

N.N.Vəliyev



Turbo

PASCAL

WINDOWS üçün



BAKİ -



UOT 681.3.06

Vəliyev Namiq Nəcəf oğlu. Turbo Pascal Windows üçün, 2 cildə. 1-ci cild.

Redaktor: Azərbaycan Respublikası Milli Elmlər Akademiyasının müxbir üzvü, Azərbaycan Dövlət Neft Akademiyası «Hesablama maşınları və proqramlaşdırma» kafedrasının müdiri, professor **S.Q.KƏRİMOV**

Rəy verənlər: Azərbaycan Dövlət Pedaqoji Universiteti «Hesablama riyaziyyatı və informatika» kafedrasının professoru **Ə.M.MƏMMƏDOV**

Bakı Dövlət Universiteti «İnformatika» kafedrasının müdiri, dosent **M.S.XƏLİLOV**

Bakı, «Qlobus-print», 2003, 142 səh.

Dərs vəsaitinə Windows əməliyyat sistemində işləyən Turbo Pascal proqramlaşdırma dili haqqında nəzəri və praktiki məlumatlar daxil edilmişdir. Vəsaitdə çoxsaylı alqoritmlər və proqramlar nəzərdən keçirilir. Hər bir fəsil laboratoriya işləri ilə müşayiət edilir. Dərs vəsaitindən orta məktəb şagirdləri, universitet tələbələri, eləcə də proqramlaşdırma ilə məşğul olmaq istəyən hər bir şəxs istifadə edə bilər.

MÜNDƏRİCAT

	GİRİŞ	5
1.	İNTEQRALLAŞDIRILMIŞ INSTRUMENTAL ÖRTÜK	8
1.1.	İnteqrallaşdırılmış instrumental örtüyün əsas pəncərəsi	8
1.2.	Proqram mətninin daxil edilməsi, sazlanması və icrası	11
1.3.	<i>Laboratoriya işi №1.</i> Turbo Pascal mühitində proqram mətnləri ilə əməliyyatlar	13
2.	ALQORİTMLƏR, ONLARIN XASSƏLƏRİ VƏ TƏSVİR ÜSULLARI	15
2.1.	Alqoritmlərin tipik strukturları	15
2.2.	Alqoritmlərin xassələri və təsvir üsulları	18
2.3.	<i>Laboratoriya işi №2.</i> Alqoritmın təbii dildə və blok-sxem üsulları ilə təsviri	21
3.	PASCAL DİLİNİN ƏSASLARI	22
3.1.	Pascal dilinin əsas anlayışları	22
3.2.	Hesabi və məntiqi ifadələr, mənsub etmə operatoru	24
3.3.	<i>Laboratoriya işi № 3.</i> Hesabi və məntiqi ifadələr	25
3.4.	Pascal-proqramın elanlar bölməsi	27
3.5.	Daxil etmə və xaric etmə standart proseduraları	30
3.6.	<i>Laboratoriya işi № 4.</i> Xətti proqramların tərtibi	33
4.	PASCAL DİLİNİN İDARƏEDİCİ KONSTRUKSIYALARI	34
4.1.	Pascal dilinin operator qrupları	34
4.2.	Şərtsiz konstruksiyalar	34
4.3.	Şərti konstruksiyalar	36
4.4.	<i>Laboratoriya işi № 5.</i> Şərti və şərtsiz konstruksiyalar	44
4.5.	Dövri konstruksiyalar	45
4.6.	<i>Laboratoriya işi № 6.</i> Dövri konstruksiyalar	54
5.	VERİLƏNLƏRİN SADƏ TIPLƏRİ	59
5.1.	Verilənlərin tiplərinin struktur təsviri	59
5.2.	Tam tiplər	60
5.3.	Məntiqi tiplər	68
5.4.	Simvol tipləri	71
5.5.	Sadalanən tiplər	73
5.6.	Diapazon tipləri	74
5.7.	+ Həqiqi tiplər	75
5.8.	<i>Laboratoriya işi № 7.</i> Verilənlərin sadə tipləri	78
6.	VERİLƏNLƏRİN SƏTİR TIPLƏRİ	81

- O Y U -
K İ T A B X A N A

6.1.	Sətir tipli verilənlərin təsviri	81
6.2.	Sətir tipli verilənlərin emalı	82
6.3.	<i>Laboratoriya işi № 8. Verilənlərin sətir tipləri</i>	89
7.	MASSİVLƏR	91
7.1.	Massivlərin təsviri və emalı	91
7.2.	<i>Laboratoriya işi № 9. Massivlərin emalı</i>	109
8.	Yazılar	114
8.1.	Yazıların təsviri və emalı	114
8.2.	<i>Laboratoriya işi № 10. Yazıların emalı</i>	118
9	Çoxluqlar	121
9.1.	Çoxluqların təsviri və emalı	121
9.2.	<i>Laboratoriya işi № 11. Çoxluqların emalı</i>	123
10	FAYLLAR	125
10.1.	Faylların növləri	125
10.2.	Mətn faylları	125
10.3.	<i>Laboratoriya işi № 12. Mətn faylları</i>	139
	Ədəbiyyat	142

GİRİŞ

Fərdi kompüterlərin ildən-ilə təkmilləşdirilməsi, qısa dövrlərdə yeni-yeni kompüter texnologiyalarının yaranması, insan və kompüter arasındakı ünsiyyət vasitələrinin sadələşdirilməsi, eləcə də Respublikamızda kompüter ixtisasçılarına olan böyük tələbat xalqımızda kompüter elminə yüksək maraq yaratmışdır. Fərdi kompüterlər artıq fəaliyyət dairəsindən asılı olmayaraq, hər birimizin gündəlik həyatına daxil olmuşdur. Təsədüfi deyildir ki, son zamanlarda bir çoxumuzun ailəsində ənənəvi olan mebel, soyuducu, televizor və s. kimi fərdi kompüterlərə də rast gəlinir.

