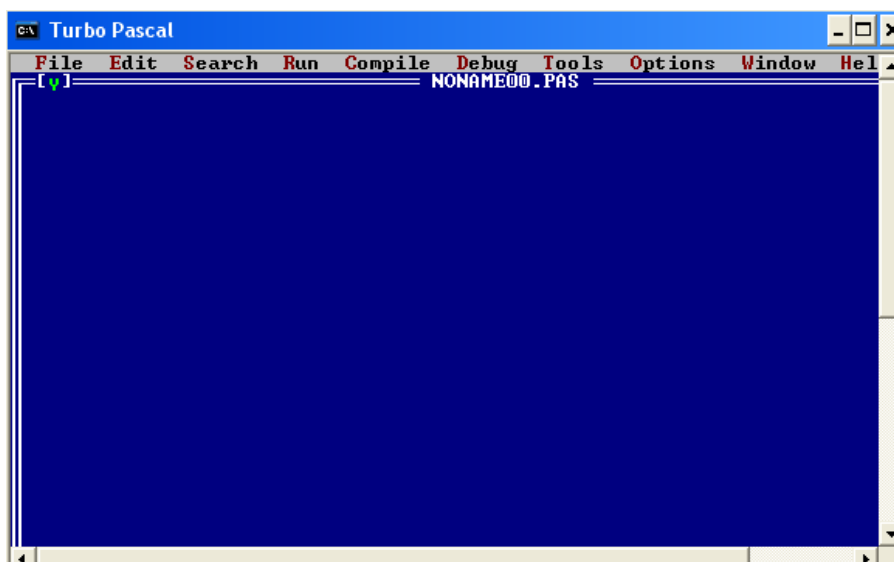
бағдарламаның денесі

End.

Бағдарламада барлық бөлімдердің болуы міндетті емес.

2.3. TurboPascal тілін іске қосу

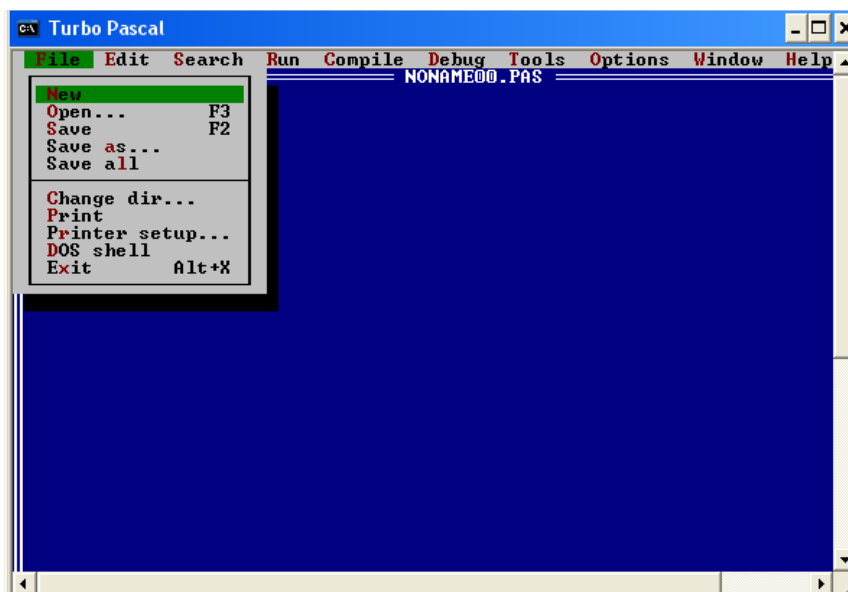
TurboPascal бағдарламалау тілін іске қосу үшін  батырмасын шерту қажет. TurboPascal бағдарламалау ортасының жалпы түрі:



1-сурет. TurboPascal бағдарламалау ортасы.

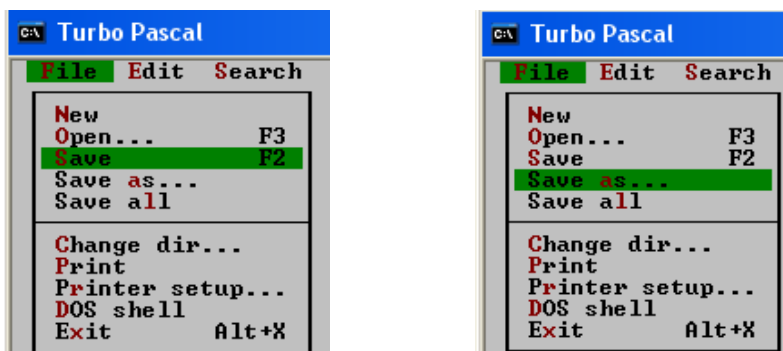
TurboPascal ортасы жоғарғы және төменгі менюлерден тұрады. Бағдарламаның дұрыстығын тексеру (компиляциялау) үшін *F9* пернесін шерту немесе **Compile** менюіндегі **Compile** бұйрығын таңдау керек. Ал бағдарламаны жіберу үшін *Ctrl+ F9* пернесін шерту немесе **Run** менюіндегі **Run** бұйрығын таңдау керек.

TurboPascal ортасында жаңа терезе ашу үшін **File**→**New** бұйрықтар тізімін орындау қажет.

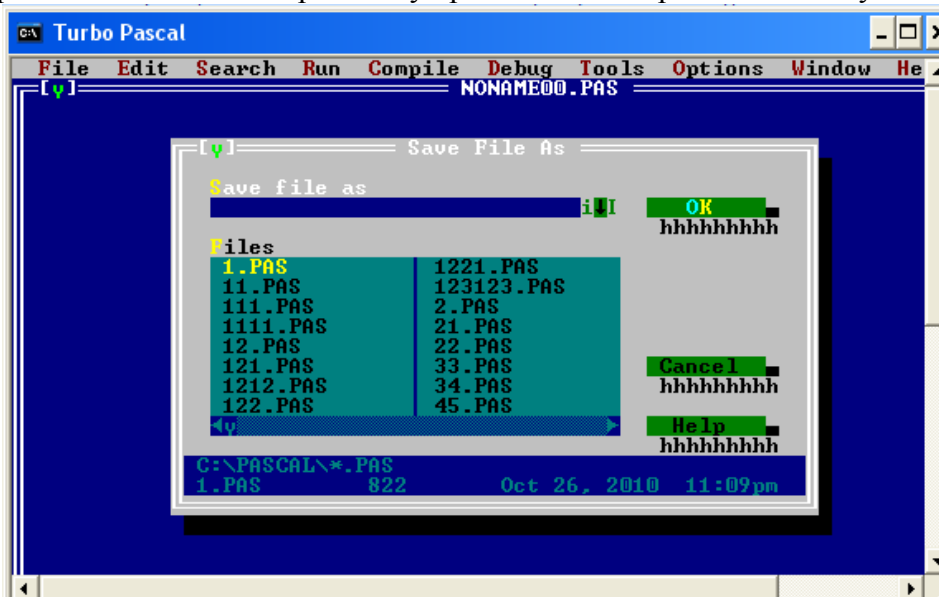


2-сурет. TurboPascal бағдарламалау ортасында жаңа терезе ашу.

Бағдарламаны сақтау үшін **File**→**Save** бұйрықтар тізімі, ал бірінеше жазылған бағдарламалардың барлығын бірден сақтау үшін **File**→**Save as** бұйрықтар тізімін орындау қажет.



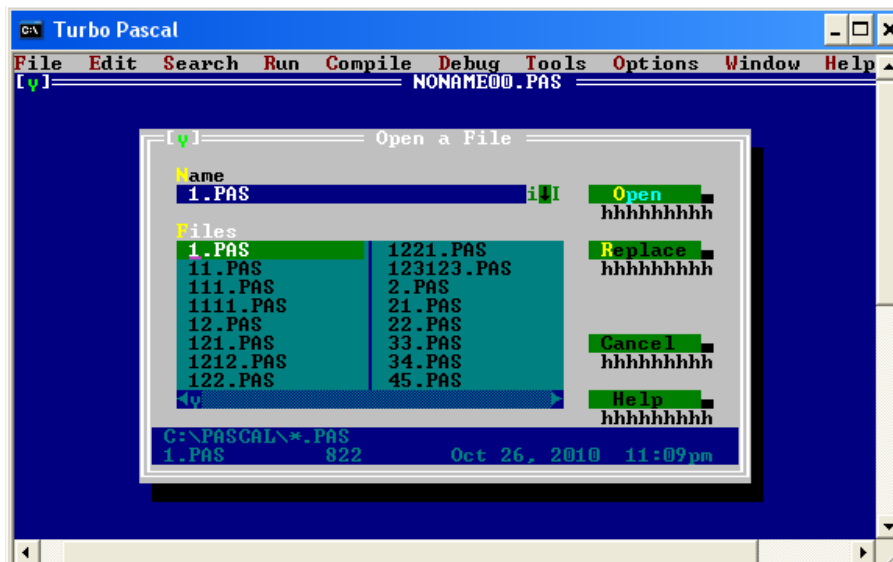
3-сурет. TurboPascal бағдарламалау ортасында бағдарламаны сақтау жолдары.



4-сурет. TurboPascal бағдарламалау ортасында бағдарламаны сақтау.

Сақталған бағдарламаларды ашу үшін **File**→**Open** бұйрықтар тізімін орындау қажет. Нәтижесінде төмендегідей сұқбаттасу терезесі ашылады. Аталған терезеден қажетті файлды таңдап, **Open** батырмасын шерту керек.





5-сурет. TurboPascal бағдарламалау ортасында бағдарламаны сақтау.

3. ПАСКАЛЬ ТІЛІНДЕ АЛГОРИТМНІҢ ТҮРЛЕРІН БАҒДАРЛАМАЛАУ

3.1. СЫЗЫҚТЫҚ АЛГОРИТМДІ БАҒДАРЛАМАЛАУ

Алгоритмнің машина түсінетін тілде жазылуын **бағдарлама** деп аталады.

Машиналық амалдарды түсіндіретін ережелер жүйесін **оператор** деп атайды.

Сызықтық бағдарлама деп реті бойынша белгілі бір тәртіппен орындалатын бағдарламаны айтады. Сызықтық бағдарламаны құру үшін мынадай операторлар қолданылады:

Read, Write, меншіктеу операторы - :=.

Read (ағылшынның “оқу” деген сөзінен алынған) бағдарламаға айнымалылардың мәнін енгізу үшін қолданылады (енгізу операторы). Айнымалы мәндерін енгізу *Enter* пернесі арқылы жүреді.

Write – бағдарлама нәтижесіндегі айнымалылардың мәнін баспаға шығару үшін қолданылады (шығару операторы).

Айнымалыға мәндерді және амалдарды меншіктеу үшін меншіктеу операторы := қолданылады. Меншіктеу операторы сол жақтағы айнымалы оң жақтағы мәнді өзіне меншіктейді.

Бағдарлама ішінде қосымша ескертулерді, түсініктемелерді орналастыруға болады. Ол үшін **апостроф** (‘ ’) қолданылады. Апострофқа жазылған мәтіндер бағдарламаның орындалуына ешқандай әсер етпейді.

Мысалы:

Үш санның қосындысын табатын бағдарлама құр.

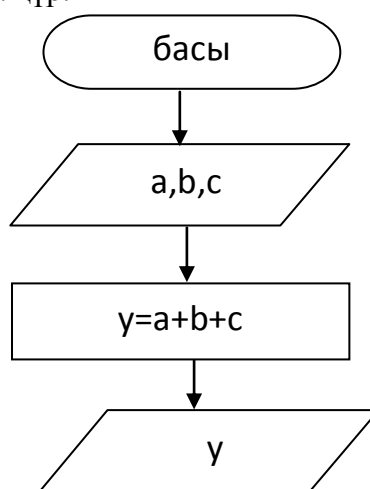
```
Program esep1;
```

```
Var a,b,c,y:integer;
```

```
Begin
```

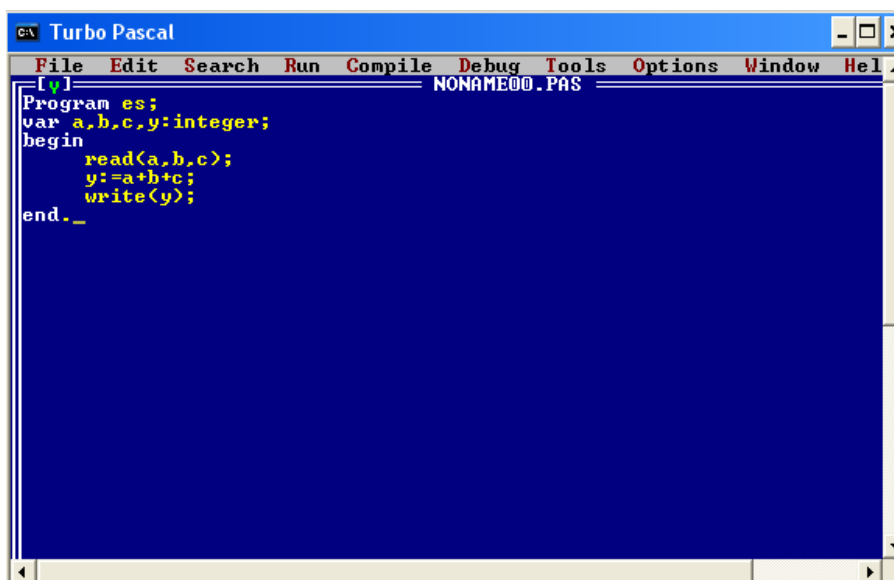
```
  Read (a,b,c);
```

```
  y:=a+b+c;
```



Write (y);

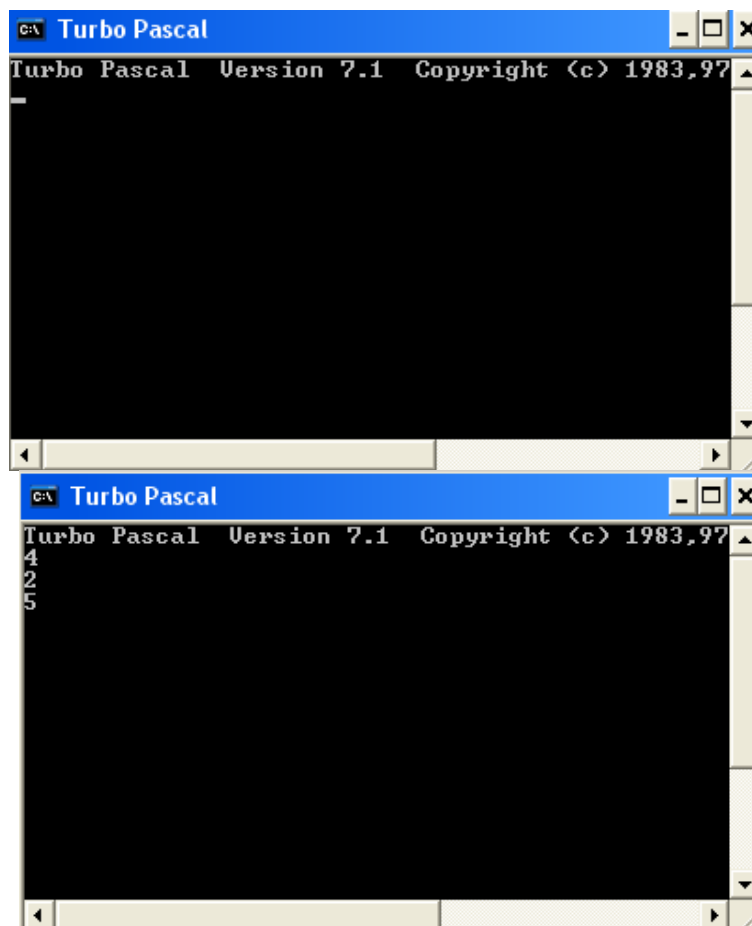
End.

A screenshot of the Turbo Pascal IDE window. The title bar reads "Turbo Pascal". The menu bar includes "File", "Edit", "Search", "Run", "Compile", "Debug", "Tools", "Options", "Window", and "Help". The main editing area has a dark blue background with white text. The code is as follows:

```
[v] NONAME00.PAS  
Program es;  
var a,b,c,y:integer;  
begin  
    read(a,b,c);  
    y:=a+b+c;  
    write(y);  
end._
```

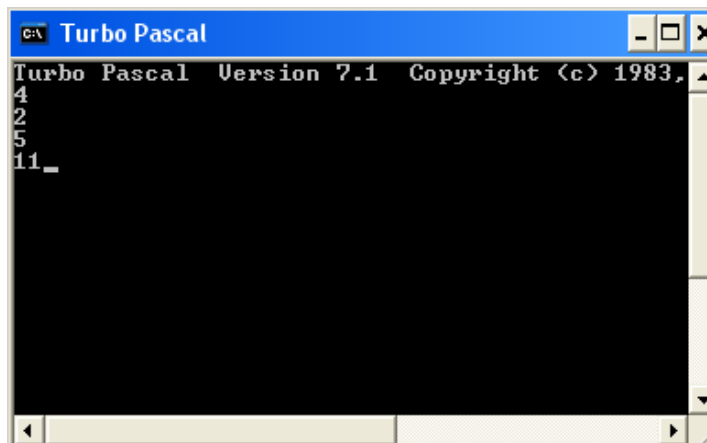
6-сурет. Сызықтық бағдарлама құру.

Бағдарламаны жіберу үшін **Ctrl+F9** батырмаларын басу қажет. Ашылған терезеде a,b,c мәндерін енгізіледі.



7-сурет. Мәндерді енгізу.

Нәтижені қарау үшін **Alt+F5** батырмаларын басу керек.



8-сурет. Нәтижені тексеру.

Alt+F5 батырмаларын баспау үшін бос **Readln** операторын жазу керек.

Ln жұрнағы курсорды жаңа жолдан бастайды.

Бағдарлама Program сөзінен кейін басталады. Begin, End қызметші сөздері операторлық жақшалар деп аталады. Бағдарламаның негізгі бөлімі Begin сөзінен басталады.

3.2. Тармақталушы алгоритмді бағдарламалау

Есепті шешуде барлық уақытта сызықтық алгоритм түрінде өрнектеуге болмайды. Кейбір есептерді шешу кезінде бұл тәртіпті өзгертуге тура келеді. Осы мақсатпен басқаруды беру операторлары If, Then, Else, Goto операторлары қоладанылады.

If – шартты өту операторы. Егер шарт орындалса, онда Then операторынан кейін тұрған әрекет орындалады, ал шарт орындалмаса Else операторынан кейін тұрған әрекет орындалады. Оның жалпы жазылу түрі:

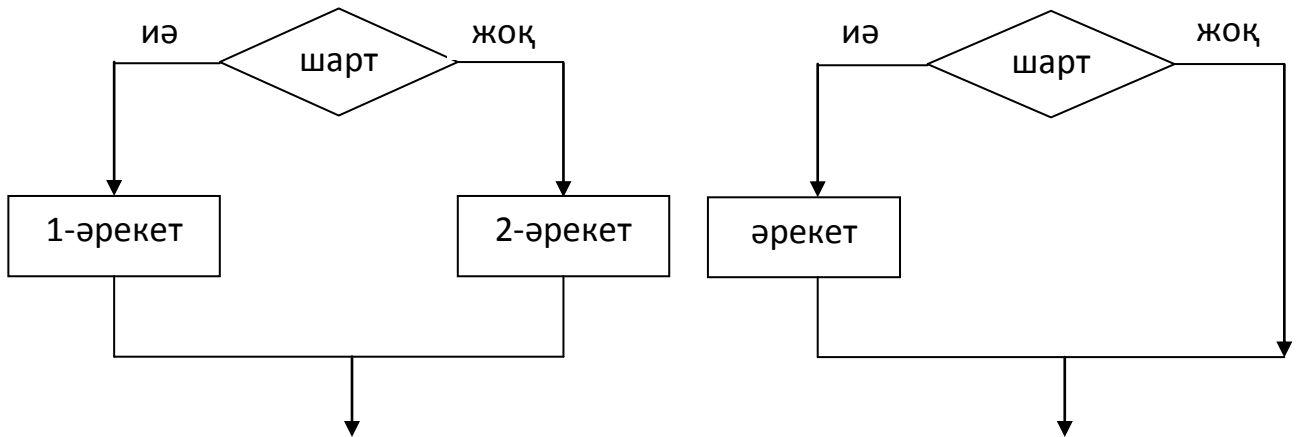
n If шарт Then 1-әрекет Else 2-әрекет

If – егер

Then – онда

Else - әйтпесе

Шартты өту операторының екі түрі бар: толымды шарт және толымсыз шарт.



Мысалы: Екі санның үлкенін табатын бағдарлама құру.

```
Program esep1;
```

```
Uses CRT;
```

```
Var a,b,max:integer;
```

```
begin
```

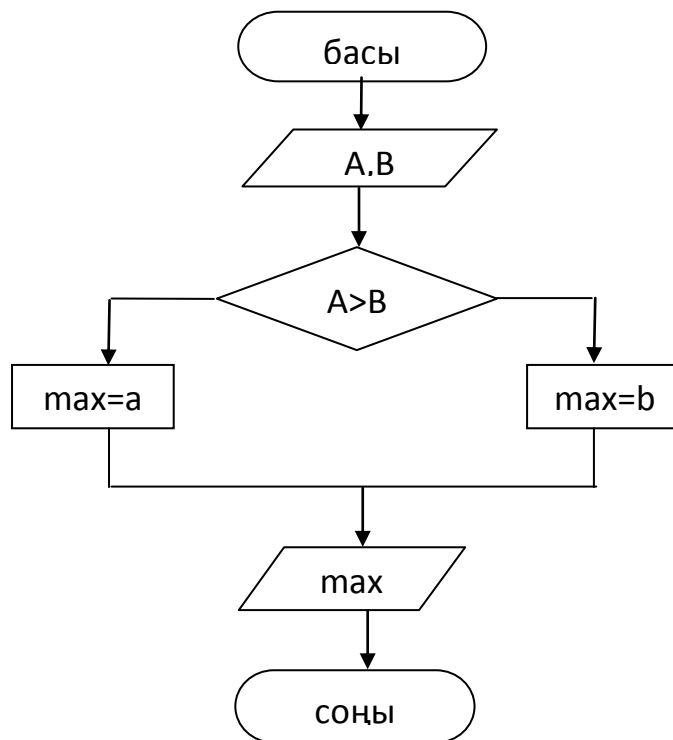
```
ClrScr;
```

```
    readln(a,b);
```

```
    If a>b Then max:=a Else max:=b;
```



```
writeln(max);  
end.
```



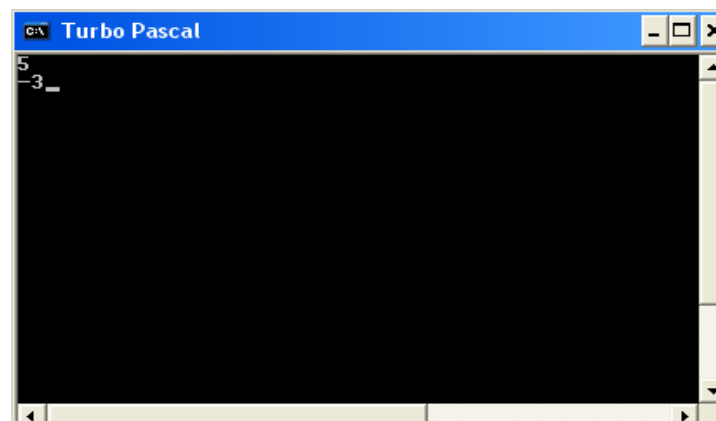
```
File Edit Search Run Compile Debug Tools Options Window Help  
NONAME00.PAS  
Program es;  
var a,b,max:integer;  
begin  
  read(a,b);  
  if a>b Then max:=a Else max:=b;  
  write(max);  
end.
```

The screenshot shows the Turbo Pascal development environment. The window title is "Turbo Pascal". The menu bar includes "File", "Edit", "Search", "Run", "Compile", "Debug", "Tools", "Options", "Window", and "Help". The main text area contains the following Pascal code:

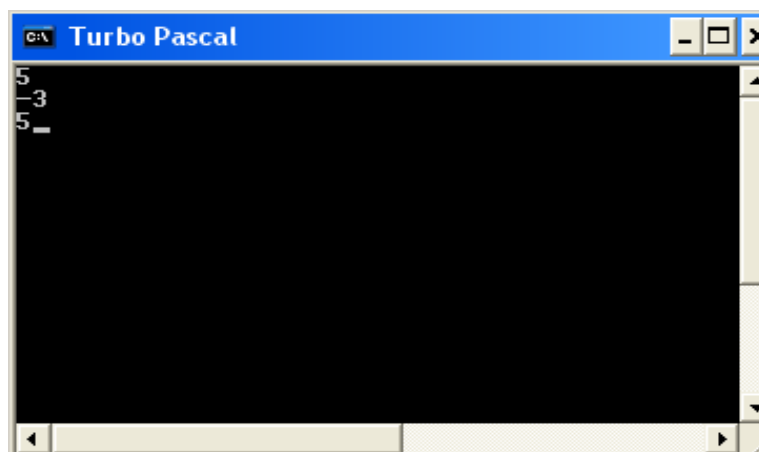
9-сурет. Тармақталушы бағдарлама құру.

Бағдарламаның орындалу реті:

If операторы a және b мәндерін салыстырады. Егер шарт қанағаттандырса, Then операторынан кейінгі max:=a амалы орындалады.



10-сурет. Мәндерді енгізу.



11-сурет. Нәтижені тексеру.

Ал шарт қанағаттандырмаса, Else операторынан кейінгі max:=b амалы орындалады. Амалдар орындалғаннан кейін max мәні нәтижеге шығады.

Бағдарламада Uses CRT модулі іске қосылады. Аталған модульдегі ClrScr бұйрығы орындалады. Бұл бұйрық экранды тазалау болып табылады.

Мысалы: y мәнін төмендегі формула бойынша есептейтін бағдарлама құру.

$$y = \begin{cases} x^2, & x > 0 \\ 2x, & x < 0 \end{cases}$$

```

Program mysal1;

Var x,y:integer;

Begin

    Readln (x);

    If x>0 Then y:=Sqr(x) Else y:=2*x;

    Writeln(y);

End.

```

Goto – шартсыз өту операторы. Goto операторынан кейін орындалатын жолдың белгісін жазу керек. Оның жалпы жазылу түрі:

```
Goto m
```

мұндағы m – орындалатын жолдың белгісі

Goto операторы бағдарламаның орындалуы рет-ретімен орындалмай, белгіге өту керек болатын болса қолданылады. Егер Goto операторы жол ішінде басқа операторлармен бірге жазылатын болса, онда ол операторлардың ең соңғысы болуы тиіс. Әйтпесе, бұл оператордан кейін жазылған операторлар ешқашан орындалмайды.

Белгі Label – белгілерді сипаттау бөлімінде сипатталады.

3.3. Таңдау операторы

Case операторы екі немесе бірнеше берілген амалдардан бір амалды таңдауды қамтамасыз етеді. Операитодың жалпы жазылу түрі:

```

Case <өрнек> of

< өрнектің мәні>:<оператор 1>;

< өрнектің мәні>:< оператор 2>;

...

< өрнектің мәні >:< оператор n>

else < оператор n+1>;

end;

```

Өрнектің орнында реттік типтің кез келгені бола алады. өрнектің мәнінәі орнына аталған типтің кез келген тұрақтысы бола алады. Оператордың орнына Паскаль тілінің кез келген операторы бола алады. Case операторы әр уақытта End қызметші сөзімен аяқталады.

Case операторының орындалуы:

Case сөзінен кейінгі өрнек орындалады. Өрнектің мәнінің тізімі қарастырылады. Егер тізімнен табылатын болса, онда мәнмен белгіленген оператор орындалады. Ал тізімде жоқ болатын болса Else операторынан кейінгі амал орындалады. Егер Else операторы болмаса Case операторынан кейінгі амалдар орындалады.

Енгізілген санға сәйкес келетін апта күнін шығаратын бағдарлама құру.

```
Program apta;
```

```
Var x:1..7;
```

```
Begin
```

```
  Writeln('апта күнінің нөмірін енгіз');
```

```
  Readln(x);
```

```
  Case x of
```

```
    1: writeln('дүйсенбі');
```

```
    2: writeln('сейсенбі');
```

```
    3: writeln('сәрсенбі');
```

```
    4: writeln('бейсенбі');
```

```
    5: writeln('жұма');
```

```
    6: writeln('сенбі');
```

```
    7: writeln('жексенбі');
```

```
  end;
```

```
End.
```

3.4. Циклдық алгоритмді бағдарламалау

Цикл – бұл бағдарламаның орындалу кезінде бірнеше рет қайталанатын бөліктер. Циклдерді пайдалану арқылы салыстырмалы түрде қысқа бағдарламалар жазуға, бағдарламаның көлемін қысқартуға болады. Циклды күрделі түрде бағдарламалау үшін шартты және шартсыз операторларды қолдануға болады. Циклдық алгоритмді бағдарламалаудың үш түрі бар.

3.4.1. Параметрлі цикл

Цикл операторлары белгілі бір операторлардың қайталанып орындалуын қамтамасыз етеді. Циклдық оператор:

FOR параметр = бастапқы мән TO соңғы мән DO

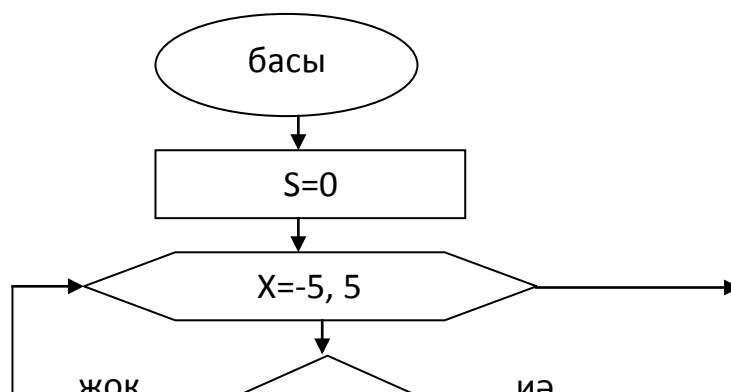
Егер қайталаудың саны белгілі болса, онда FOR, TO, DO (DOWNTO) операторлары қолданады. FOR циклдық операторы оператордың белгілі бір рет қайталанып орындалуын қамтамасыз етеді.

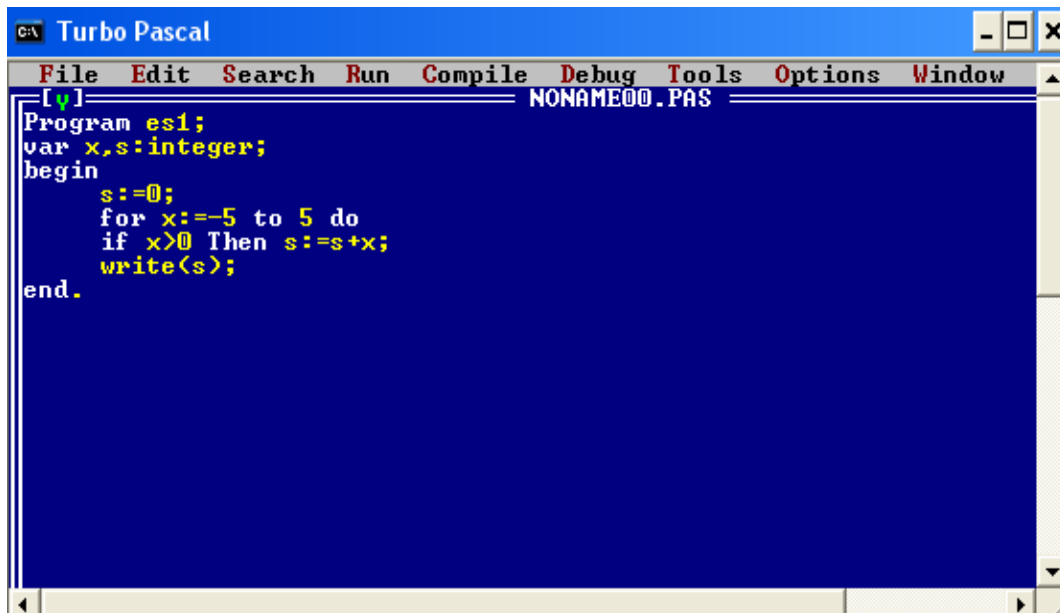
FOR операторы орындала бастаған кезде бастапқы және соңғы мәндерін бір рет анықтап алады және бұл мәндер FOR операторының орындалу барысында сақталады. Айнымалының типі реттік болуы керек. Бастапқы және соңғы мәндердің типі осы реттік типпен меншіктеу бойынша сәйкес болуы керек.

FOR операторында тұрған оператор бастапқы және соңғы мәндер диапазонындағы әр мән үшін бір рет орындалады.

Мысалы: -5-тен 5-ке дейінгі сандар берілген. Осы сандардың оң сандарының қосындысын табатын бағдарлама құру.

```
Program prim1;  
Var x:integer;  
    S:real;  
Begin  
    S:=0;  
    For x:=-5 to 5 do  
        If x>0 Then S:=s + x;  
    Writeln(S);  
End.
```





```
File Edit Search Run Compile Debug Tools Options Window
NONAME00.PAS
Program es1;
var x,s:integer;
begin
  s:=0;
  for x:=-5 to 5 do
    if x>0 Then s:=s+x;
  write(s);
end.
```

12-сурет. Параметрлі циклға бағдарлама құру.

Бағдарламада x параметр мәні -5 -тен 5 -ке дейінгі сандарды қарастырады. Осы аралықтағы оң сандардың қосындысын тауып, баспаға шығарады. Қосындыны тапқан уақытта қандай санды 0 -ге қосса да, сол санның өзі шығады. Сондықтан S -тің бастапқы мәнін $S=0$ деп аламызу. Осыдан кейін $x>0$ шарты тексеріледі. Егер шарт қанағаттандырса, $S=s + x$ амалы орындалады. Бұл формула бойынша орындалу былай жүреді:

$$S1=s + x1=0+1=1$$

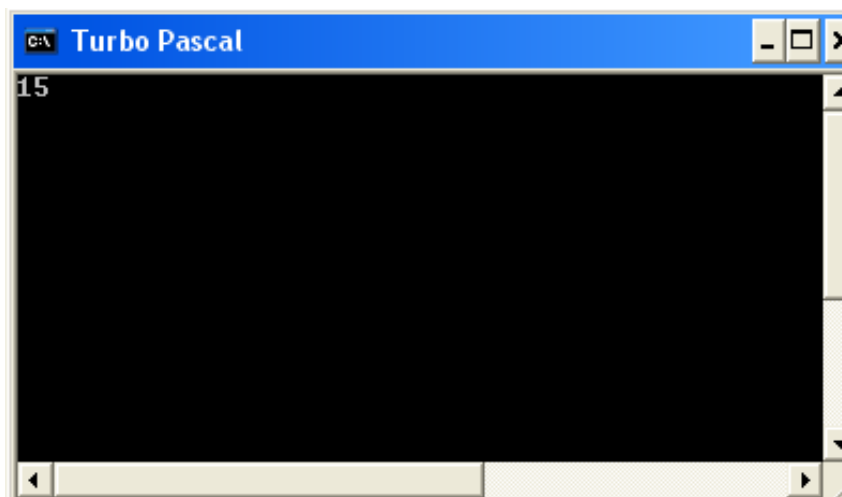
$$S2=s1 + x2=1+2=3$$

$$S3=s2 + x3=3+3=6$$

$$S4=s3 + x4=6+4=10$$

$$S5=s4 + x5=10+5=15$$

Жауабы 15 -ке тең болады.



13-сурет. Нәтижені тексеру.

3.4.2. Дейін циклы

While операторының құрамында операторының қайталанып орындалуын басқаратын өрнек болады. Дейін циклының жалпы жазылу түрі:

```
While <шарт> DO
```

```
  Begin
```

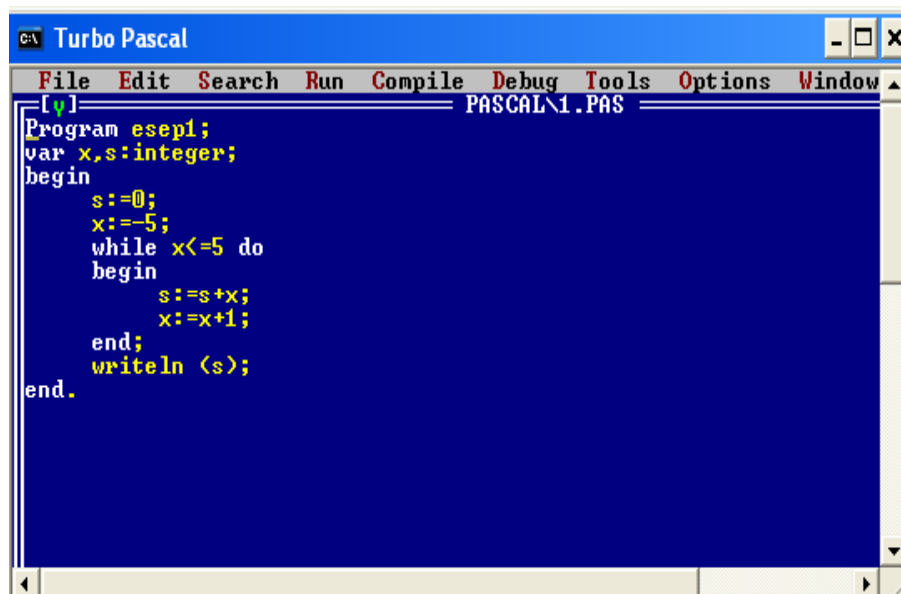
```
    операторлар;
```

```
  end;
```

Орындалу ережесі: Begin және End сөздерінің арасында жазылған операторлар шарт ақиқат болғанда орындалады. Шарт жалған болса орындалмайды. Егер алғаш рет тексерілгенде шарт жалған болса, онда операторлар бір рет те орындалмайды.

Мысалы: -5-тен 5-ке дейінгі сандар берілген. Осы сандардың оң сандарының қосындысын табатын бағдарлама құру.

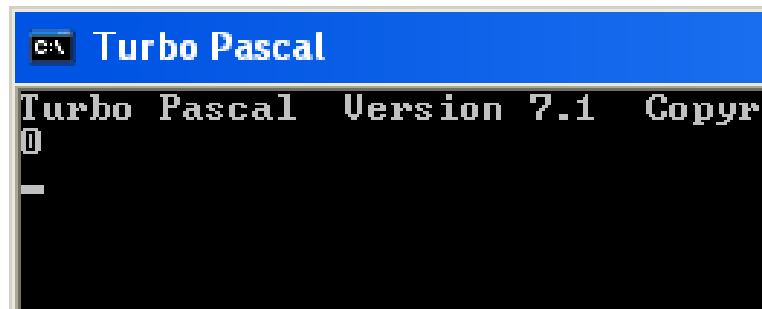
```
Program prim1;  
Var i:integer;  
    S:real;  
Begin  
    S:=0;  
    x:=-5;  
    While x<=5 do  
    Begin  
        S:=s + x;  
        x:=x+1;  
    end;  
    Writeln(S);  
End.
```



The image shows a screenshot of the Turbo Pascal IDE. The window title is "Turbo Pascal". The menu bar includes "File", "Edit", "Search", "Run", "Compile", "Debug", "Tools", "Options", and "Window". The main text area contains the following Pascal code:

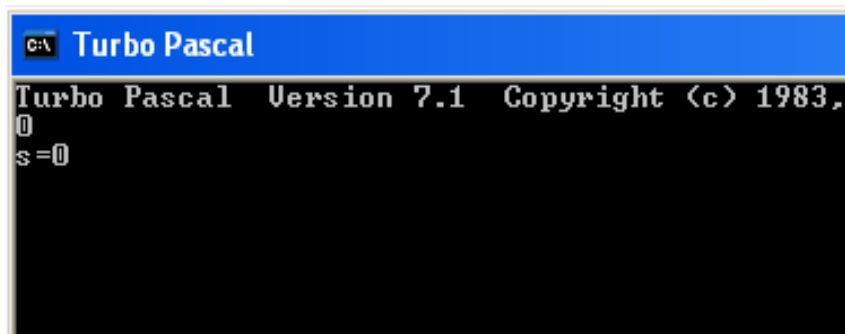
```
Program esep1;  
var x,s:integer;  
begin  
    s:=0;  
    x:=-5;  
    while x<=5 do  
    begin  
        s:=s+x;  
        x:=x+1;  
    end;  
    writeln (s);  
end.
```

14-сурет. Дейін циклына бағдарлама құру.



15-сурет. Бағдарлама жауабын шығару терезесі.

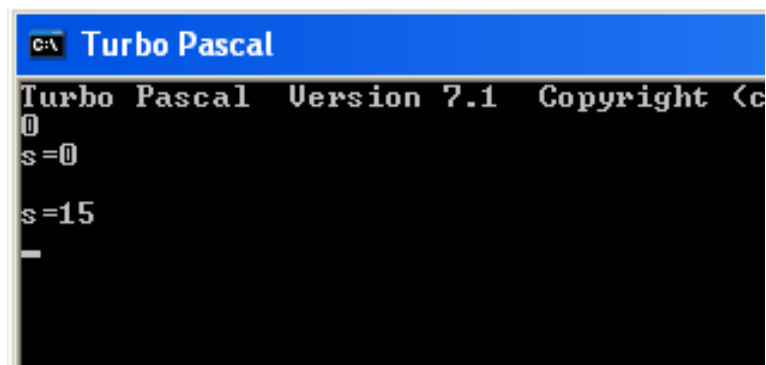
```
Writeln('S=', S);  
Readln;  
End.
```



16-сурет. Бағдарлама жауабын шығару терезесі.

Оң сандардың қосындысын табу керек болса төмендегідей жазылады:

```
If x>0 Then S:=s + x;
```



17-сурет. Бағдарлама жауабын шығару терезесі.

```
Writeln('S=', S);  
Readln;  
End.
```

3.4.3. Кейін циклы

Операторлар тізбегінің қайталанып орындалуын басқаратын Repeat операторы. Циклдық оператордағы өрнек сол Repeat операторының ішінде орналасады. Кейін циклының жалпы жазылу түрі:

```
Repeat  
    операторлар;  
Until <шарт>
```

Орындалу ережесі: Repeat сөзінен кейін жазылған операторлар бір рет орындалады. Шарт жалған болса, операторлар қайталанып орындала береді. Шарт ақиқат болғанда ғана тоқтайды.

Мысалы: -5-тен 5-ке дейінгі сандар берілген. Осы сандардың оң сандарының қосындысын табатын бағдарлама құру.

```
Program prim1;  
    Var i:integer;  
        S:real;  
    Begin  
        S:=0;  
        x:=-5;  
        Repeat  
            If x>0 Then S:=s + x;  
            x:=x+1;
```

```
Until x>5;  
Writeln(S);  
End.
```

Кейін циклының дейін циклынан айрмашылығы:

- өрнекті тексеру цикл денесінің соңында орындалғандықтан циклдегі операторлар ең болмаса бір рет орындалады. While циклында өрнектің мәні жалған болса, онда оның денесі бірден өткізіліп жіберіледі.
- Repeat циклы өрнек жалған мәнін қабылдағанша орындала береді.
- Repeat циклында құрамдас оператор емес, жай операторлар тобы пайдаланылуы мүмкін. Бұл циклды пайдаланған кезде While циклындағыдай Begin және End сөздері қолданылмайды.

Циклдық алгоритмге берілген мысалдар:

1. Экранға $y=x^2$ функциясы мәнінің кестесін шығару. x – бүтін және нақты болғанда,

Егер x – бүтін болса:

```
Program esep1;  
Var x:byte;  
Begin  
  For x:=-5 to 5 do  
    y:= x*x;  
    writeln ('x=', x:3, 'y=' ,y:5);  
Readln;  
End.
```

Егер x – нақты болса (0.3 қадамымен):

```
Program esep2;  
Var x:real;
```

```
Begin
    x:=-5;
    while i<=5 do
        begin
            y:= x*x;
            x:=x+0.03;
            writeln ('x=', x:3, 'y=' ,y:5);
        end;
    Readln;
End.
```

2. N натурал сандарының көбейтіндісін есептеу.
Program esep3;

```
Var i, N:byte;
    p: real;
Begin
    p:=1;
    Writeln('N енгіз');
    Readln(N);
    For i:=1 to n do
        p:=p+i;
        Writeln('көбейтінді=', p:10:2)
    Readln;
End.
```

3. а-дан б-ға дейінгі аралығындағы 3-ке бөлінетін сандардың санын табу.
Program esep4;

```

Uses CRT;

Var i, a, b, k:word;

Begin
ClrScr;

    k:=0;

    Writeln('a,b мәндерін енгіз');

    Readln(a, b);

    For i:=a to b do

        If i mod 3=0 then k:=k+1;

        Writeln('саны =', k)

Readln;

End.