Hal-hazırda məktəb yaşlı uşaqlarımızın və gənclərimizin böyük bir qismi Windows əməliyyat sistemini, Internet kompüter şəbəkəsini, tərcüməçi proqramları, mətn, cədvəl və qrafik verilənlər redaktorlarını kifayət dərəcədə mənimsəmişlər və onlar bu sistemlərlə işləyərək öz informasiya tələbatlarını ödəyirlər. Lakin sadaladığımız sistemlər onları tam qane etmir və bu şəxslər proqramlaşdırmanın sirlərini öyrənməyə daha çox can atırlar. Bu işdə əldə etdiyiniz «Turbo Pascal Windows üçün» dərs vəsaiti sizə çox kömək edər. Qeyd etmək istərdim ki, Pascal proqramlaşdırma dili ilk növbədə məhz öyrədici dil kimi yaradılmışdır və hal-hazırda demək olar ki, bütün ali tədris ocaqlarında ilk proqramlaşdırma dili kimi tələbələrə öyrədilir. Bu dil gələcəkdə daha mükəmməl proqramlaşdırma dillərini öyrənmək üçün sizə geniş imkanlar yaradar.

Kitaba daxil edilmiş materiallar elə səviyyədə yazılmışdır ki, oxucularda proqramlaşdırma haqqında ilkin məlumatlar olmadığı hallarda belə, onlar bu materialları dərk edə bilsinlər. Ayrı-ayrı fəsillərdə nəzəri məlumatlarla yanaşı, oxuculara kifayət qədər alqoritmlər və proqramlar da təqdim edilmişdir. Hər bir fəsilin sonu laboratoriya işləri ilə başa çatır. Laboratoriya işlərinə daxil edilmiş bütün məsələlərin sizin tərəfinizdən yerinə yetirilməsi tövsiyə edilir. Nəzərə alın ki, vəsaitə daxil edilmiş hər bir məsələnin həllində bir incəlik, özünəməxsus bir xüsusiyyət gizlənir. Bu incəlikləri hiss edib, seçib, həm sizə təqdim edilmiş məsələnin proqramında, həm də tərtib edəcəyiniz daha mühüm və mürəkkəb strukturlu proqramlarda nəzərə alsaz, bu sizə proqramlaşdırma sahəsində daha yüksək zirvələrə qalxmaq üçün zəmin yaradar.

Dərs vəsaiti 2 cildən ibarətdir. Birinci cildə Turbo Pascal proqramlaşdırma dilinin ilkin anlayışları daxil edilmişdir.

Birinci fəsildə siz Turbo Pascal proqramlaşdırma dilinin integrallaşdırılmış instrumental örtüyü, örtüyə daxil edilmiş müəyyən funksiyaları yerinə yetirən müxtəlif düymələrlə tanış olacaqsınız. Fəsildə verilmiş ma-

terialların mənimsənilməsi sizə, sonralar tərtib etdiyiniz proqram modullarını Pascal mühitində kompüterin yaddaşına daxil edərək icrasını həyata keçirmək imkanı yaradacaqdır.

Dərs vəsaitinin ikinci fəslə sadə alqoritmlərin qurulmasına həsr edilmiş, alqoritmlərin xassələri və təsvir üsulları verilmişdir. Fəsilə alqoritmlərin təbii dildə təsvirinə geniş yer verilib. Bu alqoritmlərlə tanış olduqdan sonra, güman edirəm, hiss edəcəksiniz ki, bizim gündəlik həyatımız çoxsaylı alqoritmlərin icrası ilə bağlıdır. Bu alqoritmlər bizim hər birimizin beynində çoxdan bəri əks olunmuşdur, beynimizin yüksək sürətlə işləməsi nəticəsində, özümüz də hiss etmədən, biz bu alqoritmləri doğru və səlis realizə edirik. Qeyd etmək istərdim ki, proqramlaşdırmada məhz əsas çətinlik belə alqoritmlərin bizim tərəfimizdən düzgün ifadə etməyimizdən ibarətdir. Vəsaitə daxil edilmiş materialları oxuduqca, sizin çətinlikləriniz, zənnimcə, belə alqoritmlərin qurulması prosesində baş verəcək. Bu sizi heç də qorxutmasın və siz tutduğunuz yoldan, proqramlaşdırma elminə yiyələnməkdən çəkinməyin, çətinliklər sizi ruhdan salmasın. Onu bilin ki, sizin səyiniz hədəf getməyə də bilər, çətinlikləri dəf etmək üçün, sadəcə olaraq, bir qədər çox çalışmaq lazımdır.

Üçüncü fəslə Pascal dilinin əsas anlayışları, bəzi operatorları, hesabi və məntiqi ifadələrin yazılış qaydaları daxil edilmişdir. Bu fəsildəki materialları mənimsədikdən sonra, siz artıq öz ilk proqramınızı tərtib edib, Pascal mühitində icra edərək, ilk nəticələrinizi ala bilərsiniz.