```

4. Пернетақтадан қосындысы 100-ге жеткенше сандар енгізетін бағдарлама құру.

```

Program esep5;

Uses CRT;

Var N, Sum:byte;

Begin
ClrScr;

    S:=0;

    repeat

        Writeln('санды енгіз');

        Readln(N);

        s:=s+N;

    until s>100;

    Writeln('қосынды=', s)

Readln;

End.

```

4. ПАСКАЛЬ ТІЛІНДЕГІ МАССИВТЕР.

ЭЕМ-нің көмегімен шешілетін есептердің көбісі ақпараттың үлкен көлемдерін өңдеумен байланысты. Есептерді шешу кезінде осындай үлкен көлемдегі бір типтегі мәліметтерді (айнымалыларды) енгізуге және де осы мәліметтермен әртүрлі амалдар орындауға тура келеді. Сондықтан да бір амалды бірнеше рет қайталамас үшін және есептерді шешуді жеңілдету үшін айнымалыларды бір атауға біріктіруге болады.

Алгоритмдік тілде белгілі тәртіп бойынша орналасқан айнымалы шамалардың біріктірілген жиынтығын **массив** деп атайды. Ал, массивті құрайтын айнымалы шамаларды *массивтің элементтері* деп атайды. Массивпен жұмыс жасау үшін оның әр элементінің мәні белгілі болуы тиіс. Массивтегі әр элемент номерін *элементтің индексі* деп атайды. Массивтің аты болады, атты латын алфавитіндегі А-дан Z-ке дейін әріптермен белгілейді. Паскаль тілінде массивтің индексі массив атынан кейін тік жақшаға [] алынып жазылады.

Массивтің екі түрі бар:

- 1) бір өлшемді (сызықтық)
- 2) екі өлшемді (матрица)

4.1. Бір өлшемді массив

Бір өлшемді массив деп аталу себебі, айнымалылардың бір қатарға орналасуы. Бағдарламада массивті енгізбес бұрын, айнымалыларды сипаттау бөлімінде массив сипатталуы тиіс. Паскаль тілінде массивті сипаттау үшін ARRAY қызметші сөзі қолданылады. Атау қызметші сөзі массивтің элементтерінің санын, типін сипаттайды және ол одан кейін өзгермейді. Атау қызметші сөзінің жалпы жазылуы түрі:

Массивтің аты: array[1..n] of массив элементтерінің типі;

Массив элементтері i параметрімен белгіленеді.

Мысалы: 7 элементтен тұратын бір өлшемді В массиві берілген. Осы

массивтің элементтерін енгізетін және элементтерінің қосындысын табатын бағдарлама құру. Нәтижесінде экранға массивтің элементтері және элементтерінің қосындысы шығуы керек.

```
Program prim1;
```

```
Var I:integer;
```

```
S:real;
```

B:array[1..7] of real;

Begin

For i:=1 to 7 do

 Readln(b[i]);

 S:=0;

 For i:=1 to 7 do

 begin

 S:=s+b[i];

 Writeln(b[i]);

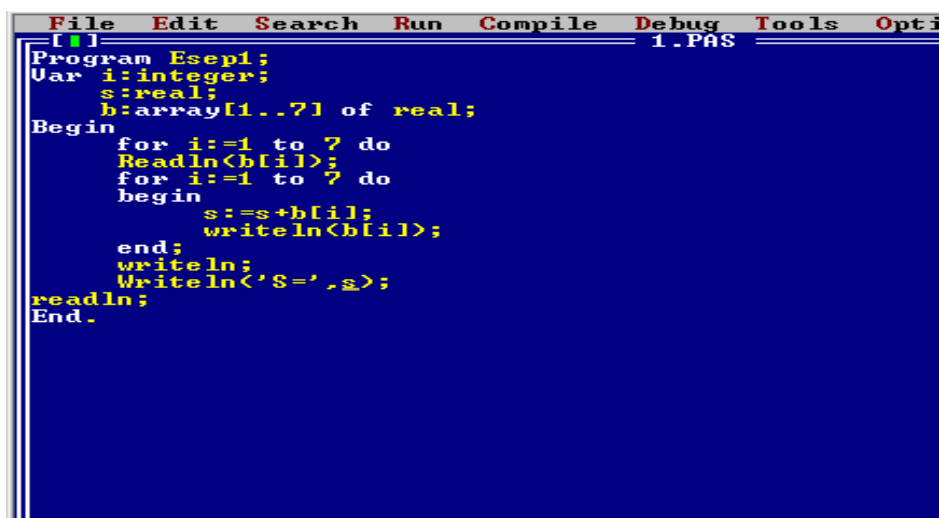
 end;

 Writeln;

 Writeln('s=',s);

Readln;

End.

A screenshot of a Pascal IDE window titled "1.PAS". The window has a menu bar with "File", "Edit", "Search", "Run", "Compile", "Debug", "Tools", and "Opti". The code is displayed on a dark blue background with yellow text. The code is identical to the text provided in the previous blocks, showing a Pascal program that reads an array of 7 real numbers, calculates their sum, and prints each element and the total sum.

```
Program Esepl;  
Var i:integer;  
    s:real;  
    b:array[1..7] of real;  
Begin  
  for i:=1 to 7 do  
    Readln(b[i]);  
  for i:=1 to 7 do  
    begin  
      s:=s+b[i];  
      writeln(b[i]);  
    end;  
  writeln;  
  Writeln('S=',s);  
readln;  
End.
```

18-сурет. Бағдарлама құру терезесі.

```
C:\ Turbo Pascal
Turbo Pascal Version 7.1 Copyright (c) 1985
4
5
-2
-4
-3
5
1
4 .0000000000E+00
5 .0000000000E+00
-2 .0000000000E+00
-4 .0000000000E+00
-3 .0000000000E+00
5 .0000000000E+00
1 .0000000000E+00
S = 6 .0000000000E+00
```

19-сурет. Бағдарлама жауабын шығару терезесі.

Мұндағы I параметрін бүтін тип болу себебі, I параметрі массивтің элементтерінің саны, ал элементтер саны міндетті түрде бүтін болады. В массивінің типі нақты болу себебі, массивтің элементтері бүтін де, нақты да болуы мүмкін. S-қосындысы нақты тип болу себебі, массивтің типіне байланысты.

Массивпен қандай әрекеттер орындалса да циклдық операторлар арқылы орындалады. Қосынды мен массивті баспаға шығару амалдары операторлық жақшаларға алынып жазылады, себебі екі амалда да бір амал бірнеше рет қайтланады, сондықтан циклдың денесі болып табылады.

Бағдарламаны жібергенде экранда курсор пайда болады. Осы жерде массивтің элементтерін Enter пернесі арқылы енгізіледі. Соңғы элементті енгізгеннен кейін экранға массивтің енгізілген элементтері мен элементтердің қосындысы шығады.

CRT модулінің бірнеше процедуралары мен функциялары бар:

Random – функциясы 0-ден 1-ге дейінгі аралықтағы кездейсоқ сандар.

Randomize – процедурасы кездейсоқ сандар генераторының қорын өзгертеді.

1. 8 элементтен тұратын A массивінің ең үлкен элементін табу.


```

program es1;
const n=8;
type vector=array [1..n] of integer;
var a:vector;
I,max:integer;
begin
{бір өлшемді массивті енгізу алгоритмі}
    For i:=1 to n do
        Begin
            Writeln('a['I,'] =');
            Readln(a[i]);
        End;
{ең үлкенін табу}
    max:=a[1] ;
    for i:=2 to n do
        if a[i]>max then max:=a[i];
        writeln('max=',max);
readln;
end.

```

2. $a[i]=5i^2$ формуласы бойынша берілген 5 элементтен тұратын А массивінің жұп мәндерінің санын табу.

```

program es2;
const n=5;
var a:array [1..n] of integer ;
i, k:integer;
begin

```

```

For i:=1 to m do
  A[i]:=5*sqr(i);
  K:=0;
  for i:=1 to m do
    if a[i] mod 2=0 then k:= k+1;
    writeln('жұп мәндер саны=',k);
  readln;
end.

```

4.2. Екі өлшемді массив

Екі өлшемді массивтің бір өлшемді массивтен айырмашылығы айнымалылар қатар және бағана бойынша орналасады. Екі өлшемді массивтерде жол мен бағананың қиылысуында орналасқан әрбір элементтің орны сол жол мен бағана номерімен анықталады. Сондықтан екі өлшемді массивті сипаттау үшін екі индекс қолданылады. Екі индекстің арасы үтір арқылы бөлініп жазылады. Екі өлшемді массивтің қатар саны – i -мен, ал бағана саны – j -мен белгіленеді.

Екі өлшемді массивті сипаттаудың жалпы жазылуы түрі:

Массивтің аты: $\text{array}[1..n,1..m]$ of массив элементтерінің типі;

Мұндағы, n - массивтің қатар саны, m - массивтің бағана саны.

Мысалы: 1) $a(2,3)$ массиві берілген. Осы массивтің элементтерін енгізетін

және теріс элементтерін 1-ге айналдыратын бағдарлама құру. Нәтижесінде экранға нәтижеге шыққан массив шығуы керек.

```

Program prim1;

```

```

Var i,j:integer;

```

```

  A:array[1..2,1..3] of real;

```

```

Begin

```

```

  For i:=1 to 2 do

```

```

begin
    For j:=1 to 3 do
        Readln(a[i,j]);
    end;
For i:=1 to 2 do
begin
    For j:=1 to 3 do
        begin
            If a[i,j]<0 Then a[i,j]:=1;
            Writeln(a[i,j]);
        end;
    end;
end;
Readln;
End.

```

```

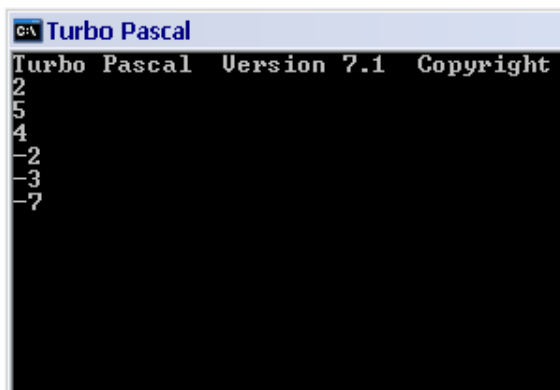
Turbo Pascal
File Edit Search Run Compile Debug Tools Options
[ ] 1.PAS
Program Esep1;
Var i,j:integer;
    a:array[1..2,1..3] of real;
Begin
    for i:=1 to 2 do
        begin
            for j:=1 to 3 do
                Readln(a[i,j]);
            end;
        for i:=1 to 2 do
            begin
                for j:=1 to 3 do
                    begin
                        if a[i,j]<0 then a[i,j]:=1;
                        writeln(a[i,j]);
                    end;
                end;
            end;
        readln;
    End.

```

20-сурет. Бағдарламаны құру терезесі.

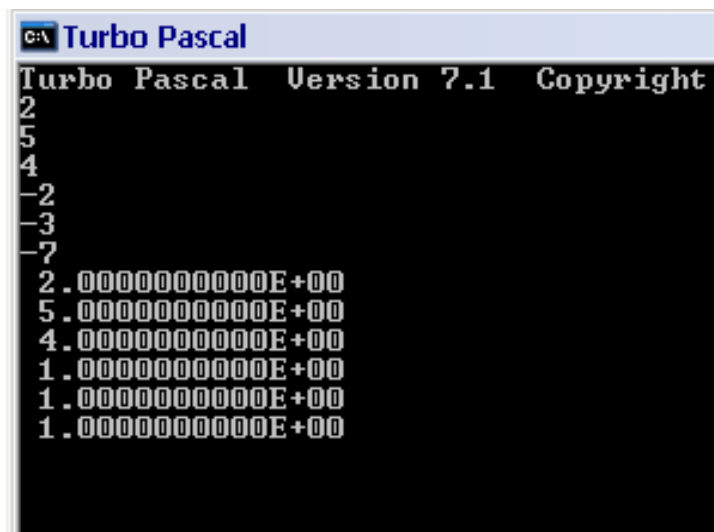
Мұндағы, *i*-сыртқы цикл, *j*-ішкі цикл болып саналады. *For i:=1 to 2 do, For j:=1 to 3 do* операторларынан кейін операторлық жақшаны ашылу себебі, бұл оператордан кейінгі орындалатын амалдар сыртқы циклға бағынышты (осы циклдың ішінде орындалады). Алдыңғы сабақта айтылғандай, егер де циклдық оператордан кейін бірнеше амалдар орындалса, онда ол амалдар операторлық жақшаларға алынып жазылады және олар циклдың денесі болып табылады.

Бағдарламаны жібергенде экранда курсор пайда болады. Екі өлшемді массивтің элементтері қатар бойынша енгізіледі.



```
с:\ Turbo Pascal
Turbo Pascal Version 7.1 Copyright
2
5
4
-2
-3
-7
```

21-сурет. Элементтерді енгізу.



```
с:\ Turbo Pascal
Turbo Pascal Version 7.1 Copyright
2
5
4
-2
-3
-7
2.000000000000E+00
5.000000000000E+00
4.000000000000E+00
1.000000000000E+00
1.000000000000E+00
1.000000000000E+00
```

22-сурет. Бағдарлама нәтижесі.

Элементтерді енгізуге түсініктеме жазу үшін

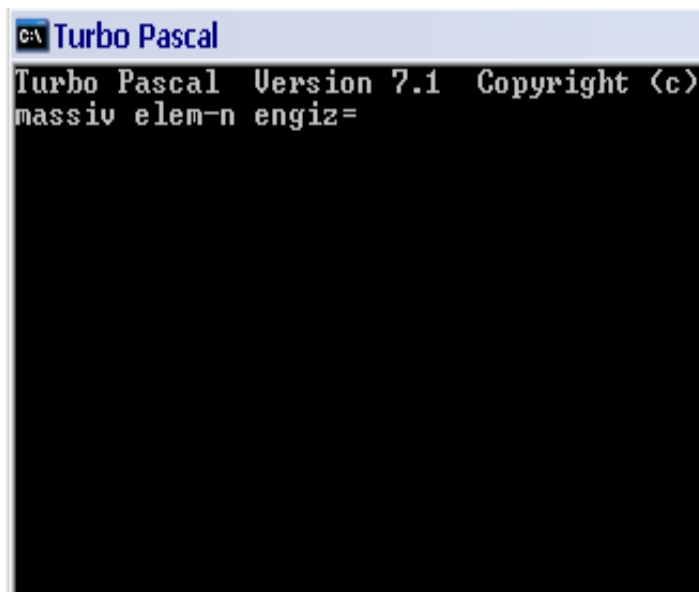
```
For i:=1 to 2 do
```

```
begin
```

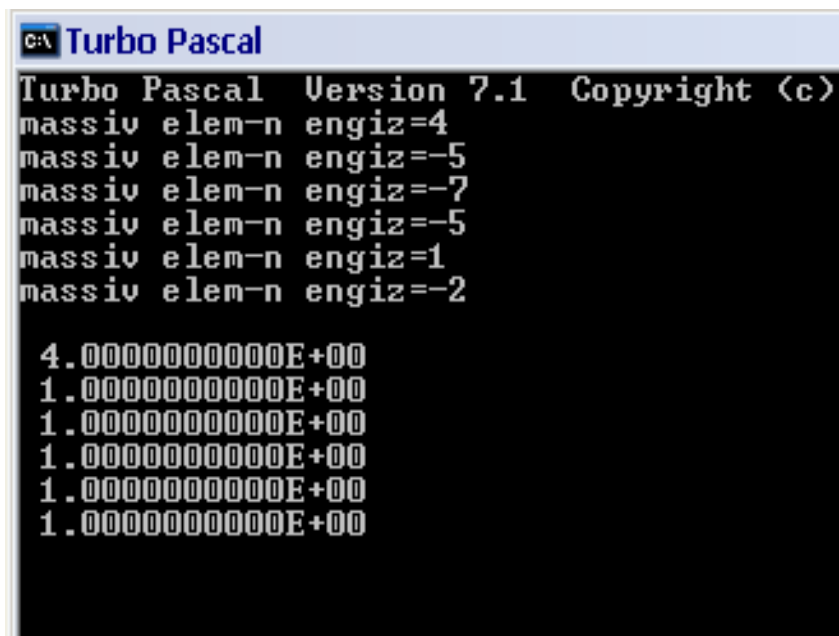
```
For j:=1 to 3 do
```

```
begin
```

```
write ('massiv elem-n engiz=')  
  
Readln(a[i,j]);  
  
end;  
  
end;
```



23-сурет. Элементтерді енгізу.



24-сурет. Бағдарлама нәтижесі.

2) Квадрат массив берілген (қатар және бағана саны тең). Осы массивтің әрбір бағанасында орналасқан элементтерінің қосындысын табатын және осы қосындыны K бір өлшемді массивінің элементтері ретінде алатын бағдарлама құру. Нәтижесінде,

берілген массив элементтері қатар және бағана бойынша орналасып шығуы және К массиві шығуы керек.

```
Program prim2;
```

```
Var i,j:integer;
```

```
    S:real;
```

```
    d:array[1..3,1..3] of real;
```

```
    k:array[1..3] of real;
```

```
Begin
```

```
    For i:=1 to 3 do
```

```
        begin
```

```
            For j:=1 to 3 do
```

```
                Readln(d[i,j]);
```

```
            end;
```

```
        For j:=1 to 3 do
```

```
            Begin
```

```
                S:=0;
```

```
                For i:=1 to 3 do
```

```
                    begin
```

```
                        s:=s+d[i,j];
```

```
                        k[j]:=s;
```

```
                    end; end;
```

```
        For i:=1 to 3 do
```

```
            Begin
```

```
                For j:=1 to 3 do
```

```
                    begin
```

```
                        write(d[i,j]:3:0);
```

```
                    end;
```

```
                writeln;
```

```
            end;
```

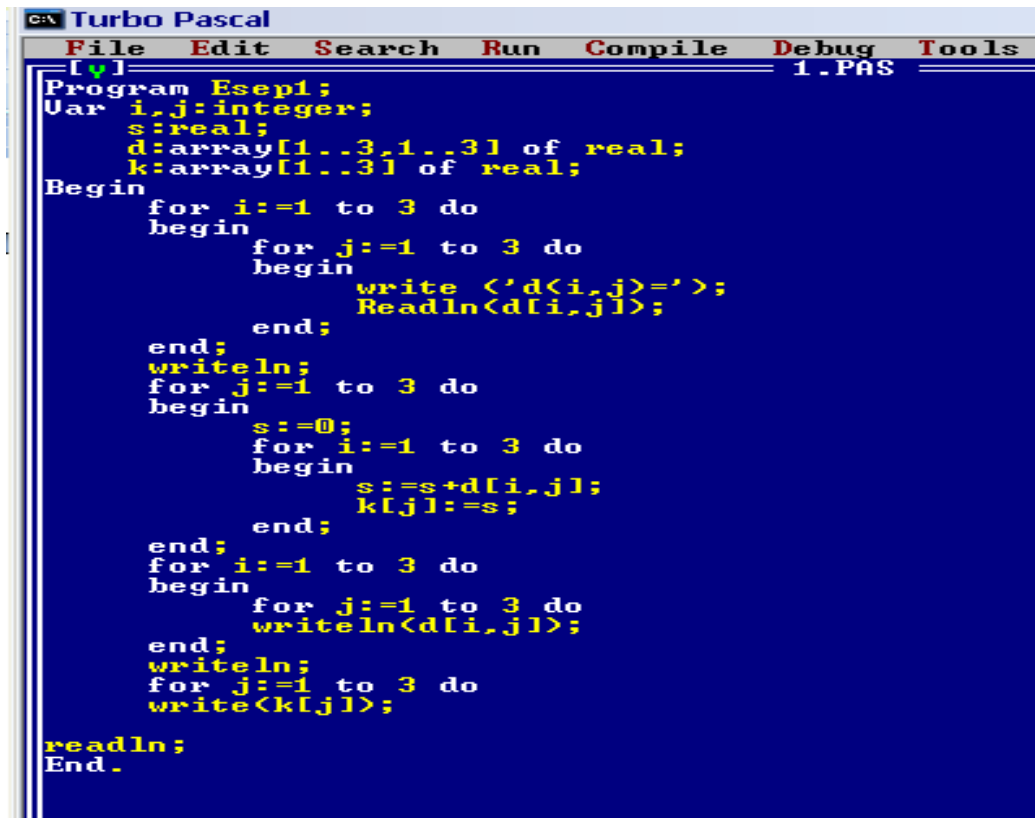
```
writeln;
```

```
For j:=1 to 3 do
```

```
Writeln(k[j]:3:0);
```

```
Readln;
```

```
End.
```



```
File Edit Search Run Compile Debug Tools
[ v ] 1.PAS
Program Esep1;
Var i,j:integer;
s:real;
d:array[1..3,1..3] of real;
k:array[1..3] of real;
Begin
  for i:=1 to 3 do
  begin
    for j:=1 to 3 do
    begin
      write ('d(i,j)=');
      Readln<d[i,j]>;
    end;
  end;
  writeln;
  for j:=1 to 3 do
  begin
    s:=0;
    for i:=1 to 3 do
    begin
      s:=s+d[i,j];
      k[j]:=s;
    end;
  end;
  for i:=1 to 3 do
  begin
    for j:=1 to 3 do
      writeln<d[i,j]>;
    end;
  end;
  writeln;
  for j:=1 to 3 do
    write<k[j]>;
  end;
  readln;
End.
```

25-сурет. Бағдарламаны құру терезесі.