Dördüncü fəsilə Pascal dilinin idarəedici konstruksiyalarına baxılmışdır. Fəsilə çoxsaylı alqoritmlər, proqram modulları və tapşırıqlar əlavə edilmişdir. Bu materialları öyrəndikdən sonra siz artıq həlli o qədər də asan olmayan məsələlərin öhdəsindən gələ bilərsiniz. Qarşıda isə sizi daha çətin, lakin çox maraqlı məsələlərin həlli gözləyir. Sizə tövsiyə olunur ki, bu fəsilə daxil edilmiş materialları tamamilə dərk etmədən, sonrakı fəsillərə keçməyə tələsməyin.

Beşinci fəsil verilənlərin sadə tiplərinə həsr edilmiş, verilənlərin hər bir mümkün sadə tipi geniş təhlil edilmiş, materiallar konkret misallarla izah edilmişdir. Bu fəsildə artıq siz həll alqoritmə mürəkkəb olan bir məsələ ilə rastlaşacaqsınız.

Altıncı fəsilə verilənlərin sətir tipləri öyrənilir, bu tiplərə aid edilən standart funksiya və proseduraların işləmə prinsipləri misallarla sizə çatdırılır.

Yeddinci fəsil vəsaitin daha əsas fəsillərindən biri hesab edilə bilər. Bu fəsildə müxtəlif məsələlərin həllində bizim daha çox rastlaşdığımız massivlərdən və onların emalından söhbət açılır. Fəsilə daxil edilmiş

çalışmalar sizin proqramlaşdırma sahəsində biliklərinizi xeyli artırmağa kömək edəcəkdir.

Səkkizinci fəsil yazıların təsvirinə və emalına həsr edilmişdir. Bu fəsildəki materiallar ola bilsin ki, əvvəlki materiallar kimi sizdə o qədər də maraq oyatmasın. Lakin onu bilmək lazımdır ki, sonrakı materialların mənimlənməsində, xüsusilə fayllarla işləyərkən, öyrənəcəyiniz bu materiallar sizə çox lazım olacaqdır.

Doqquzuncu fəsil həcmə çox kiçik olsa da, çox maraqlı bir mövzunu, çoxluqların təsviri və emalı mövzusunun əhatə edir. Bu biliklər sizə riyaziyyat kursundan tanış olmalıdır və sizdən bir qədər riyazi bilik tələb edir. Yeri gəlmişkən, sizə çatdırmaq istərdim ki, proqramlaşdırma elmi xüsusilə riyaziyyat elmi ilə sıx bağlıdır və gələcəkdə görəcəksiniz ki, dərin riyazi biliyi olamayan şəxslər üçün yüksək səviyyəli, dünya standartlarına uyğun proqram təminatları yaratmaq həddindən artıq çətindir, bir çox hallarda isə mümkün deyil. Bu səbəbdən, indiki vaxtınızdan səmərəli istifadə edin və vaxtınızın bir hissəsini riyazi biliklərinizin artırılmasına həsr edin.

Onuncu fəsil vəsaitin birinci cildinin sonuncu fəsilidir və mətn fayllarının yaradılmasına, onların emalı üsullarına həsr edilmişdir. Fəsil çoxsaylı çalışmalar və tapşırıqlarla təchiz edilmişdir.

Dərs vəsaitinin ikinci cildinə aşağıda göstərilən fəsillərin daxil edilməsi nəzərdə tutulmuşdur:

- *Tipləşdirilmiş və tipləşdirilməmiş fayllar;*
- *Göstəricilər və dinamik yaddaş;*
- *Obyektivli proqramlaşdırma;*
- *Modullar;*
- *Qrafik verilənlərin emalı;*
- *Təqribi hesablama üsullarının proqramlaşdırması;*
- *Olimpiada xarakterli çalışmalar.*

Dərs vəsaiti Azərbaycan Dövlət Neft Akademiyasında və Odlar Yurdu Universitetində oxuduğum mühazirələr əsasında işlənilib hazırlanmışdır. Vəsaitin yazılması dövründə mənə mənəvi dayaq olan çoxsaylı tələbələrə, müəllim kollektivinə, verdikləri dəyərli tövsiyələrə görə dərs vəsaitinin redaktoru professor Kərimov Sabit Qəhrəman oğluna və həmkarım, dosent Həbibullayev Səlahəddin Bəxtiyar oğluna, vəsaitdə istifadə olunan alqoritmlərin blok-sxemlərinin, illüstrasiyaların, üz qabığındakı dizayn işinin yerinə yetirilməsində mənə yaxından kömək göstərən oğlanlarım Fərid və Fəxriyə dərin minnətdarlığımı bildirirəm.

1. İNTEQRALLAŞDIRILMIŞ INSTRUMENTAL ÖRTÜK

1.1. İnteqrallaşdırılmış instrumental örtüyün əsas pəncərəsi

Turbo Pascal for Windows inteqrallaşdırılmış instrumental örtüyü uyğun proqram vasitələri yerləşdirilmiş kompakt disk vasitəsilə kompüterin vinçes-terində installaşdırılır. İntallaşma prosesi başa çatdıq-
dan sonra Turbo Pascal-a aid olan bütün proqram
modulları **C:\TPW** direktoriyasına yazılır və monito-
run ekranında olan işçi stola şəkil 1.1-də göstərilmiş
yarlık əlavə edilir.