Мұнда, бағанада орналасқан элементтердің қосындысын табу керек болғандықтан, j-параметрін сыртқы цикл, ал i-параметрін ішкі цикл ретінде аламыз. Егер де, $s:=0$ амалы сыртқы циклынан бұрын жазылса, онда әрбір бағананың қосындысын бір-біріне қосады (1-ші бағанадағы мәнді, 2-ші бағанаға қосады, т.с.с.). Бір өлшемді K массивінің параметр мәнін j деп белгілену себебі, қосынды бағана бойынша есептелінуіне байланысты (соңғы бағанадағы сандарды қосады).

```
c:\ Turbo Pascal
Turbo Pascal Version 7.1 Copyright (c)
d(i,j)=4
d(i,j)=-5
d(i,j)=2
d(i,j)=-1
d(i,j)=2
d(i,j)=4
d(i,j)=-2
d(i,j)=3
d(i,j)=-3_
```

26-сурет. Массив элементтерін енгізу.

```
c:\ Turbo Pascal
Turbo Pascal Version 7.1 Copyright (c) 1983,97 Borl
d(i,j)=4
d(i,j)=-5
d(i,j)=2
d(i,j)=-1
d(i,j)=2
d(i,j)=4
d(i,j)=-2
d(i,j)=3
d(i,j)=-3

  4.0000000000E+00
 -5.0000000000E+00
  2.0000000000E+00
 -1.0000000000E+00
  2.0000000000E+00
  4.0000000000E+00
 -2.0000000000E+00
  3.0000000000E+00
 -3.0000000000E+00

  1.0000000000E+00  0.0000000000E+00  3.0000000000E+00
```

27-сурет. Бағдарлама нәтижесі.

Массив элементтері нақты болғандықтан, нәтижеге шығатын элементтерді қысқартып жазу түрі:

```
For i:=1 to 3 do
```

```
  Begin
```

```
    For j:=1 to 3 do
```



```

writeln(d[i,j]:4:1);
end;
writeln;
For j:=1 to 3 do
Write(k[j]:4:0);

```

```

Turbo Pascal Version 7.1 Copyright (c)
d<i,j>=1
d<i,j>=2
d<i,j>=-4
d<i,j>=5
d<i,j>=-3
d<i,j>=2
d<i,j>=5
d<i,j>=-1
d<i,j>=2
1.0
2.0
-4.0
5.0
-3.0
2.0
5.0
-1.0
2.0
11 -2 0

```

28-сурет. Бағдарлама нәтижесі.

Екі өлшемді массивтің элементтеріне жеке-жеке амалдар қолдануға болады. Мысалы, төменде көрсетілген квадрат матрицаның негізі диагональ бойындағы, жоғарғы үшбұрышта орналасқан (негізгі диагональдың жоғарғы жағында орналасқан) және төменгі үшбұрышта орналасқан (негізгі диагональдың төменгі жағында орналасқан) элементтерімен әртүрлі амалдар орындауға болады. Мұнда, негізі диагональ бойындағы элементтерді - $i=j$;

жоғарғы үшбұрышта орналасқан элементтерді - $i < j$;

төменгі үшбұрышта орналасқан - $i > j$ болып белгіленеді.

$$A(3,3) = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$$

3) $N(3,3)$ екі өлшемді массив берілген. Осы массивтің негізі диагональ бойындағы элементтерінің қосындысын есептейтін, жоғарғы үшбұрышта орналасқан элементтерді 2-ге айналдыратын және берілген массивтің ең үлкен элементін табатын бағдарлама құру. Нәтижесінде, қосындысы, ең үлкен элемент, берілген массив элементтері мен жоғарғы үшбұрыштағы элементтерінің 2-ге айналған түрі қатар және бағана бойынша орналасып шығуы керек.

```

Program prim3;
Var i,j:integer;
    S,max:real;
    H:array[1..3,1..3] of real;
Begin
  For i:=1 to 3 do
  begin
    For j:=1 to 3 do
      Readln(H[i,j]);
    end;
  For i:=1 to 3 do
  Begin
    For j:=1 to 3 do
      write(H[i,j]:3:0);
      writeln;
    end;
    s:=0;
    max:=a[1,1];
    For i:=1 to 3 do
    Begin
      For j:=1 to 3 do
      begin
        if max<H[i,j] then max:=H[i,j];
        if i=j then s:=s+H[i,j];
        if i<j then H[i,j]:=2;
      end;
    end;
  writeln;

```

```

For i:=1 to 3 do
Begin
    For j:=1 to 3 do
        write(H[i,j]:3:0);
        writeln;
    end;
    writeln;
    Writeln(max:2:0, ' ',S:2:0);
End.

```

5. КӨМЕКШІ БАҒДАРЛАМАЛАР.

Көмекші бағдарламалар және олардың құрылысы.

Бағдарламалау барысында көптеген бір типті есептеулерге бірнеше бір-біріне ұқсас бағдарлама фрагменттерін қайта-қайта жазуға тура келеді. Осындайда артық дұмыс жасамай, барлық есептеулерді бір ғана фрагмент ішінде орындау үшін, көмекші бағдарламалар – процедуралар мен функциялар қолданылады. Мысалғы, бағдарламаларды орындау барысында бірнеше рет ұқсас есептеулер кездеседі, онда есептеу блогын бір-ақ рет көмекші бағдарламаға жазып қойып, бағдарламаның кез-келген нүктесінен көмекші бағдарламаны шақырып, осы блог арқылы есептеуді шақыруға болады. Сонымен көмекші бағдарламалар деп бағдарламаның орындалу барысында бірнеше рет шақырылап орындалып тұратын операторлар блогынан тұратын көмекші блогын айтамыз.

Көмекші бағдарламалар стандартты және пайдаланушылар көмекші бағдарламалар болып екіге бөлінеді. Стандартты функциялар мен процедуралар Turbo Pascal тілінің құрамында тұрақты системалық атпен аталып орындалады. Оларға арнаулы USES пайдаланушылар кітапханасында орналасқан – SYSTEM, CRT, GRAPH, GRAPH3, OVERLAY, PRINTER, TURBO3, TURBO VISION модульдерінің барлық функциялары мен процедуралары жатады. SYSTEM модулдерінің функциялары мен процедуралары Turbo Pascal бағдарламасымен бірге автоматты түрде іске қосылады, сондықтан оны пайдаланушылар бөлімінде көрсетпесе де болады. SYSTEM модулінен өзге модульдер

процедуралары мен функцияларын қолдану үшін бағдарламаның басында міндетті түрде олардың аттарын USES пайдаланушылар бөлімінде жазу керек, мысалы:

PROGRAM Көмек;

USES Graph, Crt;

// Graph, Crt модульдеріндегі стандартты процедуралар мен функцияларды бағдарламада пайдалану үшін ашу.

Стандартты функциялар мен процедуралардың көбісі бұдан бұрынғы бөлімдерде бағдарламалар құрауда пайдаланылды, мысалы:

Abs(x) – x-тің абсолюттік шамасын есептеу функциясы

Sin(x) – x-тің синусын есептеу функциясы

Random(x) – 0-ден x-ке дейінгі кездейсоқ сан шығару

ClrScr – экранды тазалау процедурасы.

Пайдаланушылар көмекші бағдарламалары деп бағдарламалаушыны өзі құрып, аты да өзіне ыңғайлы идентификатормен белгіленетін процедуралар мен функцияларды айтады. Олар PROCEDURE және FUNCTION қызметші сөздерімен жазылады.

Көмекші бағдарлама құрылымы:

<Көмекші бағдарлама қызметші сөзі> <көмекші бағдарлама тақырыбы>;

<Көмекші бағдарлама кездесетін идентификаторларды сипаттау бөлімдері>;

<Көмекші бағдарлама денесі>.

Көмекші бағдарламалардың жазылу тәртібі де негізі бағдарлама сияқты бірнеше бөлімдерден тұруы мүмкін,мысалы:

1) Тұрақтылар бөлімі;

2) Типтер бөлімі;

3) Айнымалылар бөлімі;

4) Операторлардан тұратын бағдарлама денесі.

Көмекші бағдарламаның аты идентификатормен белгіленеді. Бағдарламада бірнеше процедура мен функция болуы мүмкін. Сол сияқты, көмекші бағдарламаның да өзінің көмекші бағдарламасы болуы мүмкін. Оларды ішкі бағдарламалар деп атайды. Негізі бағдарламада сипатталған идентификаторларды көмекші бағдарламалардың ішіне де сол

сипатталған типпен қолдануға болады. Олармен белгіленген айнымалылар ауқымды деп атайды. Ал көмекші бағдарламада сипатталған идентификаторларды тек өзінің ішкі бағдарламасында ғана қолдануға болады. Ол айнымалыларды жергілікті деп атайды.

Көмекші бағдарлама өздерінен орындалмайды,оның орындалу үшін негізі бағдарламаның аты мен параметрлерін жазып шақырады,ол жерде көмекші бағдарламаны шақыру нүктесі деп атайды. Көмекші бағдарламалардың орындалу саны шақыру нүктелерінің санына байланысты болады.

Егер негізі бағдарламада бір шақыру нүктесі болмаса,онда көмекші бағдарлама орындалмайды. Бағдарлама орындалу барысында алдымен негізі бағдарлама орындалады да, керек кезде көмекші бағдарлама шақырылады. Көмекші бағдарламаны шақыру нәтежесінде негізі бағдарлама уақытша тоқтайды да, басқару бағдарламаға барады. Көмекші бағдарлама денесі, негізі бағдарламадан алған барлық деректерді толық өңдеп, орындалғаннан кейін, басқаруды шақыру нүктесінде өделген жаңа деректермен бірге қайтарып береді де,негізі бағдарлама орындалуын ары қарай жалғастырады.

Негізі бағдарламадан деректер қабылдау және көмекші бағдарламадан өңделіп жанарған деректерді негізгі бағдарламаға қайтару көмекші бағдарламалар параметрлері көмегімен атқарылады. Процедуралар мен функциялардың нақты және формальды параметрлері болады. Бағдарлама орындау барысында есептелген нақты деп,ал көмекші бағдарламалардың тақырыбында көрсетілген параметрлер формальді деп аталады. Нақты және формальды параметрлердің типі,саны,реттілігі сәйкес болуы тиіс. Олар бірдей білгіленуі де мүмкін. Нақты параметрлер тұрақтылардан, айнымалылардан және өрнектерден тұруы мүмкін. Кей жағдайларда көмекші бағдарламаның формальды параметрлерін көрсетпесе де болады.

6. ФАЙЛ. МӘТІНДІК ФАЙЛДАРМЕН ЖҰМЫС.

Дискіге жазылатын мәліметтер файлын және дискідегі файл жазылатын орынды **файл** деп атайды. Файлмен жұмыс жасауға арналған бағдарламаны **бағдарламалық файл** деп атайды. Файлды оқып, экранға шығару үшін қолданылатын операторлар болады.

Файлмен жұмыс алдында оның атын файлдық айнымалымен байланыстырып алу үшін *Assign* операторы қолданылады. Жалпы жазылу түрі:

Assign(x,n);

мұндағы, x-файлдық айнымалы, n-ашылатын файлдың аты.

Бағдарлама орындалу кезінде ашылатын файлдың атын файлдық айнымалыға береді. Бағдарламада ашылған файлға қолданған файлдық айнымалыны екінші бір файлды ашуға Assign операторында қолдану үшін міндетті түрде бірінші ашқан файлды жабу керек немесе екінші файлдың файлдық айнымалысын басқа идентификатормен белгілеу керек.

Файлды жабу үшін *Close* операторы қолданылады.

Close(x);

x-файлдық айнымалы.

Assign операторының көмегімен файлдың атын файлдық айнымалымен байланыстырғаннан кейін, олармен көптеген амалдар жүргізуге болады.

Дискіден бос орын ашып, жаңа файл жазуға дайындау үшін стандартты *Rewrite* операторы қолданылады. Оператордың орындалу нәтижесінде жаңа бос файл ашылып арнаулы файлдық сілтеме-көрсеткіш файлдың басына барып орналасады да, әрі қарай енгізілетін мәліметтер жаңа файлға сол жерден бастап енгізіледі. Бұл процедураның көмегімен дискіде бұрыннан бар файл ашылып жазылға жағдайда, ескі файлда, мәліметтер өшіріледі де орнына жаңа мәліметтер жазылады.

Rewrite(x);

x-дискіге жазылатын файл аты

Жаңа файлға мәліметтер Read және Write енгізу-шығару операторлары көмегімен толтырылады.

Write('мәтін'); - файлға жазылатын мәтіндік мәлімет

Read(y) - y-файлға жазылатын айнымалы

Енгізілген мәліметтер файлда сақталып жазылу үшін Write операторы қолданылады.

Write(x,y); - Y айнымалысының мәні X файлында сақталады

Дискідегі файлды ашу үшін стандартты *Reset* операторы қолданылады. Оператор дискідегі файлды оқуға дайындап ашады да, арнаулы файлдық сілтеме-көрсеткішті файлдың басына апарып орналастырады. Егер файлдық айнымалы дискіде жоқ файлды қабылдаса, онда бағдарлама орындалуы тоқтайды.

Reset(x);

x – дискіден ашылатын файл аты

Ашылған файлдағы айнымалылар мәндерін оқу үшін **Read** операторы қолданылады.

Read(x,y); - x-ашылған файл, y-файлдағы айнымалы

Файлдағы айнымалылардың мәндерін экранға шығару үшін **Write** операторы қолданылады.

Write(y); -y-файлдағы айнымалы

Ашылған файлға жаңа мәліметтер **Read** және **Write** операторлары көмегімен жазылады.

Мәтіндік файлдың соңына жаңа мәліметтер енгізу үшін **Append** операторы қолданылады. Оператордың орындалу нәтижесінде дискідегі файл ашылады да, арнаулы сілтеме-көрсеткіш файлдың ең соңына барып орналасады. Әрі қарай файлдың соңына жаңа мәліметтер **Reset**, **Rewrite** операторлары көмегімен ашылған файлдардағыдай енгізіледі.

Append(x);

x - text-типті файлдық айнымалы

Файлдың ішкі буферін тазалап, соңғы енгізілген мәліметтердің дискіде сақталуын қамтамасыз ету үшін **Flush** операторы қолданылады. Бұл операторды тек **Rewrite** немесе **Append** операторларының көмегімен ашылған мәтіндік файлдарда ғана қолданады.

Flush(x);

x - text-типті файлдық айнымалы

Файлдың көрсетілген компонентінен бастап соңына дейін өшіру үшін **Truncate** операторы қолданылады. Бұл оператор типі анықталған файлдарда қолданылады.

Truncate(x);

x-типі анықталған файлдық айнымалы

Дискідегі кез келген ашылмаған файлдың атын өзгерту үшін **Rename** операторы қолданылады.

Rename(x,New);

x-файлдық айнымалы, New-файлдың жаңа аты

Дискідегі кез келген файлды өшіру үшін *Erase* операторы қолданылады.

Erase(x);

дискіден x файлын өшіру

7. ЖОЛДЫҚ ҚАТАРЛАР.

Бағдарламада кездесетін әртүрлі символдардан тұратын тізбектерді жолдық қатарлар деп атайды. Өрнектерде жолдық қатарлар апострофқа (' ') алынып жазылады.

Turbo Pascal – да жолдық қатарларға компьютердің жадынан 0 байттан 255 байтқа дейін орын беріледі (жолдық қатар ұзындығына - 255символ, ұзындықтың мәніне – 1 символ).

Жолдық қатарлардан тұратын идентификаторлар String типі арқылы сипатталады. Идентификатордың типінен кейін тік жақшаның ішінде осы идентификаторлардың ұзындығының мәні неше символдан тұруы мүмкін екені көрсетіліп кетеді. Егер бұл мән көрсетілмесе, онда жолдық қатар ұзындығына компьютердің жадынан 255 символға дейін орын беріледі.

Жолдық қатарлы идентификаторларды типтер бөлімінде (TYPE) немесе бірден айнымалылар бөлімінде (VAR) сипаттауға болады. Сонымен қатар программалауда жолдық қатарлар тұрақты шамалар ретінде де қолданылады (таңдау селекторы тұрақтысынан басқа).

Жолдық қатарлардың сипаттау мысалдары:

PROGRAM Katarlar;

CONST S = 'Қайсар'; - {S тұрақтысы Қайсар тіркесіне тең}

TYPE Gymn = String [10];

VAR X: String [17] - {X айнымалысының ұзындығы 17 символға дейін }

B: String; - {B айнымалысының ұзындығы 255 символға дейін }

C: Gymn; - {X айнымалысының ұзындығы 10 символға дейін }

7.1. Жолдық өрнектермен амалдар.

Операндалары жолдық қатар болып табылатын өрнекті **жолдық өрнектер** деп атайды. Идентификаторларға жолдық айнылалылардын мәнін беру үшін меншіктеу операторы қолданылады. Егер меншіктеу операторы орындалғанда, символдар саны сипаттамалар бөлімінде көрсетілген шамадан асып кетсе, онда идентификатор қатардың сол жағынан бастап көрсетілген шамаға тең болатын символдарды, ғана меншіктейді, ал оң жақтағы артық символдар алынып қалады.

Мысалы:

```
VAR X:String[4];
```

```
BEGIN
```

```
    X:='Gymnazy'    - {'azy' символдары алынып қалып, X-ке 'Gymn'          тіркесі  
                    ғана меншіктеледі}
```

```
    Writeln(X);
```

Жолдық қатарлармен біріктіру және салыстыру амалдарын орындауға болады. Біріктіру амалы бірнеше жолдық қатарды біріктіріп шығару үшін қолданылады. Біріктірілетін жолдық қатардың ұзындығы 255-тен аспауы тиіс

Мысалы:

```
A:='Менің';
```

```
B:='Қазақстанымның';
```

```
Writeln(A+' '+B)    - {Менің Қазақстанымның – тіркесі шығады}
```

```
X:='Тәуелсіздігіне';
```

```
Y:= '10 жыл';
```

```
Z:=X+' '+Y;        - {Z Тәуелсіздігіне 10 жыл-тіркесін меншіктейді де, осы тіркесті  
                    экранға шығарады}
```

```
Writeln(Z);
```

Жолдық қатарларды салыстыру амалы екі жолдық қатарды салыстыру үшін қолданылады. Қатарларды салыстыру солдан оңға қарай ең біріеші кездесетін бірдей емес символға дейін жүргізіледі. Егер қай жолдық қатардың бірдей емес символының ақпарат

ауыстыру стандартты кестесіндегі нөмірі үлкен болса, сол жолдық қатар үлкен деп есептелінеді. Егер жолдық қатардың ұзындығы мен барлық символдары сәйкес келсе, олар тең деп есептелінеді. Ал жолдық қатарлардың ұзындығы мен символдар сәкес келсе, онда үлкен болып ұзындығы үлкен жолдық қатар болып есептілінеді. Жолдық қатарлардың салыстыру амалдарының нәтижесі әрқашан бульдік тип болады.

'Pentium' < 'PENTIUM' {False}

'Duron' = 'Duron' {True}

'Celeron' <> 'Celeron' {False}

'Hewlett' <= 'Hewlett Packard' {True}

'Laser' >= 'Laserjet' {False}

Жолдық қатардан қандай да бір фрагмент көшіріп алу үшін Copy операторы қолданылады. Жалпы жазылу түрі:

Copy(a,n,m); - a жолынан n орнынан бастап m символын көшіріп алу.

Concat(a,b) – a жолына b жолын қосу.

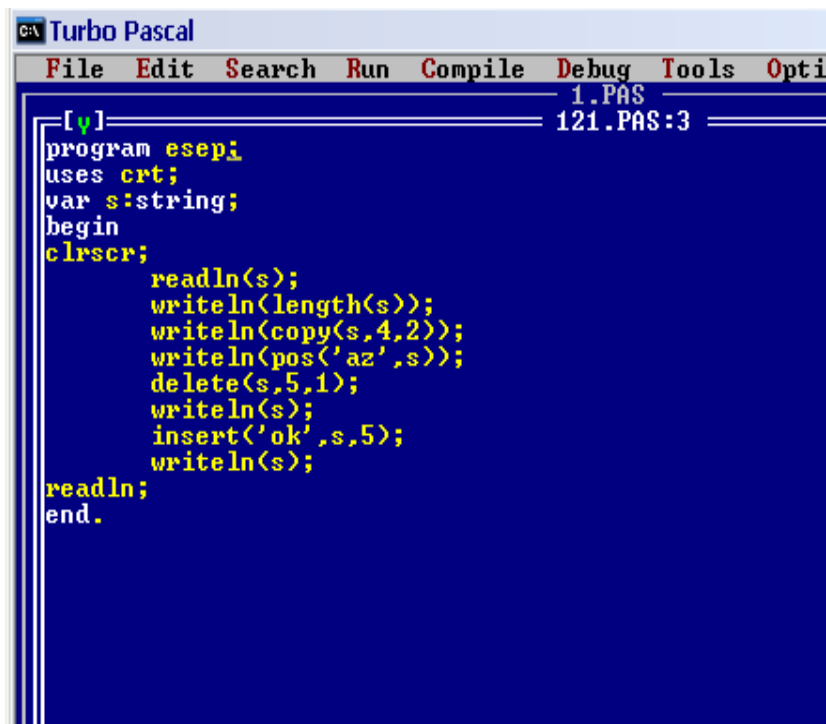
Pos(k,a) – k сөзі a жолының нешінші орнынан бастап орналасқанын анықтайды.

Delete(a,n,m) – a жолынан n орнынан бастап m символын өшіреді.

Insert(k,a,n) – k сөзін a жолының n орнынан бастап орналастырады.

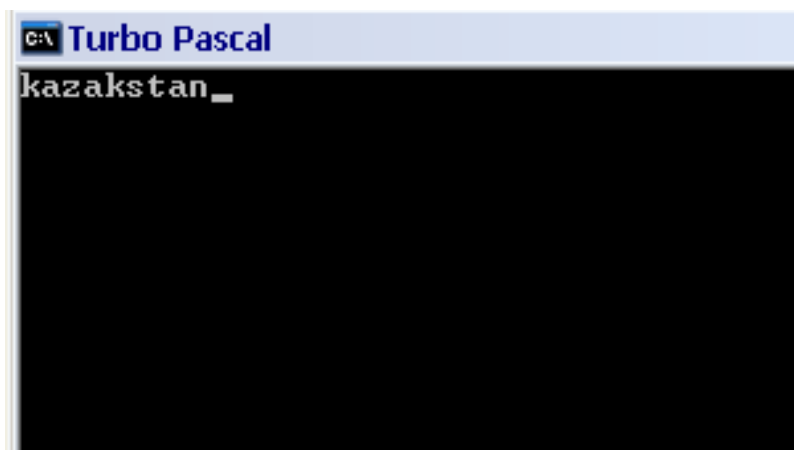
Str(x,a) – x санын a жолына айналдырады.

Val(a,x,c) – a жолын x санына айналдырады. x санын жолға айналдыру мүмкін болмаса с нөл емес мәнді қабылдайды.