Mausun göstəricisini işçi stoldakı və ya
C:\TPW direktoriyasındakı həmin yarlığın üzərinə
gətirib, sol klavişini basmaqla Turbo Pascal mühitinə
daxil olmaq olar. Bu zaman ekranda ani zamanlıq is-
tehsalçı firmanın adını, proqram məhsulunun versiya-
sını, yaradılma tarixini və Paskalın şəklini özündə əks
etdirən ön şəkil (şəkil 1.2) və Turbo Pascal-ın əsas

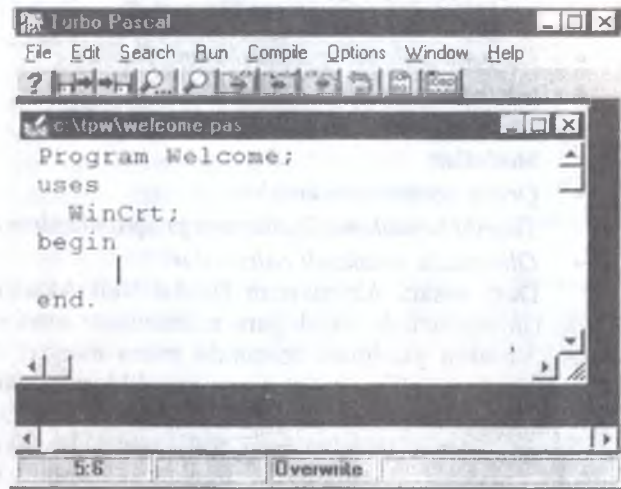


Şəkil 1.1



Şəkil 1.2

pəncərəsi gö-
rünür (şəkil
1.3). Turbo
Pascal-ın əsas
pəncərəsi öz
funksional tə-
yinatlarına gö-
rə fərqlənən
bir neçə his-
sədən təşkil
olunmuşdur:
menyu sətri,
funksional
düymələr sə-
tri, işçi zona və
cari vəziyyət
sətri. Menyü
sətri aşağıda
göstərilən



Şəkil 1.3. Turbo Pascal-ın əsas pəncərəsi

menyulardan və altmenyulardan ibarətdir:

File – fayllar ilə əsas əməliyyatları yerinə yetirir, məsələn, yeni fayl yaradır (**New**), mövcud faylı əməli yaddaşa yükləyir (**Open**), faylı vinçesterdə yadda saxlayır (**Save**), faylın tərkibini printerdə çap edir (**Print**) və s.;

Edit – proqram mətninin redaktə edilməsində əvvəlki əməliyyatın ləğv edilməsi (**Undo** və ya **Alt+BkSp**), sonrakı əməliyyatın bərpa edilməsi (**Redo**), qeyd olunmuş fraqmentin silinərək buferə göndərilməsi (**Cut** və ya **Shift+Delete**), qeyd olunmuş fraqmentin buferdə nüsxəsinin alınması (**Copy** və ya **Ctrl+Insert**), buferin tərkibinin kursurun vəziyyətindən bərpa edilməsi (**Paste** və ya **Shift+Insert**), qeyd olunmuş fraqmentin silinməsi (**Clear** və ya **Ctrl+Delete**) kimi əsas əməliyyatları yerinə yetirir;

Search – məndə fraqmentin axtarışını həyata keçirir (**Find**), tələb olunduqda tapılan fraqmeti yenisi ilə əvəz edir (**Replace**), tələbə görə verilmiş nömrəli sətə keçidi yerinə yetirir (**Go to line number**) və s.;

Run – işçi zonada yerləşən proqramı icra edir (**Run** və ya **Ctrl+F9**);

Compile – işçi zonada yerləşən proqramı kompilyasiya edir (**Compile** və ya **Alt+F9**);

Options – kompilyatorun işi üçün parametrlərin qiymətlərinin qeyd olunmasına imkan yaradır;

Window – pəncərələrin açılması, bağlanması, sürüşdürülməsi, ölçülərinin dəyişməsi və s. kimi əsas əməliyyatları yerinə yetirir;

Help – sistemə daxil edilmiş arayış tipli informasiyanı almağa imkan yaradır.

Hər bir menyudan çıxış klaviaturadan **Esc** klavişini və ya ekranın boş sahəsində mausun sol klavişinin basılması ilə həyata keçirilir.

Proqramların daxil edilməsi və sazlanması zamanı istifadəçilərin iş rahatlığını təmin etmək üçün əsas pəncərəyə cədvəl 1.1-də göstərilmiş funksional düymələr daxil edilmişdir.

Əsas pəncərənin işçi zonasında klaviaturadan proqram mətni daxil edilir və ya mətn üzərində düzəlişlər aparılır. Bu zaman klaviaturanın aşağıda göstərilən klavişlərindən və klavişlər kombinasiyalarından istifadə etmək olar:

Page Up – bir səhifə yuxarı keçid;

Page Down – bir səhifə aşağı keçid;

Home – sətirin əvvəlinə keçid;

End – sətirin sonuna keçid;

BackSpace – klaviatura kursurunun solundakı simvolu silir;

Əsas pəncərəyə daxil edilmiş funksional düymələr

Düymə	Menyu sətri	Təyinatı
		Arayış məlumatları
	File/Open	Mövcud faylın aktivləşdirilməsi
	File/Save	Faylın yaddaşa yazılması
	Search/Find	Proqram mətnində fraqmentin axtarışı
	Search/Again	Proqram mətnində fraqmentin axtarışının davamı
	Edit/Cut	Qeyd olunmuş fraqmentin silinib bufer yaddaşına göndərilməsi
	Edit/Copy	Qeyd olunmuş fraqmentin nüsxəsinin bufer yaddaşına göndərilməsi
	Edit/Paste	Bufer yaddaşının tərkibinin kursorun vəziyyətindən bərpası
	Edit/Undo	Sonuncu əməliyyatın ləğvi
	Compile/Compile	İşçi zonada yerləşən proqramın kompilyasiyası
	Compile/Make	Müəyyən şərtlər daxilində proqramın kompilyasiyası
	File/Exit	Turbo Pascal mühitindən çıxış

Delete – klaviatura kursorunun sağındakı simvolu silir;

Ctrl+Y – klaviatura kursorunun yerləşdiyi sətri silir;

Enter – kursorun vəziyyətindən sətri parçalayaraq yeni sətərə keçid.