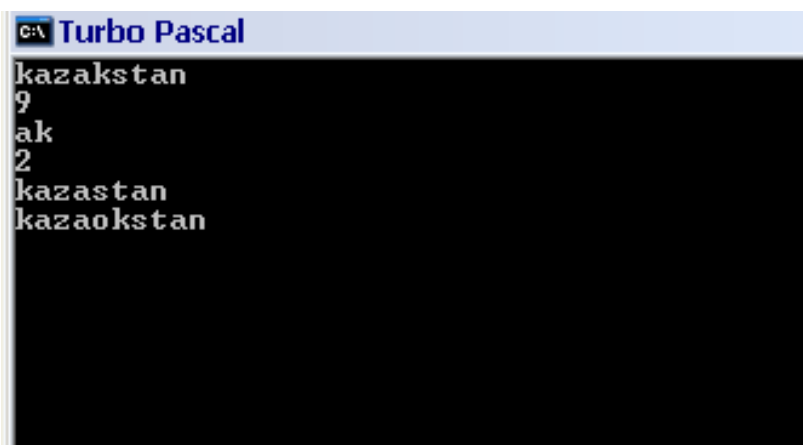
```
C:\ Turbo Pascal
File Edit Search Run Compile Debug Tools Opti
1.PAS
121.PAS:3
[ ]
program esep;
uses crt;
var s:string;
begin
clrscr;
  readln(s);
  writeln(length(s));
  writeln(copy(s,4,2));
  writeln(pos('az',s));
  delete(s,5,1);
  writeln(s);
  insert('ok',s,5);
  writeln(s);
readln;
end.
```

29-сурет. Бағдарлама енгізу терезесі.



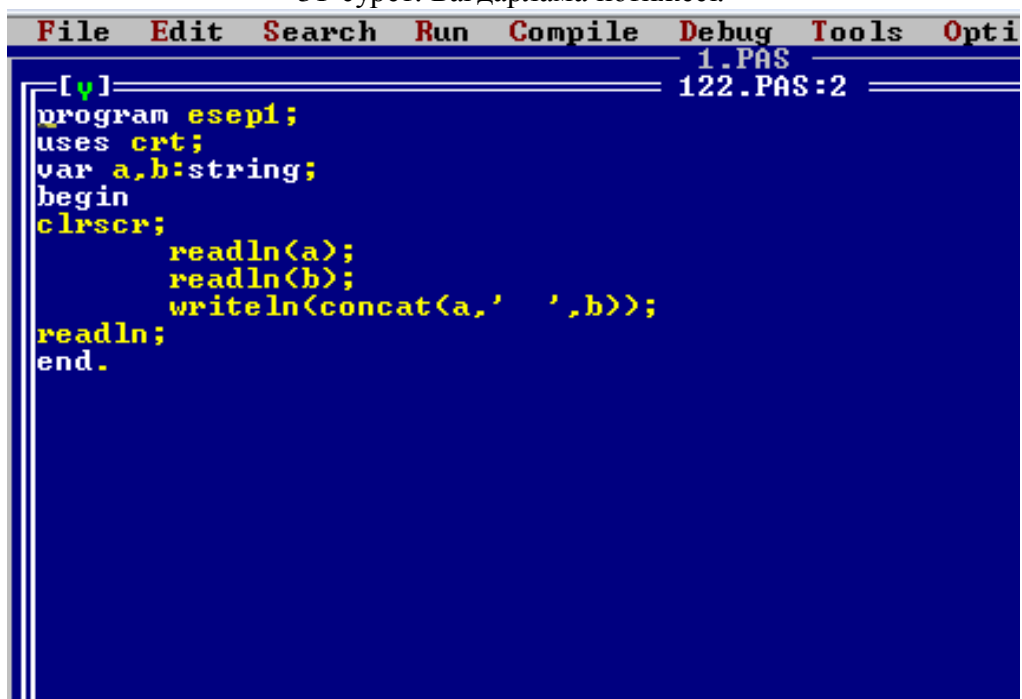
```
C:\ Turbo Pascal
kazakstan_
```

30-сурет. Символдарды енгізу.



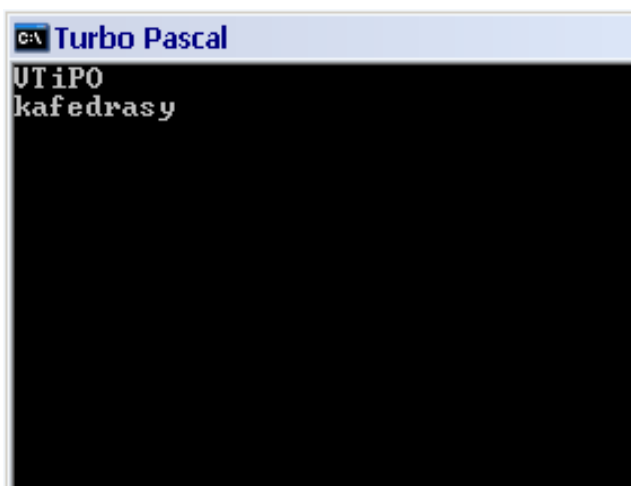
```
C:\ Turbo Pascal
kazakstan
9
ak
2
kazastan
kazaokstan
```

31-сурет. Бағдарлама нәтижесі.



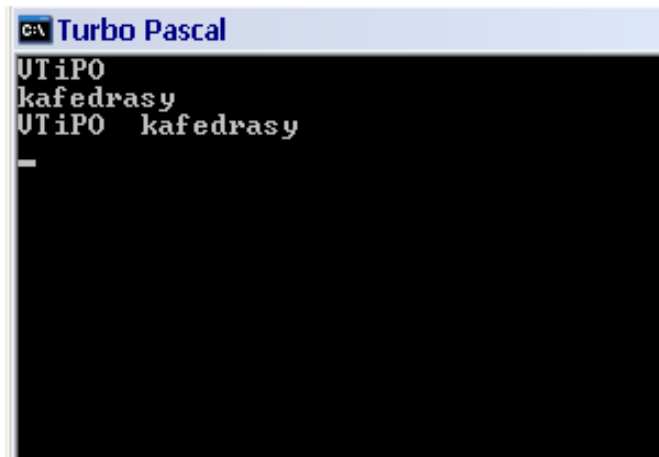
```
File Edit Search Run Compile Debug Tools Opti
1.PAS
122.PAS:2
[ ]
program esep1;
uses crt;
var a,b:string;
begin
clrscr;
      readln(a);
      readln(b);
      writeln(concat(a,' ',b));
readln;
end.
```

32-сурет. Бағдарлама енгізу терезесі.



```
CA Turbo Pascal
UTiPO
kafedrasy
```

33-сурет. Символдарды енгізу.



```
C:\ Turbo Pascal
UTiPO
kafedrasu
UTiPO kafedrasu
-
```

34-сурет. Бағдарлама нәтижесі.

7.2. Жиын. Символдық массивтермен амалдар орындау.

Алдыңғы тақырыптарда айтылғандай массив дегеніміз алгоритмдік тілде белгілі тәртіп бойынша орналасқан айнымалы шамалардың біріктірілген жиынтығы. Массив элементтері ретінде сандық мәндермен қатар символдық мәндерді де беруге болады. Және символдық мәндермен әртүрлі амалдар орындауға да болады.

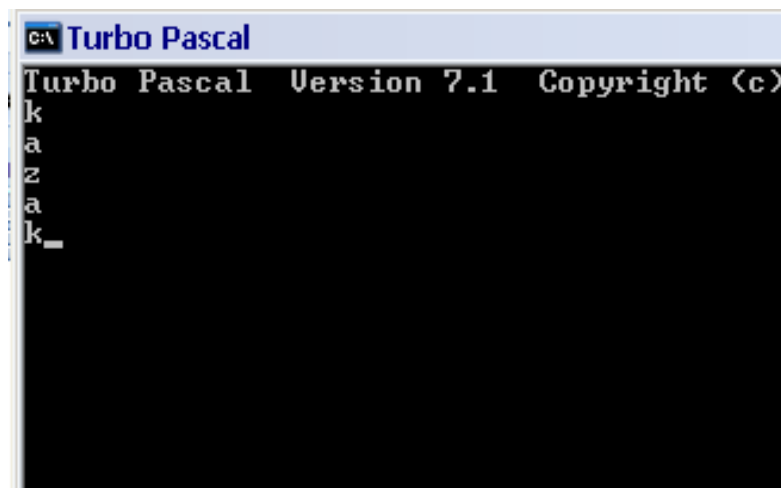
Мысалы: Бір өлшемді N массивінің элементтерін енгізу және баспаға шығаратын бағдарлама құру.

```
Program esep1;
Var i: string;
    N:array[1..5] of char;
Begin
    For i:=1 to 5 do
        Readln(n[i]);
    For i:=1 to 5 do
        Write(n[i]);
Readln;
End.
```



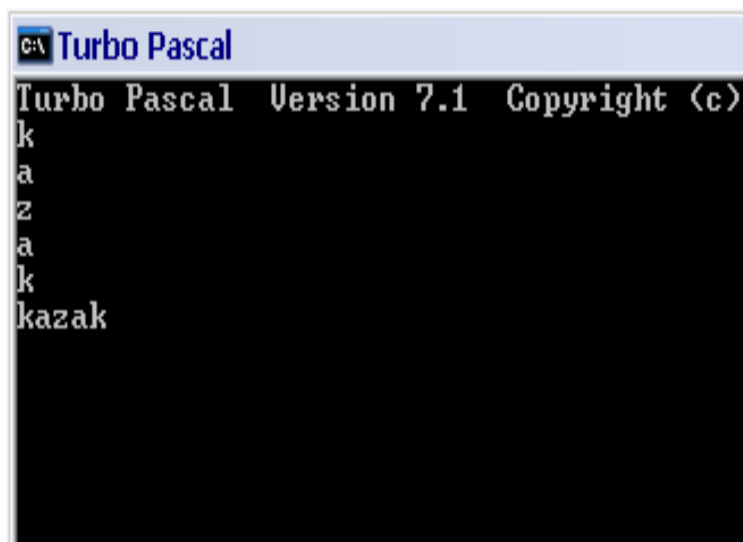
```
Program Esep1;
Var i:integer;
    n:array[1..5] of char;
begin
    for i:=1 to 5 do
        Readln(n[i]);
    for i:=1 to 5 do
        write(n[i]);
readln;
end.
```

35-сурет. Бағдарлама енгізу терезесі.



```
Turbo Pascal Version 7.1 Copyright (c)
k
a
z
a
k_
```

36-сурет. Символдарды енгізу.



```
C:\ Turbo Pascal
Turbo Pascal Version 7.1 Copyright (c)
k
a
z
a
k
kazak
```

37-сурет. Бағдарлама нәтижесі.

Сонымен қатар жолдармен де амалдар орындауға болады.

7.3. Жазу.

Паскальда түрлі типті элементтерден құралған күрделі типті мәліметтік берілгендер ерекше орын алады. Көптеген есептерде және кеңселік қағаздарында ведомостарды, құжаттарды, тізімдерді өңдеуге тура келеді. Өңдеу кезінде әртүрлі типтегі мәліметтерді бір топқа біріктіру қажеттігі туындайды. Мәліметтер тобымен жұмыс істеу үшін Паскаль тілінде жазулар түсінігі енгізілді. **Жазу** – әртүрлі типтегі мәліметтердің шектеулі жиынтығынан тұрады. Ал жазуды құрайтын элементтерді **жазу өрістері** деп атайды. Жазулар басқа мәліметтер сияқты сипаттамалар бөлімінде сипатталып, операторлар бөлімінде пайдаланылады. Жазудың сипатталуы:

type <тип атауы> = record

<өріс атауы>: <өріс типі>;

* * *

<өріс атауы>: <өріс типі>;

end;

var <жазу атауы>: <тип атауы>;

Мысалы: Бір студент туралы ақпарат (жазу) енгізу. Жазу 4 өрістен тұрады.

1-өріс: 5246 (студенттің шифры)

2-өріс: Маратов А. (аты-жөні)

3-өріс: қазақ (ұлты)

4-өріс: 1993 (туған жылы)

Мұндағы Record (жазу), end қызметші сөздері жазуды анықтайтын синтаксистік шектеулер. Record сөзі операторлық жақшаны ашады, end операторлық жақшаны жабады. Операторлық жақшаның ішінде жазудың элементтері сипатталады. Жазудың аты ретінде үтір арқылы ажыратылған атаулар тізімін пайдалануға болады.

Сипаттау алдында жазудың, өрістердің, типтің атауларын оқуға ыңғайлы етіп, түсінікті түрде таңдап алған жөн.

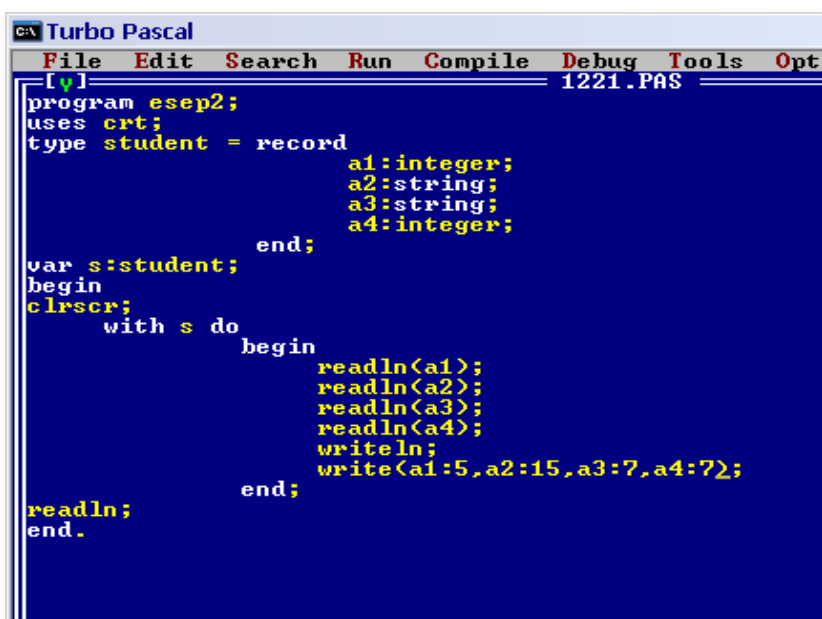
Басқа атаулар сияқты өрістің аты да бірнеше символдардан тұруы мүмкін. Сондықтан операторларда өріс айнымалыларын толық түрде жазу күрделі бағдарламаларды оқуға қиындық тудырады. Оны оңайлату үшін **With (қосу, біріктіру)** операторын ұолданады. Ол бағдарламада өрістік элементтерді өріс белгілері бойынша ғана өрнектеуге мүмкіндік береді. Жалпы жазылу түрі:

with <жазу атауы> do

begin

жазу өрістерінің белгілері енгізілген операторлар

end;



```

Turbo Pascal
File Edit Search Run Compile Debug Tools Opt
[ v ] 1221.PAS
program esep2;
uses crt;
type student = record
                a1:integer;
                a2:string;
                a3:string;
                a4:integer;
            end;
var s:student;
begin
clrscr;
    with s do
        begin
            readln(a1);
            readln(a2);
            readln(a3);
            readln(a4);
            writeln;
            write(a1:5, a2:15, a3:7, a4:7);
        end;
    readln;
end.
```

38-сурет. Бағдарлама енгізу терезесі.


```
C:\ Turbo Pascal
5246
Maratov A.
kazak
1993_
```

39-сурет. Символдарды енгізу.

```
C:\ Turbo Pascal
5246
Maratov A.
kazak
1993
5246      Maratov A.  kazak  1993_
```

40-сурет. Бағдарлама нәтижесі.

8. ПАСКАЛЬ ТІЛІНІҢ ГРАФИКАЛЫҚ МҮМКІНДІКТЕРІ. GRAPH МОДУЛІ.

8.1. Графикалық режим

Ақпаратты енгізу құралы болып монитор табылады. Монитордың екі режимі бар: мәтіндік және графикалық. Мәтіндік режимде экранда бейнеленген объект *символ* болып табылады. Қарапайым жағдайда алфавиттік цифрлы дисплей режимінде жұмыс жасайтын монитор экраны көлденең бойынша 80 символ және тігінен 25 символ, барлығы 2000 визуальды объекті бар. Мұндай режимде экранға тек қана қарапайым мәтінмен қатар графикалық бейнелерді де шығаруға болады. Бірақ бейнелердің сапасы төмен болады.

Мұндай бейнелердің сапасы жоғары болу үшін *графикалық режимге* өту керек. Графикалық режимде экран *пиксельдерге* бөлінеді.

8.2. Графикалық координаттар

Графикалық режимде экран X және Y өстерінен тұратын координаттық жүйеге бөлінеді. Координаттар жүйесінің бастапқы нүктесі экранның сол жақ жоғарғы бұрышында орналасады. X өсі экранның жоғарғы сол жағынан оңға қарай, ал Y өсі жоғарыдан төменге қарай орналасады.

TextBackGround(<түс>) – экран кеңістігін бояу;

TextColor(<түс>) – символдарды бояу;

OutTextXY(x,y,s) – s жолын (x,y) орнынан бастап экранға шығарады.

Мұндағы <түс> орнына түстің аты немесе оның коды енгізіледі.

Режимде пайдаланылатын түстер мен оларға сәйкес кодтар бар. Олар:

Black = 0 (қара)

Blue = 1 (көк)

Green = 2 (жасыл)

Cyan = 3 (көгілдір)

Red = 4 (қызыл)

Magenta = 5 (күлгін сары)

Brown = 6 (қоңыр)

LightGray = 7 (ашық сары)

DarkGray = 8 (күңгірт сұр)

LightBlue = 9 (ашық көгілдір)

Yellow = 14 (ашық сары)

White = 15 (ақ)

Мысалы:

TextBackGround(<blue>) – экран кеңістігін көк түске бояйды.

TextColor(<2>) – символдарды жасыл түске бояйды.

Экранның кез келген н.ктесінен бастап мәтінді жазуға болады. Нүктеге курсорды орналастыру операторы:

GotoXY(x,y);

Турбо Паскаль тілінде графикалық бейнелерді қалыптастыру үшін GRAPH модуль стандартты кітапханасы қолданылады. Графикалық жүйені іске қосу:

uses graph;

graph модулінің негізгі операторлары:

PutPixel (x,y,<түс>) – боялған нүкте орнатады.

SetbkColor (<түс>) – графикалық режимдегі экран кеңістігін бояйды. Бұл оператор бағдарламаға енгізілмесе, экран ағымдық қара немесе көгілдір түске боялып көрсетіледі.

SetColor (<түс>) – кескін түсін бояйды. Оператор енгізілмесе, кескін ақ түске боялады.

GetColor (<түс>) – SetColor операторы орындалғаннан кейін ағымдағы түсті қайта орнатады.

Line (x1,y1,x2,y2) – (x1,y1) нүктесінен (x2,y2) нүктесіне дейін кесінді сызады.

Circle (x,y,R) – орталық нүктесі (x,y), радиусы R болатын шеңбер сызады.

Rectangle (x1,y1,x2,y2) – сол жақ жоғарғы төбесі (x1,y1), оң жақ төменгі төбесі (x2,y2) болатын тік төртбұрыш сызады.

CloseGraph – графикалық режимді жауып, мәтіндік режимге өтеді.

Graph Result – жүйенің графикалық режимге дұрыс өтуін тексеру операторы.

Program mysal;

Uses graph;

Begin

 TextBackGround(<blue>);

 TextColor(<4>);

 GotoXY(25,30);

 Write ('студент');

End.

Зертханалық жұмыс №1

Тақырыбы: Алгоритм түсінігі. Блок схема. Алгоритм түрлері.

Жұмыстың мақсаты: Блок схеманың қолданылуы туралы алғашқы мағлұмат беру. Алгоритм түрлерімен танысу.

Қажетті құралдар: Дербес компьютер. ОЖ Windows XP, TurboPascal.

Теориялық қажетті материалдарға сілтеме: C:\ Алгоритмдік тілде бағдарламалау;

Есепті шешу үшін орындаушыға қажетті амалдардың тізбегін анық көрсететін ережелер жүйесін **алгоритм** деп атайды.

Алгоритмді жасау барысында оны 3 тәсілмен сипаттауға болады:

- 4) Табиғи тілде
 - 5) Блок схема түрінде
 - 6) Арнаулы тілде
- Қарапайым тілде әрбір адамға түсінікті етіп жазу *табиғи тілде жазу* деп аталады.

Блок схема деп әр түрлі фигуралар арқылы бағдарламада орындалатын әрекеттер мен олардың орындалу ретін көрсететін графикалық схеманы айтады.

Арнаулы тілде жазу деп ЭЕМ түсінетін тілде жазуды айтады.

Алгоритмнің 3 түрі бар:

- 4) Сызықтық
 - 5) Тармақталушы
 - 6) Циклдық
- Амалдардың бірінен кейін бірі орындалуын **сызықтық алгоритм** деп атайды.

Есепте шешуде барлық уақытта сызықтық алгоритм түрінде өрнектеуге болмайды. Амалдар тізбегін таңдау қандай да бір шартқа байланысты болса, онда ондай алгоритмді **тармақталушы алгоритм** деп атайды.

Кейбір амалдар бірнеше рет қайталанатын болса, онда ондай алгоритмді **циклдық алгоритм** деп атайды.

Тапсырмалар

1. Герон формуласы бойынша үшбұрыштың ауданын табу. Үшбұрыштың 3 қабырғасы белгілі (3 қабырғаны есепті шешуші пернетақтадан енгізу қажет). Ауданын табу үшін бірінші үшбұрыштың мына формула бойынша периметрін табамыз: $p = \frac{a+b+c}{2}$. Периметрін тапқаннан кейін, ауданын есептейміз. Ол төмендегідей формуламен жүреді: $s = \sqrt{p(p-a)(p-b)(p-c)}$. Осы есепті табиғи тілде және блок схема түрінде сызу қажет.
2. Екі санның үлкенін табу.
3. 1-ден 10-ға дейінгі сандардың қосындысын табу.