Cari vəziyyət sətri proqram mətni və onun icrası zamanı nəticəvi məlumatların əks etdirilməsi məqsədilə istifadə olunur.

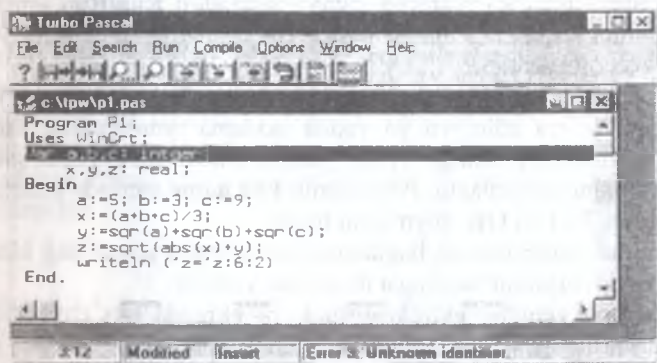
1.2. Proqram mətninin daxil edilməsi, sazlanması və icrası

Turbo Pascal mühitinə daxil olub, **File/New** əmrini seçin və mausun sol düyməsini onun üzərində basın. Ekranda **Noname00.pas** başlıqlı

boş işçi zona görünəcək. Klaviatura kursorunun vəziyyətindən aşağıda göstərilən proqram mətnini daxil edin. Qeyd edək ki, proqramın sazlanması prosesinin daha dərinədən dərk edilməsi üçün proqramda bilərəkdən bəzi səhvlər buraxılmışdır və bu səhvlər sonradan aradan qaldırılacaqdır.

```
Program P1;  
Uses WinCrt;  
Var a,b,c: integer;  
    x,y,z: real;  
Begin  
    a:=5; b:=3; c:=9;  
    x:=(a+b+c)/3;  
    y:=sqr(a)+sqr(b)+sqr(c);  
    z:=sqrt(abs(x)+y);  
    writeln('z='z:6:2)  
End.
```

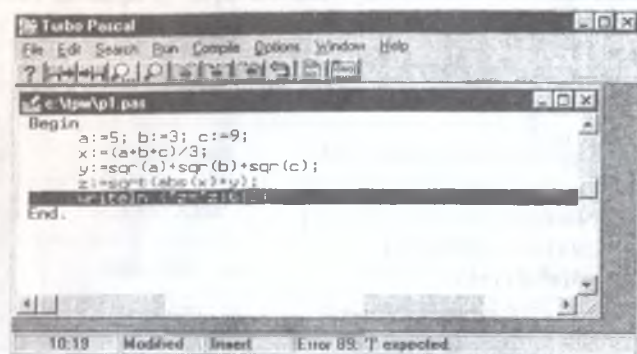
Menyu sətrindən **Run/Run** əmrini icra edin. Turbo Pascal kompilyatoru proqram mətninin 3-cü sətrində **Integer** operandında səhv aşkar edəcək. Həmin sətir şəkil 1.4-də göstərildiyi kimi qara zolaqla seçilir, sətir nömrəsi və səhv haqqında məlumat isə cari vəziyyət sətrində əks olunmuşdur.



Şəkil 1.4. P1 proqram mətnində səhvin aşkar edilməsi

Proqram mətnindəki **Integer** operandında «t» simvolundan sonra «e» simvolunu əlavə etməklə düzəliş aparın və yenidən menyu sətrindən **Run/Run** əmrini icra edin. Turbo Pascal kompilyatoru proqram mətninin

10-cu sətərdə **Writeln** operandında səhv aşkar edəcək (şəkil 1.5). Səhv haqqında məlumat cari vəziyyət sətərdə əks olunmuşdur.



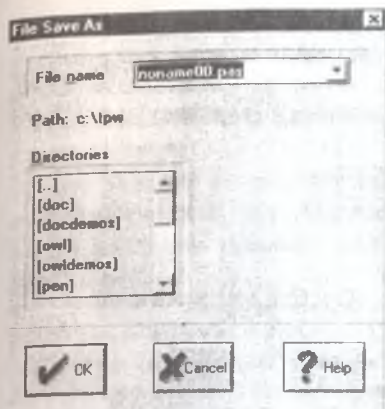
Şəkil 1.5. P1 proqram mətnində səhvin aşkar edilməsi

İşçi zonasının 10-cu sətiri ilə 19-cü sütununun kəsişməsində yerləşən klaviatura kursurunun mövqesindən operatorun yazılışında çatışmayan «**,**» simvolunu əlavə edin və təkrarən menyü sətərdən **Run/Run** əmrini icra edin. Proqramın icrasından alınan nəticə (**z=10,98**) ekranda yeni yaranan pəncərədə əks olunacaqdır.