1. Жұмысты орындалу реті

I. Табиғи тілде жазылуы:

- 1) алгоритмнің басы

2) a,b,c мәндерін енгізу

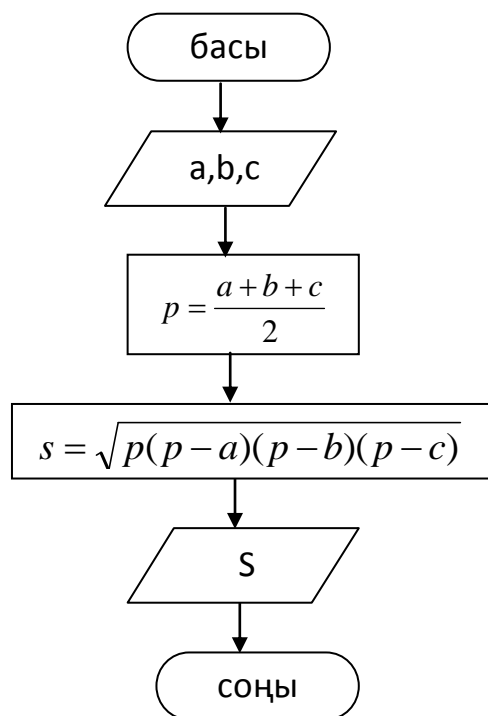
3) периметрдің формуласы бойынша P табу керек

4) ауданның формуласы бойынша S табу керек

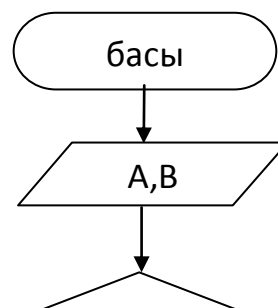
5) ауданның мәнін баспаға шығару керек (периметрдің мәнін де баспаға шығаруға болады)

6) алгоритмнің соңы

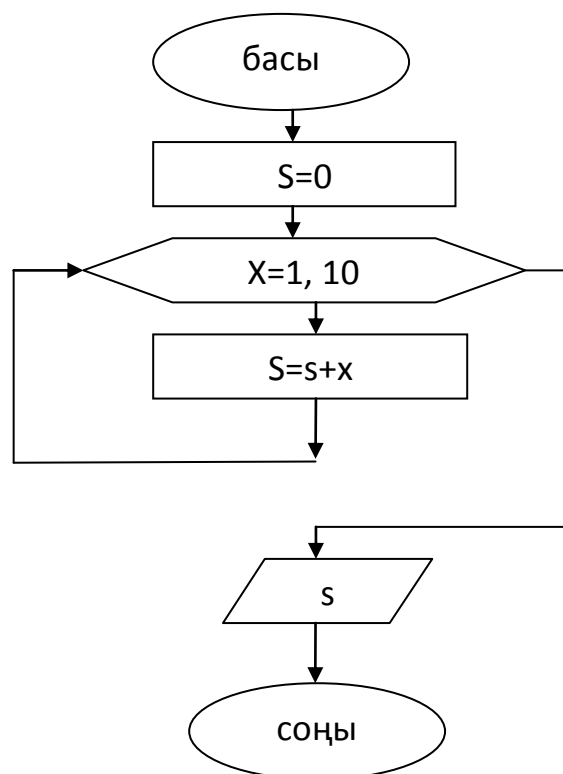
II. Блок схема түрінде көрсету:



2. Екі санның үлкенін табу.



3. 1-ден 10-ға дейінгі сандардың қосындысын табу.



ЖАТТЫҒУЛАР

1. Трапецияның a, b қабырғалары және h биіктігі берілген. Трапецияның ауданын анықтау қажет. $S = \frac{(a+b) \cdot h}{2}$.

2. $y = 5x^2 - 3x + 2$, мұндағы $x=0.5; -3; 1.2$

3. $y = \frac{x^2 + 2}{x + 5}$.

4. $V = \pi r^2 h$ формуласы бойынша цилиндрдің көлемін анықтау қажет. Радиусы мен биіктігі берілген.

5. Берілген x, y . Табу керек: $Z. z = 2x + \frac{3 + y^2}{x}$.

6. Төмендегі теңдеулерді Паскаль тілінде жазыңдар:

$$Y = (a-b)(a+b)$$

7. Арифметикалық өрнектердің мәндерін есептеу үшін меншіктеу операторларын қолданыңдар:

$$p = 2,6x^3 + 4,2x^2 - 1,8x$$

8. Қай меншіктеу операторлары қате жазылған:

$$a := b$$

$$a = b$$

$$a := b + 1$$

$b+1:=a$

9. Берілген екі санның қосындысын, көбейтіндісін есептеу программасын жазыңдар.

10. Үш санның арифметикалық ортасын тап.

Зертханалық жұмыс №2

Тақырыбы: Паскаль тілінің алфавиті. Паскаль тілінің құрылымы.

Мақсаты: Паскаль тілінің алфавитімен танысу. Стандартты функцияларды пайдалану. Паскаль тілінің құрылымымен танысу.

Қажетті құралдар: Дербес компьютер. ОЖ Windows XP, TurboPascal.

Теориялық қажетті материалдарға сілтеме: C:\ Алгоритмдік тілде бағдарламалау

Паскаль тілінде бағдарлама жазу үшін негізгі түсініктер қолданылады.

Паскаль тілінің негізін құратын алфавит бар. Тілдің алфавитіне кіретіндер:

7) A-Z-ке дейінгі және a-дан z-ке дейінгі латын алфавитінің 26 әрпі;

8) 0-9 дейінгі араб цифрлары;

9) Арнайы таңбалар: + (қосу), - (азайту), * (көбейту), / (бөлу), . (нүкте), , (үтір), : (қос нүкте), ; (нүктелі үтір), @, #, \$;

10) Салыстыру амалдары: > (үлкен), < (кіші), >= (үлкен немесе тең), <= (кіші н/е тең), <> (тең емес);

11) Арифметикалық және логикалық функциялар: DIV, MOD, NOT, AND, OR, XOR, IN, SHL, SHR;

12) әртүрлі тыныс белгілер:

{ } н/е (* *) – түсініктеме жазатын жақшалар;

[] – массив индекстерін және жиын элементтерін белгілеу;

() – айнымалылар тізімін белгілеу;

' ' – апостроф;

:= - меншіктеу белгісі.

Сонымен қатар стандартты функциялар, резервтелген сөздер, идентификаторлар қолданылады.

Паскаль тілінде бағдарлама құрылымы тақырыптан, блоктан және “.” (нүкте) белгісінен тұрады. Бағдарламаның тақырыбы бағдарлама атынан және параметрлер тізімінен тұрады:

Program бағдарлама - аты;

Паскальда блок екі бөлімнен тұрады: сипаттау және орындалу. Әрбір блок процедураның немесе функцияның хабарлануының бөлігі не болмаса бағдарламаның немесе модульдің бөлігі болады. Хабарлау бөлімінде хабарланған барлық идентификаторлар мен белгілер блок үшін жергілікті болып табылады.

Орындалу бөлімі бағдарлама операторларын қамтитын құрамды оператордан тұрады.

Сипаттау бөліміне модульдерді іске қосу, белгілерді, тұрақтыларды, типтерді, айнымалыларды, процедуралар мен функцияларды сипаттау сияқты бөлімдер енеді.

Тапсырмалар

Арифметикалық өрнектерді Паскаль тілінде жазу.

Жұмысты орындалу реті

1. $\frac{a+b}{c+d} = (a+b)/(c+d)$
2. $\frac{3(a+b)^2}{cd} = 3*((a+b)+(a+b))/c*d$
3. $A\sqrt{B} = A * SQRT(B)$
4. $\frac{1}{2} \sin 35^\circ = 1/2 * SIN(3.14 * 35/180)$
5. $a + \frac{b}{x} - \frac{2}{y} = a + b/x - 2/y$

ЖАТТЫҒУЛАР

Арифметикалық өрнектерді Паскаль тілінде жаз.

1. $\frac{\sin x}{\sqrt{\cos x}} - \frac{c}{d}$;
2. $\frac{\sqrt{A}\sqrt{B}}{C - D^{-3}}$;
3. $\frac{|x-1|}{\sqrt{x+20}} - \left(\frac{1}{3}\right)^2$;
4. Арифметикалық өрнектердің мәндерін есептеу үшін меншіктеу операторларын қолданыңдар:
 $p=2,6x^3+4,2x^2-1,8x$
5. Паскаль тіліндегі өрнектерді арифметикалық өрнек түрінде жаз.
 1. $A + \frac{B}{x} - \frac{2}{Y}$
 2. $\frac{7}{3} + SQR(EXP(x*x-1))$
 3. $7/3 + \sqrt{e^{x^2-1}}$

Зертханалық жұмыс №3

Тақырыбы: Сызықтық алгоритмді бағдарламалау.

Мақсаты: Енгізу және баспаға шығару операторларымен танысу. Сызықтық алгоритмде бағдарлама құрып үйрену.

Қажетті құралдар: Дербес компьютер. ОЖ Windows XP, TurboPascal.

Теориялық қажетті материалдарға сілтеме: C:\ Алгоритмдік тілде бағдарламалау

Сызықтық бағдарлама деп белгілі бір тәртіппен орындалатын бағдарламаны айтады. Сызықтық бағдарламаны құру үшін мынадай операторлар қолданылады:

Read, Write, меншіктеу операторы.

Read – айнымалыларды енгізу операторы. Бұл машинаның сұрауы бойынша пернетақтадан сандар енгізу үшін қолданылады.

Write – баспаға шығару операторы. Бұл оператор бағдарлама орындалғаннан кейін шыққан нәтижені экранға шығару үшін қолданылады. Егер де баспаға шығатын айнымалылар бірнешеу болса, онда олардың арасына үтір (,) қойылады.

Меншіктеу операторы (:=) сол жақтағы айнымалыға оң жақтағы өрнекті меншіктейді.

Ln жұрнағы курсорды жаңа жолдан бастайды.

Тапсырмалар

Алғашқы сабағымызда қарастырылған мысалға Паскаль тілінде бағдарлама құрайық. Герон формуласы бойынша үшбұрыштың ауданын табу.

Жұмысты орындалу реті

1. Program esep1;

Var a,b,c:integer;

p,s:real;

begin

read(a,b,c);

p:=(a+b+c)/2;

s:=Sqrt(p*(p-a)*(p-b)*(p-c));

write(s);

end.

2. Бағдарлама дұрыстығына көз жеткізіп, Ctrl+ F9 тетігін басу арқылы бағдарлама нәтижесін алыңыз.

3. Құрылған бағдарламаны сақтап, оқытушыға істелген жұмыс бойынша есеп беріңіз

ЖАТТЫҒУЛАР

Арифметикалық өрнектерді Паскаль тілінде жаз.

1. $y = 5x^2 - 3x + 2$, мұндағы $x=0.5; -3; 1.2$

2. $y = \frac{x^2 + 2}{x + 5}$, мұндағы $x=0.5; 12.5; -6$

3. $V = \pi r^2 h$ формуласы бойынша цилиндрдің көлемін анықтау қажет. Радиусы мен биіктігі берілген.

4. Шеңбердің радиусы берілген, оның ауданын $S = \pi r^2$ формуласы бойынша анықтау қажет.
5. Радиусы R-ге тең шеңбер берілген. Шеңбердің ұзындығын есептеу программасын құр.
($L=2\pi R$)

Зертханалық жұмыс №4

Тақырыбы: Тармақталушы алгоритмді бағдарламалау. Шартсыз өту операторы. Таңдау операторы.

Мақсаты: Тармақталушы алгоритмдерді Pascal ортасында бағдарламалауды қалыптастыру, өзіндік тапсырмаларды орындау. Есептерді шартсыз өту операторы көмегімен шығару. Таңдау операторымен танысу.

Қажетті құралдар: Дербес компьютер. ОЖ Windows XP, TurboPascal.

Теориялық қажетті материалдарға сілтеме: C:\ Алгоритмдік тілде бағдарламалау

Тапсырмалар

1. Екі санның үлкенін табатын бағдарлама құру.
2. $y=3a$ функциясының мәнін табу керек. Мұндағы a-ның мәні 5-ке тең. Егер 5-тен өзге сан берілген жағдайда a-ны қайта енгізу қажет

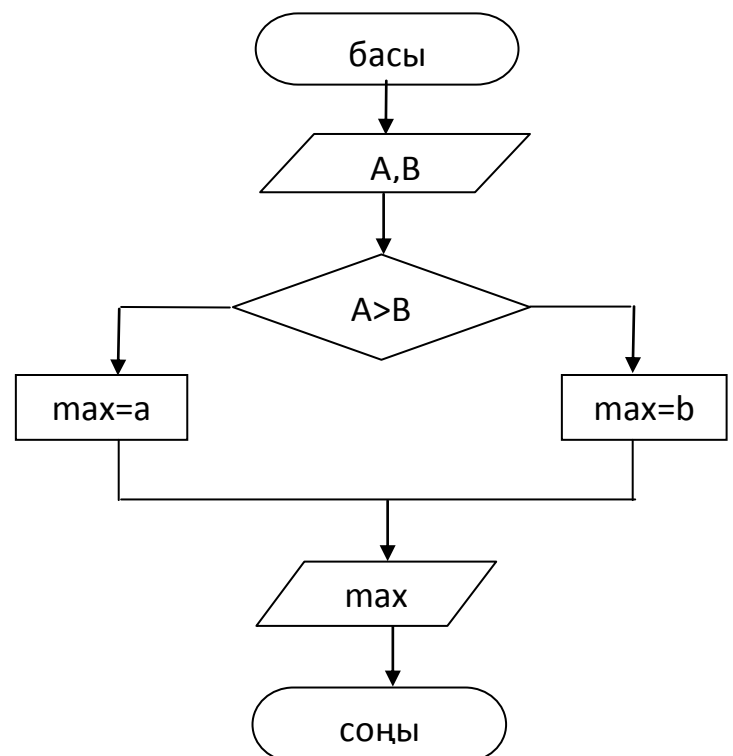
Жұмысты орындалу реті

1.

```

Program esep1;
Var a,b,max:integer;

begin
readln(a,b);
If a>b Then max:=a Else max:=b;
writeln(max);
end.
```



2. If операторы a және b мәндерін салыстырады. Егер шарт қанағаттандырса, Then операторынан кейінгі max:=a амалы орындалады. Ал шарт қанағаттандырмаса, Else операторынан кейінгі max:=b амалы орындалады. Амалдар орындалғаннан кейін max мәні нәтижеге шығады.
3. Бағдарлама дұрыстығына көз жеткізіп, Ctrl+ F9 тетігін басу арқылы бағдарлама нәтижесін алыңыз.
4. Құрылған бағдарламаны сақтап, оқытушыға істелген жұмыс бойынша есеп беріңіз

2. $y=3a$ функциясының мәнін табу керек. Мұндағы a-ның мәні 5-ке тең. Егер 5-тен өзге сан берілген жағдайда a-ны қайта енгізу қажет

Жұмысты орындалу реті

1. Program esep1;
 Var a,b,max:integer;

 Label 1;

 begin

 1:readln(a);

 If a=5 Then y:=3*a Else Goto 1;

 writeln(y);

 end.
2. Бағдарламаны тексеру кезінде 5 санынан басқа сан берілсе, қайта енгізуді сұрайды.
3. Бағдарлама дұрыстығына көз жеткізіп, Ctrl+ F9 тетігін басу арқылы бағдарлама нәтижесін алыңыз.
4. Құрылған бағдарламаны сақтап, оқытушыға істелген жұмыс бойынша есеп беріңіз

ЖАТТЫҒУЛАР

1. $y = \frac{x^2 + 5}{2a}$, мұндағы $a \neq 0$. Шарт орындалса, у-тің мәні баспаға шығу креке, ал шарт орындалмаса «шешімі жоқ» деген түсініктеме шығу керек.

$$2. \quad y = \begin{cases} -4x^2 + 8, & x < 0 \\ \sqrt{x^2 + 3x + 4}, & x > 1 \\ (x^2 + 3x + 4)^2, & 0 \leq x \leq 1 \end{cases}$$

$$3. \quad y = \begin{cases} 8x^3 - 4x + 2, & x = 5 \\ \sqrt{x^2 - x/2 + 3}, & x < 5 \\ x^2 - x/2 + 3, & x > 5 \end{cases}$$

4. Өрттен сақтандырғыш құрылғының программа моделін құрындар. Бөлмедегі температура (оның мәні пернетақтадан енгізіледі) 60градустан асқан жағдайда «Өрт басталды» деген дабыл қағылады.

5. $ax^2 + bx + c = 0$ квадрат теңдеудің түбірін табындар.

Зертханалық жұмыс №5

Тақырыбы: Циклдық алгоритмді бағдарламалау. Параметрлі цикл

Мақсаты: Циклдық алгоритмдерді Pascal ортасында бағдарламалауды қалыптастыру.

Қажетті құралдар: 1 Дербес компьютер. ОЖ Windows XP, TurboPascal.

Теориялық қажетті материалдарға сілтеме: C:\ Алгоритмдік тілде бағдарламалау

Тапсырмалар

-5-тен 5-ке дейінгі сандар берілген. Осы сандардың оң сандарының қосындысын табатын бағдарлама құру.

Жұмысты орындалу реті

1.

```
Program prim1;
```

```
  Var x:integer;
```

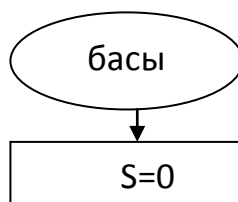
```
      S:real;
```

```
Begin
```

```
  S:=0;
```

```
  For x:=-5 to 5 do
```

```
    If x>0 Then S:=s + x;
```



WriteLn(S);

End.

2. Бағдарлама дұрыстығына көз жеткізіп, Ctrl+ F9 тегігін басу арқылы бағдарлама нәтижесін алыңыз.
3. Құрылған бағдарламаны сақтап, оқытушыға істелген жұмыс бойынша есеп беріңіз
4. Бұл бағдарламада x параметр мәні -5-тен 5-ке дейінгі сандарды қарастырады. Осы аралықтағы оң сандардың қосындысын тауып, баспаға шығарады. Қосындыны тапқан уақытта қандай санды 0-ге қосса да, сол санның өзі шығады. Сондықтан S -тің бастапқы мәнін $S=0$ деп аламызу. Осыдан кейін $x>0$ шарты тексеріледі. Егер шарт қанағаттандырса, $S=s + x$ амалы орындалады. Бұл формула бойынша орындалу былай жүреді:

Жауабы 15-ке тең болады.

ЖАТТЫҒУЛАР

1. Екі таңбалы жұп сандардың көбейтіндісін табатын бағдарлама құру.
2. 5-ке бөлінетін үш таңбалы сандардың тізімін экранға шығаратын бағдарлама құру.
3. $y = \prod_{x=-3}^6 2x + 3$
4. Кез келген n натурал санының көбейту кестесін шығаруға программа құр.
5. 1-ден 100-ге дейінгі сандардың арифметикалық ортасы анықтау
6. Үшке қалдықсыз бөлінетін k-ға дейінгі натурал сандардың көбейтіндісін табындар.
7. Алгоритм құрғанда бірінші сан 3 екендігін және әрбір келесі сан алдыңғысынан 3-ке артық болатындығын ескереміз.

Зертханалық жұмыс №6

Тақырыбы: Кейін және дейін циклдары.

Мақсаты: Кейін және дейін циклдарын қолданып есептер шығару.

Қажетті құралдар: Дербес компьютер. ОЖ Windows XP, TurboPascal.

Теориялық қажетті материалдарға сілтеме: C:\ Алгоритмдік тілде бағдарламалау

Цикл – бұл бағдарламаның орындалу кезінде бірнеше рет қайталанатын бөліктер.

While операторының құрамында операторының қайталанып орындалуын басқаратын өрнек болады. Дейін циклының жалпы жазылу түрі:

```
While <шарт> DO
```

```
    Begin
```

```
        операторлар;
```

```
    end;
```

Тапсырмалар

Алдыңғы сабақта қарастырған мысалды дейін және кейін циклдарымен шығарайық.

-5-тен 5-ке дейінгі сандар берілген. Осы сандардың оң сандарының қосындысын табатын бағдарлама құру.

Жұмысты орындалу реті

```
Program prim1;
```



```

Var x:integer;
    S:real;

Begin
    S:=0;
    x:=-5;
    While x<=5 do
        Begin
            If x>0 Then S:=s + x;
            x:=x+1;
        end;
    Writeln(S);

    End.

```

1. Бағдарлама дұрыстығына көз жеткізіп, Ctrl+ F9 тетігін басу арқылы бағдарлама нәтижесін алыңыз.
2. Құрылған бағдарламаны сақтап, оқытушыға істелген жұмыс бойынша есеп беріңіз

ЖАТТЫҒУЛАР

1. Екі таңбалы жұп сандардың көбейтіндісін табатын бағдарлама құру.
2. $y = \prod_{x=-3}^6 2x + 3$
3. $y = \frac{x}{3-x}$ функциясының мәнін табу. Мұндағы x-тің мәні - [-10,10].
4. Кез келген n натурал санының көбейту кестесін шығаруға программа құр.
5. Натурал n санының жазылуындағы тақ цифрлардың санын есептеңіз.
6. 1-ден 100-ге дейінгі сандардың арифметикалық ортасы анықтау
7. Үшке қалдықсыз бөлінетін k-ға дейінгі натурал сандардың көбейтіндісін табындар.

Зертханалық жұмыс №7

Тақырыбы: Массивтер. Бір өлшемді массив.

Мақсаты: Массив. Бір өлшемді массивке арналған есептер шығарып үйрену.

Қажетті құралдар: ДК /Pentium 3,4 Windows /, TurboPascal

Теориялық қажетті материалдарға сілтеме: C:\ Алгоритмдік тілде бағдарламалау

Алгоритмдік тілде белгілі тәртіп бойынша орналасқан айнымалы шамалардың біріктірілген жиынтығын **массив** деп атайды. Ал, массивті құрайтын айнымалы шамаларды *массивтің элементтері* деп атайды. Массивпен жұмыс жасау үшін оның әр элементінің мәні белгілі болуы тиіс. Массивтегі әр элемент номерін *элементтің индексі* деп атайды. Массивтің аты болады, атты латын алфавитіндегі А-дан Z-ке дейін әріптермен белгілейді. Паскаль тілінде массивтің индексі массив атынан кейін тік жақшаға [] алынып жазылады.

Массивтің екі түрі бар:

- 3) бір өлшемді (сызықтық)
- 4) екі өлшемді (матрица)

Тапсырмалар

7 элементтен тұратын бір өлшемді В массиві берілген. Осы массивтің элементтерін енгізетін және элементтерінің қосындысын табатын бағдарлама құру. Нәтижесінде экранға массивтің элементтері және элементтерінің қосындысы шығуы керек.