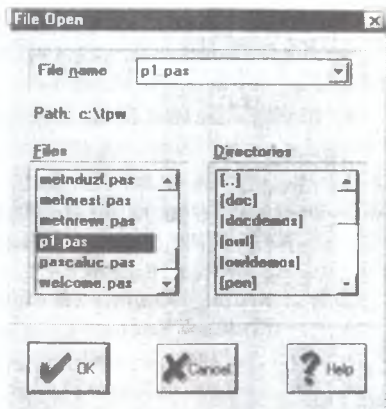
Proqram mətnini vinçesterdə yadda saxlamaq üçün **File/Save** əmrini seçin və icra edin, və ya yadda saxlama əməliyyatını **File/Save** funksional düyməsini basmaqla yerinə yetirin. Bu zaman ekranda şəkil 1.6-da göstərilən pəncərə açılacaq. Pəncərənin **File name** sətərdə proqrama ad verin (məsələn, **P1**) və **OK** düyməsini basın.

Proqram pəncərəsinin bağlanması pəncərənin yuxarı sağ küncündə yerləşən «**x**» düyməsinin basılması ilə yerinə yetirilir.

Proqramı yenidən aktivləşdirmək və ekranda əks etdirmək üçün **File/Open** əmrindən və ya **File/Open** funksional düyməsindən istifadə etmək olar. Əmr icra olunarkən, ekranda şəkil 1.7-də göstərilən pəncərə açılır. **Files** alt pəncərəsindəki siyahıdan **P1.PAS** proqramını seçin və mausun sol düyməsini bu ad üzərində basın. Bu zaman **File name** sətərdə bu ad əks olunacaq. **OK** düyməsini basdıqda, proqram mətni işçi zonaya çağırılacaqdır. Tələb olunarsa, proqram mətnində yeni düzəlişlər və əlavələr etmək olar.



Şəkil 1.6. File Save As pəncərəsi



Şəkil 1.7. File Open pəncərəsi

1.3. Laboratoriya işi №1. Turbo Pascal mühitində program mətnləri ilə əməliyyatlar

Aşağıdakı program modullarını mənasına fikir vermədən Turbo Pascal mühitində kompüterin əməli yaddaşına daxil edin, daxil edilmiş programı vizual nəzərdən keçirərək, buraxılan səhvləri aşkar edin və düzəldin. Programı icra edin. Yeni səhvlər aşkar edilərsə, programı sazlayın. Nəticə aldıqdan sonra programı yaddaşa yazın. Programı onun ilk sətrindəki identifikatorla adlandırın. Programın tərtib edilmə üsulu, onda istifadə olunan operatorların yazılış formaları və mənası sonrakı fəsilərdə izah ediləcəkdir.

```
1. Program L1_1;
   Uses WinCrt;
   Var a,b,x: real;
   Begin
     a:=1.256;
     b:=3.144;
     x:=(a+b)/2;
     writeln ('x=',x:5:3)
   End.
```

```
2. Program L1_2;
   Uses WinCrt;
   Var a,b,c,x: real;
   Begin
     a:=ln(5.67); b:=Pi*sqrt(a);
     c:=abs(a*b-sin(a)-sin(b));
     x:=arctan(a+b+c);
     writeln ('x=',x:7:3)
   End.
```

```

3. Program L1_3;
Uses WinCrt;
Var Radius:integer;
    C,S: real;
    i:byte;
Begin
    Radius:=5;
    for i:=1 to 10 do begin
        C:=2*Pi*Radius;
        S:=Pi*sqr(Radius);
        writeln ('Radius=',Radius:2, ' C=',C:5:2, ' S=',S:6:2);
        Radius:=Radius+1
    end
End.

```

```

4. Program L1_4;
Uses WinCrt;
Var y1,y2,y3:real;
    i:byte;
Begin
    for i:=1 to 16 do begin
        y1:=sqr(i);
        y2:=i*sqr(i);
        y3:=sqrt(i);
        writeln ('i=',i:2, ' y1=',y1:5:1, ' y2=',y2:6:1, ' y3=',y3:6:4);
    end
End.

```

```

5. Program L1_5;
Uses WinCrt;
Var x,y:real;
Begin
    x:=1.5;
    while x<10 do begin
        y:=sin(x)+cos(x)*sin(2*x);
        writeln (x:4:1, ' ',y:7:4);
        x:=x+0.5;
    end
End

```

2. ALQORİTMLƏR, ONLARIN XASSƏLƏRİ VƏ TƏSVİR ÜSULLARI

2.1. Alqoritmlərin tipik strukturları

Təbiətdə və gündəlik həyatımızda biz müxtəlifliyə malik çoxsaylı alqoritmlərlə rastlaşırıq. **Alqoritm dedikdə, hər hansı məqsədə çatmaq və ya hər hansı məsələni həll etmək üçün görülən işlərin ardıcılığı başa düşülür.**

Alqoritm sözü orta əsrlərdə indiki Özbəkistan Respublikasının ərazisində yerləşən Xarəzmdə yaşamış riyaziyyat, astronomiya, coğrafiya və tarix elmləri üzrə alim Əl-Xarəzminin (Əbu Abdullah Məhəmməd ibn Musa əl-Xarəzmi əl-Mədcusi) adı ilə bağlıdır. O, təxminən 783-cü ildə kəhin ailəsində anadan olmuş, 850-ci illərdə vəfat etmişdir. 813-cü ildə xəlifə Əl-Möminin «ağıllar sarayında» kitabxana müdiri vəzifəsinə qoyulmuş, bir neçə coğrafi səyahət etmişdir. Onun 37 fəsil və 116 cədvəldən ibarət ən görkəmli əsəri olan «Zidc əl-Sind-Hind» - «Astronomik cədvəllər» Qərbi Avropada sonralar inkişaf etmiş astronomik işlərin əsasını təşkil edir. Onun «Yerin şəkl kitabı» adlı digər məşhur kitabı coğrafiyaya həsr edilmişdir.



Bəzi alqoritmləri nəzərdən keçirək.

Alqoritm 2.1 (Tələbənin gündəlik həyatı). Bazar günlərindən savayı hər gün adətımız üzrə səhər yuxudan durduqdan sonra əl-üzümüzü yuyur, geyinir və səhər yeməyini yeyirik, evdən çıxıb, məktəbə yollanırıq. Dərslər başa çatdıqdan sonra evə qayıdıb, nahar edirik. Bir qədər istirahət edib, ev tapşırıqlarını yerinə yetiririk. Sonra bir qədər maraqlı televiziya verilişlərinə baxırıq, valideynlərimizlə müxtəlif söhbətlər edirik. Axşam-çağ axşam yeməyini yeyib, bir qədər açıq havada gəzişirik. Evə qayıdıb qiraət edirik və yatağımıza girib yatırıq.

Alqoritm 2.2 (Dərsin gedişi). Zəng vurularkən, sinifə daxil olub, öz yerimizdə otururuq. Müəllim gələn zaman ayağa qalxıb, onunla salamlaşırıq və onun göstərişi ilə yerimizdə əyləşib, dərsdə iştirak edirik. Dərsin sonunda zəng vurulduğu zaman ayağa qalxıb müəllimlə sağollaşırıq. Müəllim sinifdən çıxdıqda, dərs ləvazimatlarımızı yığışdırıb, sinifdən çıxırıq.

Hər iki alqoritmə bütün hərəkətlərimiz bir-birinin ardınca yerinə yetirilir. Belə alqoritmlər **xətti alqoritmlər** adlanır. Diqqət yetirin ki, bu ardıcılıqlarda hər hansı iki hərəkətimizdən birini digəri ilə əvəz etsək, mənasız bir ardıcılıq alına bilər.

Baxılan alqoritmlər ətrafında daha dərinlən düşünsək, bir çox hallarda onlardakı xətilik əlamətinin pozulduğunu aşkar edirik. Məsələn, dərslən evə qayıdarkən nahar yeməyi hələ hazır deyilsə, bir qədər istirahət edə, və ya televizıya ilə maraqlı veriliş varsa, ona baxa bilərik. Evə qonaq gəlsə, gəzintiye getməyib, vaxtımızı qonağa sərf edə bilərik. Ev tapşırığı verilməyibsə, bu vaxtı başqa bir işə sərf edə bilərik. Məktəbin zəngi xarab olubsa, dərslən sonunu müəllimin göstərişi ilə təyin edirik.

Doğrudan da, gündəlik məsələlərin həllində xətti alqoritmlərə çox az təsadüf edilir, lakin bu alqoritmik strukturun proqramlaşdırmada böyük rolu vardır. Həlli mürekkəb və çoxşaxəli olan məsələlərin həll alqoritmlərini məhz hər biri ümumiləşdirilmiş şəkildə təsvir edilmiş, xətti struktura malik mərhələlər ilə ifadə edirlər. Sonrakı iş prosesində bu mərhələlərdən hər biri bu və ya digər struktura malik alqoritmlərlə ifadə olunur. Beləliklə, biz aşkar etdik ki, bir çox məsələlərin həllində, görülən işlərin ardıcılığı hər hansı şərtin ödənilməsindən asılı olur.

Alqoritm 2.3 (Qəzet köşkü). Fərz edək ki, evdən məktəbə getmək üçün iki yolumuz vardır. Bu yollardan biri qısa, digəri isə uzundur. Uzun yolun yarısında qəzet köşkü qoyulmuşdur və hər gün köşkə yeni qəzet gətirilir. Gündəlik olaraq evdən çıxıb məktəbə getmək alqoritmını nəzərdən keçirək. Təbiidir ki, məktəbə gedərkən adətən biz qısa yolu seçirik və məktəbə tez çatırıq. Lakin biz bu alqoritmə həmişə riayət edirikmi? Əlbəttə ki, yox. Bizə səhər tezdən qəzet almaq lazımdırsa və biz dərslə gecikmiriksə, onda uzun yolu seçə bilərik. Qəzet almaq lazımdırsa, lakin vaxtımız azdırsa, biz qısa yolu seçirik və məktəbə vaxtında çatırıq. Əgər diqqət yetirdinizsə, biz «qəzet almaq» və «vaxt azlığı» kimi iki şərtədən istifadə etdik.

Alqoritm 2.4 (Bir səkidən digərinə keçid). Demək olar ki, hər gün bu alqoritmə rastlaşırıq və hər dəfə keçidimiz müvəffəqiyyətlə başa çatır. Məsələnin qoyuluşunu sadələşdirmək üçün, fərz edək ki, Bakının dar küçələrindən birindəyik, maşın yolu bir istiqamətlidir və tündə heç bir svetofor yoxdur. Alqoritm hansı mərhələlərdən ibarətdir? Yaxşı olar ki, bu məsələnin həll alqoritmı haqqında bir qədər sərbəst düşünesiniz.