Жұмысты орындалу реті

```
Program prim1;
```

```
Var I:integer;
```

```
S:real;
```

```
B:array[1..7] of real;
```

```
Begin
```

```
For i:=1 to 7 do
```

```
  Readln(b[i]);
```

```
  S:=0;
```

```
  For i:=1 to 7 do
```

```
    begin
```

```
      S:=s+b[i];
```

```
      Writeln(b[i]);
```

```
    End;
```

```
  Writeln(s);
```

```
  Readln;
```

```
End.
```

1. Бағдарлама дұрыстығына көз жеткізіп, Ctrl+ F9 тетігін басу арқылы бағдарлама нәтижесін алыңыз.
2. Бағдарламаны жібергенде экранда курсор пайда болады. Осы жерде массивтің элементтерін Enter пернесі арқылы енгізіледі. Соңғы элементті енгізгеннен кейін экранға массивтің енгізілген элементтері мен элементтердің қосындысы шығады.
3. Құрылған бағдарламаны сақтап, оқытушыға істелген жұмыс бойынша есеп беріңіз

ЖАТТЫҒУЛАР

1. Бір өлшемді K массиві берілген. Осы массивтің теріс элементтерінің қосындысын табатын бағдарлама құру.
2. A массивінің ең үлкен элементін табатын бағдарлама құру.
3. 10 элементтер тұратын массив берілген. Осы массивтің элементтерінің арифметикалық ортасын табатын бағдарлама құру.
4. 10 элементтен тұратын бір өлшемді жиым элементтерінің қосындысын есептеңіз
5. Жиымның оң элементтерінің санын анықтау.
6. 10 элементтен тұратын жиым элементінің ең үлкен индекстерін анықтау
7. Реттелмеген жиым элементтерінің ішінен алдын ала берілген санға тең элементті іздеу. Алдымен 10 сан енгізіп, оларды жиымда сақтаймыз. Программа алдын ала берілген санды енгізгеннен кейін, жиым элементтерін осы санмен салыстырылады.
8. 10 элементтен тұратын A жиымының теріс элементтерінің ішіндегі ең үлкен элементін табатын программа құр.
9. Бір өлшемді A(10) жиымындағы теріс элементтердің ең үлкенін табыңдар.
10. Егер бір өлшемді жиымның барлық элементтері оң болса, онда «шарт орындалады», басқа жағдайда «шарт орындалмайды» деген хабарды басып шығарыңдар.
11. Бір өлшемді D массиві берілген. Осы массивтің оң элементтерінің санын табатын бағдарлама құру.
12. 10 элементтен тұратын бір өлшемді жиым элементтерінің қосындысын есептеңіз
13. Жиымның оң элементтерінің санын анықтау.
14. 10 элементтен тұратын жиым элементінің ең үлкен индекстерін анықтау
15. Реттелмеген жиым элементтерінің ішінен алдын ала берілген санға тең элементті іздеу.
16. Бір өлшемді A(10) жиымындағы теріс элементтердің ең үлкенін табыңдар.
17. Егер бір өлшемді жиымның барлық элементтері оң болса, онда «шарт орындалады», басқа жағдайда «шарт орындалмайды» деген хабарды басып шығарыңдар.
18. L массивінің ең үлкен және ең кіші элементтерін табатын бағдарлама құру.
19. F массиві берілген. Осы массивтің элементтерін өсу реті бойынша орналастыр.
20. Бір өлшемді массив берілген. Массивтің 5-тен үлкен элементтерін 1-ге айналдыратын бағдарлама құру.

Зертханалық жұмыс №8

Тақырыбы: Екі өлшемді массив.

Мақсаты: Екі өлшемді массивпен танысу. Екі өлшемді массивтің қатар және бағаналарында орналасқан элементтеріне амалдар қолдану

Қажетті құралдар: Дербес компьютер. ОЖ Windows XP, TurboPascal

Теориялық қажетті материалдарға сілтеме: C:\ Алгоритмдік тілде бағдарламалау

✍ - дәптерге жазып алу. ⓘ - мәліметтерді назар салып оқу. 🖥 - тапсырманы өз бетімен орындау.

① Турбо Паскаль бағдарламалау тілі бір өлшемді массивтермен қатар екі өлшемді және көп өлшемді массивтерді қолдануға мүмкіндік береді. Екі өлшемді немесе көп өлшемді массивтермен жұмыс істеу үшін, олар сипаттау бөлімінде көрсетілуі тиіс. Екі өлшемді массивтерді бағдарламалау арқылы кесте түрінде берілетін сандық есептеулер біршама ықшамдалады.

✍ **Екі өлшемді массивті Var бөлімінде сипаттаудың жалпы түрі:**

Var Массив атауы : array [a1..an, b1..bn] of элемент типі;

✍ **Екі өлшемді массивті type бөлімінде сипаттаудың жалпы түрі:**

type Массив типінің атауы = array [a1..an, b1..bn] of элемент типі;

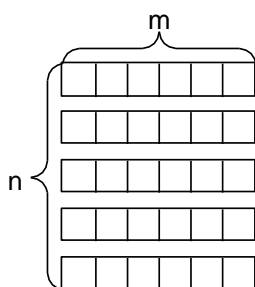
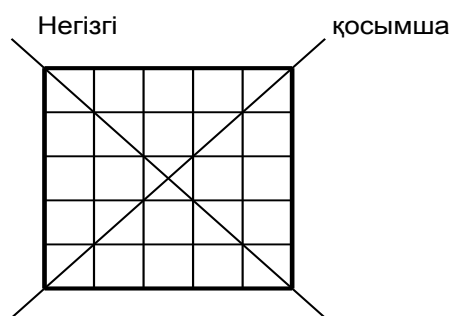
Var Массив атауы: массив типінің атауы;

Мұндағы a1..an, b1..bn – екі өлшемді массивтің көлемі;

a1 және an – массив қатарының алғашқы және соңғы мәні;

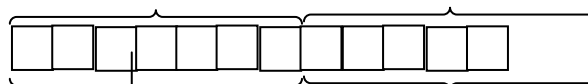
b1 және bn – массив бағанының алғашқы және соңғы мәні

| | | | | | |
|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| a ₁₁ | a ₁₂ | a ₁₃ | a ₁₄ | a ₁₅ | a ₁₆ |
| a ₂₁ | | | | | |
| a ₃₁ | | a _{ij} | | | |
| a ₄₁ | | | | | |
| a ₅₁ | | | | | |



{квадрат матрица}

Екі өлшемді массивтің компьютер жадында орналасуы



a_{ij} – i-інші жолдағы j-інші элемент

Екі өлшемді массивтің жекелеген элементі екі индексмен анықталады. Мысалы, 5-жолда 6-бағанда орналасқан ұяшық $A[5][6]$ немесе $A[5,6]$ деп белгіленеді.

☞ **Екі өлшемді массивті енгізу:**

For i:=1 to n do

For j:=1 to m do

Readln(a[I,j]);

☞ **Екі өлшемді массивті матрица түрінде шығару:**

For i:=1 to n do

Begin

For j:=1 to n do

Begin

Write(a[i,j]);

End;

Writeln;

End;

Екі өлшемді массивтерді өңдеудің негізгі алгоритмдері

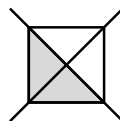
1) Қосындыны табу

S := 0;

for i := 1 to n do

for j := 1 to m do

S := S + a[i,j];



2) Негізгі диагональ астындағы

(негізгі диагональ бойындағы

S := 0;

for i := 1 to n do

for j := 1 to m do

If $i > j$ Then S := S + a[i,j];

элементтердің қосындысын табу
элементтермен қоса)

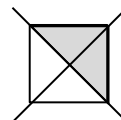
3) Негізгі диагональ үстіндегі элементтердің қосындысын табу (негізгі диагональ бойындағы элементтермен қоса)

S := 0;

for i := 1 to n do

for j := 1 to m do

If $i < j$ Then S := S + a[i,j];



4) Негізгі диагональ бойындағы элементтердің қосындысын табу

```

S := 0;
for i := 1 to n do
for j:=1 to m do
If i=j Then S := S + a[i,j];

```

5) Қосымша диагональ бойындағы элементтердің қосындысы

```

S := 0;
for i := 1 to n do
S := S + a[i,n-i+1];

```

6) Жол бойынша элементтердің қосындысын табу

```

for i:=1 to n do
begin
m[i]:=0;
for j:=1 to m do
m[i]:=m[i]+a[i,j];
end;
for i:=1 to n do
writeln(i, '→жол қосындысы = ',m[i]);

```

7) Баған бойынша элементтердің қосындысын табу

```

for j:=1 to m do
begin
f[j]:=0;
for i:=1 to n do
f[j]:=f[j]+a[i,j];
end;
for j:=1 to m do
writeln(j, '-бағанның қосындысы = ',f[j]);

```

✍ Мысал №1. В(3,3) екі өлшемді массивінің арифметикалық ортасын табу.

```

program Matrix1;
var a : array[1..3,1..3] of integer;
i, j, s : integer; sr:real;
begin
for i:=1 to 3 do
for j:=1 to 3 do
readln(a[i,j]); {Массив элементтерін енгізу}
s:=0;
for i:=1 to 3 do
for j:=1 to 3 do
s:=s+a[i,j]; {Қосындыны табу}
sr:=s/9; {Арифметикалық ортаны табу}
writeln('Арифметикалық орта=',sr:4:2);
readln end.

```

☞ Мысал №2. $A(3,3)$ екі өлшемді массивінің негізгі диагональ астындағы элементтерінің (диагональмен қоса) қосындысын табу және массивті баған түрінде шығару.

```
program Matrix2;
var a : array[1..3,1..3] of integer;
i, j, s : integer;
begin
  for i:=1 to 3 do
    for j:=1 to 3 do
      readln(a[i,j]);
s:=0;
  for i:=1 to 3 do
    for j:=1 to 3 do
      if i>=j then s:=s+a[i,j];
for i:=1 to 3 do
begin
for j:=1 to 3 do
begin
Write(a[i,j], ' ');
end;
writeln;
end;
      } {Баған түрінде шығару}
      writeln;
writeln (' Диагональ астындағы элементтер қосындысы=',s);
readln end.
```

ЖАТТЫҒУЛАР

1. Екі өлшемді D массиві берілген. Осы массивтің ең үлкен және ең кіші элементтерінің орынын табатын бағдарлама құру.
2. Екі өлшемді массив берілген. Массивтің 0-ге тең үлкен элементтерін -2-ге айналдыратын және элементтердің көбейтіндісін табатын бағдарлама құру.
3. Екі өлшемді массивтің әрбір бағанасында орналасқан
4. элементтерінің көбейтіндісін табатын және осы көбейтіндіні C бір өлшемді массивінің элементтері ретінде алатын бағдарлама құру.
5. 3 қатардан, 3 бағанадан тұратын екі өлшемді A массив берілген.
6. Осы массивтің әрбір бағанасында орналасқан элементтердің арифметикалық ортасын табатын бағдарлама құру.
7. Екі өлшемді T массиві берілген. Теріс элементтерін 1-ге, оң элементтерін -1-ге айналдыратын бағдарлама құру. Нәтижесінде, берілген массив пен нәтижеге шығатын массив элементтері қатар және бағана бойынша орналасып шығуы керек.
8. Екі өлшемді $a(3,3)$ массивінің негізгі диагоналында орналасқан элементтердің қосындысын табу;
9. Бүтін сандардан тұратын 4×4 матрицасы берілген. Матрицаның бас диагоналінің элементтерін нөлмен алмастыру керек. Экранға бастапқы және өңделген матрицаны шығарыңдар.
10. Бүтін сандардан тұратын екі өлшемді $A(4,4)$ жиымының оң элементтерінің арифметикалық орта мәнін және нөлге тең элементтерінің санын табыңдар.

Зертханалық жұмыс №9

Тақырыбы: Массивті сұрыптау алгоритмдері.

Мақсаты: Массивтерге сұрыптау алгоритмдерін қолдану.

Қажетті құралдар: Дербес компьютер. ОЖ Windows XP, TurboPascal.

Теориялық қажетті материалдарға сілтеме: C:\ Алгоритмдік тілде бағдарламалау

Массивті әр түрлі белгілері бойынша реттеу үшін (өсу, кему реті бойынша және т.б.) сұрыптау алгоритмдері қолданылады. Бұл алгоритм бойынша массивтегі элементтер сұрыптау шарттарына байланысты орнын ауыстырады. Сұрыптаудың бірнеше алгоритмдері белгілі, бірақ біз тек негізгілерімен ғана танысамыз.

1) Разрядті алмасу арқылы сұрыптау (“көбіршікті” сұрыптау).

n элементтен тұратын вектор ($n-1$) рет қаралады. Әрбір қаралған кезде біріншісінен бастап көршілес екі элемент салыстырылады. Егер келесі элемент алдындағысынан үлкен болса, олар орнын ауыстырады. Әрбір қаралғаннан кейін кезекті үлкен элемент өз орнына тұрады. (*өсу реті бойынша сұрыптау*).

{Негізгі алгоритм}

```
For k:=1 to n-1 do
  For i:=1 to n-k do
    If a[i]>a[i+1] then
      begin
        R:=a[i];
        a[i]:=a[i+1];
        a[i+1]:=R;
      end;
```

{Өзгертілген алгоритм}

```
k:=1;
Repeat
  Flag :=False;
  For i:=1 to n-k do
    If a[i]>a[i+1] then begin
      R:=a[i]; a[i]:=a[i+1];
      a[i+1]:=R;Flag:=True
    end; Inc(k);
Until Not Flag Or (k=n)
```

2) Таңдау арқылы сұрыптау

Массив ішінен ең кіші элемент табамыз да, оны біріншімен орнын ауыстырамыз. Қалған элементтердің ішінен (яғни екіншісінен бастап) қайтадан кішісін тауып, оны екіншімен орын ауыстырамыз. Осылайша бұл іс-әрекеттер соңына дейін қайталана береді. Ең соңғы элемент ең үлкен болады.

```
For i:=1 to n-1 do
  Begin
  min:=i;
  for j:=i+1 to n do
  if a[j]<a[min] then
  min:=j;
  r:=a[min];
  a[min]:=a[i];
  a[i]:=r;
  End;
```

ЖАТТЫҒУЛАР

1. Сегіз төреші жарысқа қатысқан командаға келесі бағалар қойды: 9.2, 9.4,9.6,9.3, 9.5, 9.4, 9.1, 9.3. Орташа бағаны және орташа бағадан асатын неше баға бар екендігін анықтау қажет.
2. Факультетте 15 группа бар. Әр группадағы студенттер саны келесідей: 25,23,18,16, 22, 19, 20, 21, 24, 17. Егер әр группаның 60% қыздар құраса, әр группадағы қыздар саны қанша?
3. A(3) және B(3) бір өлшемді массивтері берілген. $C_i=A_i + B_i$ теңдеуімен анықталатын C(3) массивін және оның жұп сандарының қосындысы мен теріс сандарының нешеу екендігін анықтау керек.
4. A(10) массивінің теріс элементтерінің қосындысын есептейтін бағдарлама құру.
5. Факультетте 15 группа бар. Олардың саны келесідей: 12, 15, 25,23,18,16, 13, 22, 19, 20, 21, 24, 17, 14, 10. Бюджеттік негізде оқитын студенттер саны әр группаның 20% құрайды. Әр группадағы бюджеттік негізде оқитын студенттер санын анықтау керек.
6. B(7) массивіндегі теріс элементтерді нольмен алмастырып шығаратын бағдарлама құру. Бастапқы және соңғы массивті экранға шығару.
7. Класта он оқушы бар. Олардың физикадан алған бағаларын енгізіп, озат оқушылардың және нашар баға алған оқушылардың санын анықтау керек.
8. 5 спортсменнің бағалары берілген. 100 балл алған спортсменнің фамилиясын шығаратын бағдарлама құру.
9. Берілген массивтегі P санынан асатын элементтердің нешеу екендігін анықтау.
10. C(12) массиві берілген. Массив элементтерінің тақ сандарының ең кішісін анықтайтын бағдарлама құру. Массив элементтері нақты сандар.

Зертханалық жұмыс №10

Тақырыбы: Процедуралар мен функциялар;

Мақсаты: Процедуралар мен функциялар қатысты берілген есептерді шешіп үйрену;

Қажетті құралдар: ДК /Pentium 3,4 Windows /, TurboPascal

Теориялық қажетті материалдарға сілтеме: C:\ Алгоритмдік тілде бағдарламалау

Тапсырмалар

$Z = \frac{a^5 + a^{-5}}{2a^7}$ өрнегін есептеу керек, мұнда a-берілген нақты сан

Жұмысты орындалу реті

1. Белгілеулерді енгіземіз: $R1=a^5$; $r2=a^{-5}$; $r3=a^7$;

Процедураны сипаттау кезінде парметрлер тізімінің көмегімен формальды x параметрін дәреже негізімен, n параметрін көрсеткішпен, процедура нәтижесі P-ны фактілі нәтижемен байланыстыру керек. Есепте үш дәрежелі операциясын орындау керек болғандықтан (R1,r2, r3) негізгі бағдарламада поцедураны үш рет шақырамыз.

2. Бағдарламасын жазамыз:

Program M_1;

Var

a, r1, r2, r3, z:real;

Procedure ST (x:real; n:integer; var P:real);

Var

I:integer;

Begin

P:=1;

For i:=1 to n DO P:=P*x;

End;

Begin

Writeln ('a sanin engiz-r');

Read (a);

ST(a,5,r1);

ST(1/a,5,r2);

ST(a,7,r3);

Z:=(r1+r2)/(2*r3);

Writeln('Z=',Z:6:2);

end.

3. Бағдарлама дұрыстығына көз жеткізіп, Ctrl+ F9 тетігін басу арқылы бағдарлама нәтижесін алыңыз.
4. Құрылған бағдарламаны сақтап, оқытушыға істелген жұмыс бойынша есеп беріңіз

ЖАТТЫҒУЛАР

1. Үш процедурадан және негізгі прграммалардан тұратын программа жазыңыздар. Бірінші процедура екі бүтін сан x және y-ті енгізуді ұйымдастырады, екіншісі олардың қосындысын тексереді, үшіншісі нәтижені шығаратын болсын. Бұл процедураларды негізгі программада пайдаланыңыздар. X және y-ті айнымалы ретінде қолданыңыздар.
2. Төрт санның үлкенін іздейтін бағдарламаны екі санның үлкенін табатын процедураны пайдаланып жазыңдар.
3. k санының факторриалын есептеуге арналған функцияны пайдаланып, $1!+2!+3!+\dots+n!$ Қосындысын есептейтін программа жазыңыздар.
4. Үш натурал санның ең үлкен ортақ бөлгішін анықтайтын программа жазыңыздар.
5. Ондығында жұп цифрлары болмайтын үшорынды сандардың қосындысын есептейтін программа құрыңдар.
6. N-нен артық емес және өзінің цифрларының әрқайсысына бөлінетін барлық натурал сандарды экранға шығаратын программа жазыңыздар.

7. Өзінің ицифрларының үш еселенген көбейтіндісіне тең N орынды ең кіші натурал x ($x \geq 10$) санын табатын бағдарлама құрыңыздар.
8. Цифрларының қосындысының квадраты x-ке тең болатын және M-нен кіші барлық натурал сандардың қанша екендігін есептейтін программа құрыңыз.
9. Цифрларының қосындысының квадраты x-ке тең болатын және M-нен кіші барлық натурал сандардың қанша екендігін есептейтін программа құрыңыздар.
10. Натурал m және n сандары берілген ($m < n$). m/n бөлшегін қысқартатын программа жазыңыздар.

Зертханалық жұмыс №11

Тақырыбы: Мәтіндік ақпараттарды өңдеу. Жолдар;

Мақсаты: Мәтін фрагменттерін алмастыру, қою және жою;

Қажетті құралдар: ДК /Pentium 3,4 Windows /, TurboPascal

Теориялық қажетті материалдарға сілтеме: C:\ Алгоритмдік тілде бағдарламалау;

Тапсырмалар

1. Нүктемен аяқталатын символдық жол берілген. Бос орындармен бөлінген сөздердің ең ұзыны мен ең қысқасының ұзындығын анықтаңдар.

Жұмысты орындалу реті

1. Program prog_1;

Var S1: String;

Max, min, I,J:integer;

Begin

Writeln (' нүктемен аяқталатын символдық текст енгізіңіздер');

Readln (S1);

J:=Pos ('.', S1);

S1:=Copy(S1,1, J-1);

Max:=0;Min:=256;

Repeat

I:=POS('.',S1);

Delete (S1,1,I);

If I-1 > Max Then Max:=I-1;

```
If (I-1<Min) And (I>1) Then Min:=I-1;
```

```
Until I=0;
```

```
Writeln ('Max=',Max);
```

```
Writeln ('Min=',Min);
```

```
Readln;End;
```

2. Бағдарлама дұрыстығына көз жеткізіп, Ctrl+ F9 тетігін басу арқылы бағдарлама нәтижесін алыңыз.

3. Құрылған бағдарламаны сақтап, оқытушыға істелген жұмыс бойынша есеп беріңіз

2. Жолды өңдеуге арналған функциялар:

Жұмысты орындалу реті

Pos(word, words) функциясы – жолдың қандай да бір бөлігін іздеуге мүмкіндік береді. Мұндағы: word – ізделінетін жол бөлігі, words – берілген жол.

```
PROGRAM Demo3 ;
```

```
VAR
```

```
    WORDS: STRING[79] ;
```

```
    POSITION : BYTE;
```

```
PROCEDURE SEARCH STENCIL;
```

```
BEGIN
```

```
    POSITION := POS (SERCH WORD,WORDS);
```

```
    IF POSITION<>0
```

```
        THEN WRITELN('бөлік “ ‘SERCH WORD,’ ” мына сөзде
```

```
“‘WORDS’ “орыннан бастап POSITION:3,’ .’)
```

```
ELSE WRITELN ('бөлік “ ‘SERCH WORD,’ ” бұл сөзде жоқ “ ‘WORDS’ “ .’);
```

```
END; {SEARCH STENCIL}
```

```
    WORDS: ='Электрификация ';
```

```
SEARCH WORD:= 'кац ';
```

```
SEARCH STENCIL;
```

```
SEARCH WORD:='Кац';
```

```
SEARCH STENCIL;
```

END.

'Электрификация ' сөзінде 'кац ' бөлігі 10 орыннан басталады.

'Электрификация ' сөзінде 'Кац ' бөлігі жоқ.

INSERT және DELETE процедурасы – текст бөлігін бір жолдан келесі жолға қорға және жолдан текст бөлігін өшіруге /алып тастау/ қолданылады.

INSERT процедурасы көрсетілген орыннан бастап басқа жолды қояды, мысалы, INSERT(WORD1, WORDS,4); - WORD1 жолын 4-ші орыннан бастап WORDS жолына қояды.

DELETE процедурасы – жолдағы тексттің бөлігін көрсетілген орыннан анықталған ұзындықпен өшіреді,

мысалы, DELETE (WORS, 2,3); - WORS жолынан 3 символдан, 2 орыннан басталатын текст бөлігін өшіреді.

PROGRAM Demo4 ;

VAR

WORDS: STRING[79] ;

WORD1: STRING[20] ;

BEGIN

WRITELN;

WORDS : ='бағдарлама ';

WRITELN (WORDS);DELETE (WORS,1,3);

WRITELN (WORDS); DELETE (WORS,3,3);

WORD1: ='A';

INSERT(WORD1, WORDS,1);

WRITELN (WORDS);WORD1: =' лаң';

IBSERT(WORD1,WORDS,2);

WRITELN (WORDS);END.

бағдарлама

дарлама

дама

Адама

Алаңдама

4. Бағдарлама дұрыстығына көз жеткізіп, Ctrl+ F9 тегігін басу арқылы бағдарлама нәтижесін алыңыз.

5. Құрылған бағдарламаны сақтап, оқытушыға істелген жұмыс бойынша есеп беріңіз
ЖАТТЫҒУЛАР

Берілген өрнектердің бағдарламасын құрып, мәндерін есептеңдер

1. Символдардан тұратын жол берілген. Сөз берілген. Жолдағы осы сөзді өшіріңдер.
2. Нүктемен аяқталатын символдық жол берілген. Бос орындармен бөлінген сөздердің қаншаса ұқсас әріптен басталатындығын анықтаңдар.
3. Нүктемен аяқталатын символдық жол берілген. Бос орындармен бөлінген сөздердің ішінде болмағанда бір «е» әріпі кездесетіндігін анықтаңдар.
4. Нүктемен аяқталатын символдық жол берілген. Осы жолдағы жақшалар толық, әр дұрыс қойылған ба? Соны анықтаңдар.
5. Нүктемен аяқталатын символдық жол берілген. Осы жолдың екіге еселік ондық үлестегі сандар екендігін анықтаңдар.
6. Нүктемен аяқталатын символдық жол берілген. Осы жолды экранға соңынан басына қарай шығарыңдар.
7. Студенттер фамилиялары тізімінен тұратын жол берілген. Осы тізімдегі ең ұзын фамилияны анықтаңдар.

Зертханалық жұмыс №12

Тақырыбы: Файлдармен жұмыс

Мақсаты: Файлдармен жұмыс;

Қажетті құралдар: 1. ДК /Pentium 3,4 Windows /, 2. TurboPascal

Теориялық қажетті материалдарға сілтеме: C:\ Паскаль тілінде бағдарламалау

Тапсырмалар

Мазмұны мына суреттегідей екі файл берілсін.

| | |
|----------------|------------------|
| Файл аты: Asan | Файл аты: Murat |
| 1 3 5 6 7 8 | Ata Ana Ara Sana |

| | |
|---------------|-------------|
| 2 4 4 4 | Almaty |
| 3 2 6 2 2 | A s t a n a |
| 4 3 3 3 3 3 3 | Kazakshtan |
| 5 7 7 7 7 7 7 | |
| 6 9 9 5 9 9 9 | |
| 7 | |
| 3 | |
| 1 | |

Әрбір элементті еселеу арқылы Asan файылын оқу.

Жұмысты орындалу реті

1. Бұл есепті шешу үшін (енгізу және шығару файлдарын текстік файл ретінде сипаттаймыз. Asan файылы алдын-ала құрылып, мәліметтермен толтырылған. Оны файлдық айнымалы F1ен байланыстырамыз да, оқу үшін ашамыз.

Assign(F1, 'Asan');

Reset (F1);

2. F2 фйнымалысын бос жолдың көмегімен стандартты шығару құрылғысымен (дисплей экранына) байланыстырамыз және оны жауапқа шығарамыз.

Assign(F2, '');

Rewrite (F2);

3. Дисплей орнына DOS форматындағы кез-келген файлдың атын көрсетуге болады. Сонан соң F1 файылының элементтерін оқитын және оларды бос орын арқылы F2 файылына жазатын цикл ұйымдастырамыз. Ол цикл EOF(F1) функциясының көмегімен F1 файылының соңы табылғанша орындалады. Жұмыс соңында екі файлды да жабамыз.

4. Program prog_1;

Var F1, F2:text;

X:Char;

Begin

Writeln (' нүктемен аяқталатын символдық текст енгізіңіздер');

ASSIGN (F1, 'C:\E\ONE'); { Файл орналасқан орынды көрсету}

Reset (F1);

Assign (F2, '');

```
Rewrite (F2);  
While not EOF (F1) Do  
Begin  
Read (F1, X);  
Write (F2, X+ ' ' + X)  
End;  
Writeln;  
Close (F1);  
Close(F2);  
End.
```

5. Бағдарлама дұрыстығына көз жеткізіп, Ctrl+ F9 тегігін басу арқылы бағдарлама нәтижесін алыңыз.

6. Өңделген Asan файылы мына түрде жауапқа шығарылады. Әрбір символ бос орын арқылы еселенеді.

```
11 33 33 55 66 77 88  
22 44 44  
33 22 66 22 22  
44 33 33 33 33 33 33  
55 77 77 77 77 77 77  
66 99 99 55 99 99 99  
77  
33  
11
```

7. Құрылған бағдарламаны сақтап, оқытушыға істелген жұмыс бойынша есеп беріңіз

ЖАТТЫҒУЛАР

Берілген өрнектердің бағдарламасын құрып, мәндерін есептеңдер

1. “a” әріпін еселеу арқылы Murat файылын оқындар.
2. Murat файылын оқындар және «a» әріпін санаңдар
3. Asan файылын оқындар және онда кездесетін цифрлардың қосындысын табыңдар.

4. Murat файылының элементтерін өзіне-өзін екі рет жазыңдар.
5. Murat файылын оқыңдар және соңынан басына қарай жауапқа шығарыңдар
6. Murat файылын өзіне-өзін соңынан басына қарай жазылған түрде тізбектей жазуды құрыңдар.
7. Murat файылында бірінші нүктеден кейінгі сөздерді жойыңдар.

Зертханалық жұмыс №13

Тақырыбы: Жиындармен жұмыс. Жиындармен амалдар орындау.

Мақсаты: Жиындарға қатысты берілген есептерді шешіп үйрену;

Қажетті құралдар: ДК /Pentium 3,4 Windows /, TurboPascal

Теориялық қажетті материалдарға сілтеме: C:\ Алгоритмдік тілде бағдарламалау

Тапсырмалар

Символдық типтегі үш жиын өз конструкторларымен берілген:

Y1:=['A','B','D','R','M'];

Y2:=['R','A','H','D'];

Y3:=['A','R'];

Мына формуламен анықталатын жаңа жиын құрыңдар:

$$X=(Y1 \cap Y2) \cup (Y1/Y2)$$

Алынған жиынды жауапқа шығарыңдар және Y3 жиыны X жиынының ішкі жиыны болатындығын анықтаңдар.

Жұмысты орындалу реті

```
1. Program PROG_1;
Var Y1, Y2, Y3, X:SET OF SHAR;
```

```
C:CHAR;
```

```
BEGIN
```

```
Y1:=['A','B','D','R','M'];
```

```
Y2:=['R','A','H','D'];
```

```
Y3:=['A','R'];
```

```
{Формула бойынша жиын құру және жауапқа шығару};
```

$X=(Y1*Y2) +(Y1-Y2)$

WRITELN ('X жиыны =');

For C:='A' TO 'R' DO

IF C IN X THEN WRITE(C);

WRITELN;

{Y3жиыны X жиынының ішкі жиыны екендігін тексеру}

IF Y3<=X Then WRITELN ('Y3жиыны X жиынының ішкі жиыны');

ELSE WRITELN ('Y3жиыны X жиынының ішкі жиыны емес');

END.

Жауабы: X жиыны = ABDMR

Y3жиыны X жиынының ішкі жиыны

2. Бағдарлама дұрыстығына көз жеткізіп, Ctrl+ F9 тетігін басу арқылы бағдарлама нәтижесін алыңыз.
3. Құрылған бағдарламаны сақтап, оқытушыға істелген жұмыс бойынша есеп беріңіз

ЖАТТЫҒУЛАР

Берілген өрнектердің бағдарламасын құрып, мәндерін есептендер

1. 1...20 бүтін сандар жиынынан: 6-ға қалдықсыз бөлінетін сандар жиынын; 2-ге немесе 3-ке қалдықсыз бөлінетін сандар жиынын ажыратып алыңдар.
2. N (1,2, ..., N) саннан тұратын жиын берілген. Осы (N<11) сандардан қайталанбайтын алмастырулар жасайтын процедура құрыңдар. Мысалы, N =3 үшін 6 алмастыру бар:
123 132 231 213 312 321
3. Ішкі бағдарлама - функцияны пайдаланып A(5.5), B(10.10) матрицаларының диагональ бойындағы элементтерінің қосындысын табыңдар.
4. Символдар жолы берілген. Сөздегі O әрпінің санын анықтаңдар.
5. Подпрограмма процедураны пайдаланып есептендер. A(15), B(18), C(12) массивтерінің [1,5] интервалында жататын элементтердің санын табыңдар.
6. Сары түсті экранға қызыл түзу, жасыл үшбұрыш, көк тіктөртбұрыш және қоңыр шеңбер салу программасын жаз.
7. Ақшыл көк түсті экранға боялған тік бұрышты трапецияның ішіне боялған шеңбер мен үшбұрыш сал.

Зертханалық жұмыс №14

Тақырыбы: Жазулар

Жұмыстың мақсаты: Жазуларға қатысты берілген есептерді шешу.

Қажетті құралдар: ДК /Pentium 3,4 Windows /, TurboPascal

Теориялық қажетті материалдарға сілтеме: C:\ Алгоритмдік тілде бағдарламалау

Тапсырмалар

Мына көпмүшеліктегі А айнымалысы бойынша ұқсас мүшелердің қосындысын табыңдар.

$5A+17B-4A-3K+13+9B-15R-4B$

Жұмысты орындалу реті

1. Көпмүшеліктің бір элементін жазу деп қарастырамыз, себебі түрлі типтегі мәліметтен – коэффициенттен және әріптен тұрады. Бағдарлама үшін мынадай белгілеу енгізейік M1 көпмүшеліктің элементі, COEF- коэффициент (жазу элементі); BUK- әріп (жазу элементі); ELEM-жазудың типі; SUM-коэффициенттер қосындысы.

2. Program Prog_1;
Type ELEM= Record

COEF: integer;

BUKWA:Char;

End;

Var M1: ELEM;

M1: ELEM;

SUM:INTEGER;

BEGIN

SUM:=0;

READLN;

WRITELN ('Көпмүшелікті енгізіңдер');

WHILE NOT ELON DO

BEGIN

READ(M1.COEF, M1.BUKWA);

IF M1.BUKWA='A' THEN SUM:=SUM+M1.COEF

END;

M2.COEF:=SUM;

M2.BUKWA='A'

WRITELN ('Ұқсас мүшелердің қосындысы =',M2.COEF:3, M2.BUKWA);

END.

3. Бағдарлама дұрыстығына көз жеткізіп, Ctrl+ F9 тетігін басу арқылы бағдарлама нәтижесін алыңыз.

4. Жауабы: Көпмүшелікті енгізіңіздер: $5A+17B-4A-3K+13+9B-15R-4B$ Ұқсас мүшелердің қосындысы 9A

5. Құрылған бағдарламаны сақтап, оқытушыға істелген жұмыс бойынша есеп беріңіз
ЖАТТЫҒУЛАР

1. Бағдарламалау үйрмесіне қатысатын оқушылар тізімі берілген. Беске оқитын және фамилиясы А әрпінен басталатын оқушылардың санын анықтаңдар.

2. Екі $X=4+5i$ және $Y=8+7i$ комплекс санының S қосындысын P көбейтіндісін табыңдар.

3. Нүктемен аяқталатын символдық жол берілген. Осы жолды экранға соңынан басына қарай шығарыңдар.

4. Студенттер фамилиялары тізімінен тұратын жол берілген. Осы тізімдегі ең ұзын фамилияны анықтаңдар

5. Бүтін санның цифрларының қосындысын есептейтін функция жазыңдар

6. Келесі өрнектің мәнін есептеуге бағдарлама құрыңдар

$S=5!+6!+7!+8!$ Мұндағы S нақты сан

Зертханалық жұмыс №15

Тақырыбы: Графика. Graph модулі;

Мақсаты: Негізгі графикалық объектілерді салу операторларымен танысу және оларды бағдарламада қолдана білуге дағдылану;

Қажетті құралдар: ДК /Pentium 3,4 Windows /, TurboPascal

Теориялық қажетті материалдарға сілтеме: C:\ Алгоритмдік тілде бағдарламалау

Турбо Паскальда графикамен жұмыс істеу үшін графикалық операторларды (Uses), драйверлерді (GraphDriver), модульдерді (Crt, Graph), режимдерді (GraphMode, CloseGraph) қосуымыз қажет.

Арнайы графикалық оператор – Uses болып табылады.

Турбо Паскальда графикалық бейнелерді салу үшін стандартты кітапханалық Graph модулі арналған. Ол әр түрлі адаптерлі дисплейлерде графикалық режимдерді толығымен басқаруды қамтамасыз ететін программалардың кітапханасы. Ал Crt модулі экранға информацияларды шығаруды, пернетақтадан берілгендерді енгізуді, сол сияқты терезелермен жұмыс істеуді, дыбыстық сигналдар мен түрлі-түсті тексттік жолдарды экранға шығаруды қамтамасыз етеді.

Жұмысты орындалу реті

Program Сурет;

Uses Graph, Crt;

Var GraphDriver, {Графикалық драйвер}

```

Graphmode: integer;                                {Графикалық режим}
Begin
Clrscr;                                            {Экранды тазалау}
GraphDriver := 0;                                  {Графикалық драйверді ашу}
InitGraph (GraphDriver, GraphMode, ' ');
                                                    {Графикалық жүйенің инициализациясын
                                                    жасайды және аппаратураны графикалық
                                                    режимге ауыстырады}
If GraphResult <> 0 then                            {қате пайда болды}
Begin
Writeln;                                          {Графика қатесінің инициализациясы}

Writeln;                                          {Бағдарлама жұмысын тоқтату}
Halt;                                             {Тоқтату}
End;
                                                    {Осы жолдан бастап графикалық
                                                    әрекеттерді орындау процесі басталады }
Readln;                                          {Enter түймесіне басқанға дейін
                                                    суретті экранда көрсетіп тұру}
CloseGraph;                                       {Графикалық режимді жабу}
End.

```

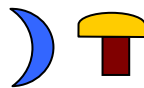
GraphDriver сөзін (G) деп, ал GraphMode сөзін (RGM) деп қысқартып алуға болады.

2. Бағдарлама дұрыстығына көз жеткізіп, Ctrl+ F9 тегігін басу арқылы бағдарлама нәтижесін алыңыз.
3. Құрылған бағдарламаны сақтап, оқытушыға істелген жұмыс бойынша есеп беріңіз

ЖАТТЫҒУЛАР

1. Сары түсті экранға қызыл түзу, жасыл үшбұрыш, көк тіктөртбұрыш және қоңыр шеңбер салу программасын жаз.
2. Ақшыл көк түсті экранға боялған тік бұрышты трапецияның ішіне боялған шеңбер мен үшбұрыш сал.

3. Боялған тең бүйірлі үшбұрыштың ішіне боялған тіктөртбұрышты, ал оның ішіне боялған шеңберді салуға программа құр.
4. Боялған тікбұрышты үшбұрышқа боялған іштей және сырттай шеңберлерді салуға программа құр.
5. Боялған тең бүйірлі трапецияға іштей шеңбер, ал оның ішіне тең қабырғалы үшбұрыш салуға программа құр.
6. Саңырауқұлақ және ай салу.
7. Адамның басын бейнелеңіз.
8. «Жаңбыр» бағдарламасын жазыңыздар-экранда (кездейсоқ күйде) 50 нүкте жоғарыдан төменге қарай қозғалып келе жатқандай суреттеледі. Нүктелердің бірі экранның төменгі шекарасына жеткенде ол экранның жоғарғы бөлігінде пайда болуы керек.
9. «Жаңбыр» бағдарламасын кейбір нүктелер шапшаң, кейбіреулері баяу қозғалатындай етіп өзгертіңіздер.
10. «Жаңбыр» бағдарламасын нүктелер өз қозғалысын төменгі сол жақ бұрыштан бастап, жоғарғы оң жақ бұрышта аяқталатындай етіп өзгертіңіздер.



ӘДЕБИЕТТЕР ТІЗІМІ

1. Б. Д. Сыдықов. Алгоритмдеу және программалау негіздері. Алматы, 2003
2. Ш.Тойкенов “Паскаль тілінде программалау” - Алматы “Д.,некер”, 2001ж.
3. Қ. С. Әбдиев,М. Ж. Бекпатшаев. Алгоритмдеу негіздері. Алматы,1998 ж.
4. Абдиев К. С., Буралкиева Г. М., Якупова А.Н. Практикум по программированию. Алматы,1996.
5. А.Н. Нуруллаев, К.М. Беркинбаев, Г.С. Тойшибекова, Б. Сыдықов. Лабораторный практикум по программированию. Кентау,1997 ж.

6. Б.Қ.Некесбеков, Г.З. Халықова. Паскаль тілінің негіздері. А., 1998 ж.
7. Фаронов В.В. Программирование на персональных ЭВМ в среде Турбо-Паскаль. М., 1991.
8. В.С.Новичков, Н.И. Парфилова, А.Н.Пылкин. Паскаль. М., 1994.
9. Г. Л. Семашко, А.И. Салтыков. Программирование на языке Паскаль. М., 1998.
10. В. Ф. Шангин, Л. М. Поддубная. Программирование на языке Паскаль. М., 1991
11. А. И. Гусева. Учимся программировать: PASCAL 7.0. М., 1998.
12. Глинский Я. Н., Анохин В.Е., Ряжская В.А. Turbo Pascal 7.0. и Delphi. Учебное пособие- М, 2001 г.
13. Фаронов В. В. «Основы Turbo Pascal 6.0» Москва 2000 ж.
14. Меженный О. А « Turbo Pascal Учитесь программировать» Москва, 2001ж.
15. Абрамов В. Г «Введение в язык Паскаль для вузов» Москва, 1988
16. Чернов Б.И. “Программирование на алгоритмических языках Бейсик, Фортран, Паскаль”
17. Шангин В.Ф., Поддубная Л.М. “Программирование на языках–Паскаль
18. Бурин Е.А. Программирование на языке Турбо-Паскаль. 2000.
19. В. Н. Пильщиков. Сборник упражнений по языку Паскаль. М., 1989.
20. Г.Г. Рапаков, С.Ю. Ржеуцкая Программирование на языке Pascal Санк – Петербург, 2004. - 470 с.

gg_begarisheva@bk.ru

МАЗМҰНЫ

| | |
|-----------------------------------|---|
| КІРІСПЕ | 3 |
| 1. АЛГОРИТМ ТҮСІНІГІ | 5 |
| 1.1. Жазу тәсілдері | 5 |
| 1.2. Алгоритм түрлері | 6 |
| 1.3. Алгоритмнің қасиеттері | 9 |

| | |
|---|----|
| 2. ПАСКАЛЬ БАҒДАРЛАМАЛАУ ТІЛІ | 10 |
| 2.1. Паскаль тілінің негізгі түсініктері | 11 |
| 2.1.1. Алфавит пен резервтелген сөздер | 11 |
| 2.1.2. Өрнек, амалдар, оперондалар | 11 |
| 2.1.3. Идентификаторлар | 11 |
| 2.1.4. Turbo Pascal тіліндегі берілгендер типі | 12 |
| 2.1.5. Сандартты функциялар | 14 |
| 2.2. Бағдарлама құрылымы | 15 |
| 2.3. TurboPascal тілін іске қосу | 16 |
| 3. ПАСКАЛЬ ТІЛІНДЕГІ АЛГОРИТМНІҢ ТҮРЛЕРІ | 18 |
| 3.1. Сызықтық алгоритмді бағдарламалау | 18 |
| 3.2. Тармақталушы алгоритмді бағдарламалау | 20 |
| 3.3. Таңдау операторы | 23 |
| 3.4. Цикдық алгоритмді бағдарламалау | 24 |
| 3.4.1. Параметрлі цикл | 24 |
| 3.4.2. Дейін циклы | 27 |
| 3.4.3. Кейін циклы | 29 |
| 4. ПАСКАЛЬ ТІЛІНДЕГІ МАССИВТЕР | 31 |
| 4.1. Бір өлшемді массив | 32 |
| 4.2. Екі өлшемді массив | 34 |
| 5. КӨМЕКШІ БАҒДАРЛАМАЛАР | 42 |
| 6. ФАЙЛ. МӘТІНДІК ФАЙЛДАРМЕН ЖҰМЫС | 44 |
| 7. ЖОЛДАР ҚАТАРЛАР | 46 |
| 7.1. Жолдық өрнектер мен амалдар | 46 |
| 7.2. Жиын. Символдық массивтермен амалдар орындау | 50 |

| | |
|---|----|
| 7.3. Жазу. | 51 |
| 8. ПАСКАЛЬ ТІЛІНІҢ ГРАФИКАЛЫҚ МҮМКІНДІКТЕРІ GRAPH МОДУЛЬ. | 54 |
| 8.1. Графикалық режим | 54 |
| 8.2. Графикалық координаттар. | 54 |
| Зертханалық жұмыс №1. | 56 |
| Зертханалық жұмыс №2. | 59 |
| Зертханалық жұмыс №3. | 60 |
| Зертханалық жұмыс №4. | 61 |
| Зертханалық жұмыс №5. | 63 |
| Зертханалық жұмыс №6. | 65 |
| Зертханалық жұмыс №7. | 66 |
| Зертханалық жұмыс №8. | 68 |
| Зертханалық жұмыс №9. | 72 |
| Зертханалық жұмыс №10. | 74 |
| Зертханалық жұмыс №11. | 75 |
| Зертханалық жұмыс №12. | 78 |
| Зертханалық жұмыс №13. | 80 |
| Зертханалық жұмыс №14. | 81 |
| Зертханалық жұмыс №15. | 83 |
| ӘДЕБИЕТТЕР ТІЗІМІ. | 85 |
| МАЗМҰНЫ | 86 |