Alqoritmın həllinin hansı mərhələlərdən ibarət olduğunu birgə aydınlaşdırıq. İlk növbədə həll alqoritmı belə təsəvvür edilir: «Maşının hərəkəti istiqamətinin əksinə baxırıq. Maşın yoxdursa, yolu keçirik. Maşın varsa, onun keçəcəyini gözləyirik və yol boşaldıqda yolu keçirik». Bu bizim birgə qurduğumuz ilk alqoritmədir. Diqqət yetirək, biz həmişə bu alqoritmə riayət edirikmi? Əlbəttə ki, yox. Əgər maşın varsa və o, dayanıbsa, onun hərəkətə gətirilib, yolu keçəcəyini biz xeyli gözləməli olarıq.

Həll alqoritmini bir qədər mükəmməlləşdirək: «Maşının hərəkəti istiqamətinin əksinə baxırıq. Maşın yoxdursa, yolu keçirik. Maşın varsa, onun hərəkətdə olduğunu təyin edirik. Maşın hərəkətdədirsə, onun keçəcəyini gözləyirik və yalnız bundan sonra yolu keçirik, maşın dayanıbsa yolu keçmək olar». Bu alqoritm haqqında düşünsək, onun heç də doğru olmadığını müəyyən etmək olar. Ola bilər ki, maşın hərəkətdədir, lakin o, bizim qarşımızdan keçmişdir və biz onun artıq arxa hissəsini görürük. Beləliklə həll alqoritmində yeni şərtlərin yoxlanılması zəruriyyəti yaranır. Alqoritmə maşının hərəkətdə olması ilə yanaşı, onun ön və ya arxa hissəsinin bizə tərəf olması, maşının bizdən uzaqda və ya yaxında olması, maşının sürəti, arxada digər maşınların olması və bu kimi şərtlər alqoritmə nəzərə alınmalıdır.

Alqoritm 2.5 (Saxta qızıl pul). 27 sayda qızıl puldan biri saxtadır və saxta pulun çəkisinin nisbətən az olması məlumdur. İkiqözlü çəki tərəzisdən ən azı neçə çəki yerinə yetirmək lazımdır ki, saxta qızıl pul aşkar edilsin.

Məsələnin həlli üçün müxtəlif alqoritmlər göstərmək olar, lakin yalnız bir alqoritm vasitəsilə məsələni ən az (üç) sayda çəki aparmaqla həll etmək olar. Bu alqoritmi nəzərdən keçirək:

1-ci çəki - 27 sayda qızıl pulu hər birində 9 sayda pul olmaqla üç hissəyə ayıraraq. Birinci hissəni tərəzinin bir gözüne, ikinci hissəni isə tərəzinin digər gözüne qoyaq. Tərəzidə natarazlıq varsa, saxta qızıl pul tərəzinin az çəki göstərən gözündəki hissədir. Əks halda, yəni tərəzi tarazdırsa, onda saxta qızıl pul yerdə qalan hissədir;

2-ci çəki - 9 sayda puldan ibarət saxta qızıl pul olan hissəni üç bərabər hissəyə ayıraraq. Hər hissədə 3 sayda qızıl pul olacaq. Birinci hissəni tərəzinin bir gözüne, ikincini isə tərəzinin digər gözüne qoyaq. Çəkilmə zamanı tərəzi tarazdırsa, deməli saxta qızıl pul yerdə qalan hissədir. Əks halda saxta qızıl pul tərəzinin az çəki göstərən gözündəki hissədir.

3-cü çəki - 3 sayda pulun birini tərəzinin bir gözüne, digərini o biri gözüne qoyaq. Tərəzi tarazdırsa, yerdə qalan qızıl pul saxtadır. Əks halda, saxta qızıl pul tərəzinin az çəki göstərən gözündədir.

Son üç sayda alqoritmlərdə müxtəlif şərtlərin ödənilməsi nəzərə alınmışdır. Şərtin ödənilməsindən asılı olaraq, bu və ya digər əməliyyatlar yerinə yetirilir. Belə strukturlara malik alqoritmlər **budaqlanan alqoritmlər** adlanırlar.

Alqoritm 2.6 (Xörək). Gün ərzində biz bir neçə dəfə xörək yeyirik. Yemək prosesinin alqoritmini nəzərdən keçirək. Fərz edək ki, qarşımızda bir boşqab sulu xörək vardır. Yeməyə başlayarkən biz qasıqı

doldurub, ağzımıza aparırıq. Bu əməliyyatı dəfələrlə təkrar edirik və boşqabdan xörək qurtardıqdan sonra qaşığı stolun üzərinə və ya boşqaba qoyub, yeməyi başa çatdırırıq.

Aydındır ki, təkrar olunan əməliyyatların sayı əvvəlcədən belli olmasa da, bu say sonludur. Qeyd etmək lazımdır ki, baxılan alqoritmə real şəraitdə təkrarlanmaların sayı heç də həmişə boşqabın tam boşalması ilə müəyyən edilmir. Bəzi hallarda biz artıq doyduğumuzdan, boşqabda xörəyin hələ qalmasına baxmayaraq, yeməyi saxlayırıq. Yemək zamanı xörəyin içərisinə kəpənək düşərsə, və ya təcili işlə qarşılaşırsaq, yenə də yemək prosesi kəsilir. Bəzən isə, tox olduğumuzdan və ya xörəyin rəngi və qoxusu xoşumuza gəlmirsə, yeməyə heç başlamırıq.

Alqoritm 2.7 (Sahildə dayaz quyu). Fərz edək ki, dəniz sahilində çox da dərin olmayan bir quyu qazıb, dənizdən vedrə ilə su götürərək bu quyuyu su ilə doldurmaq tələb olunur. Aydındır ki, bu proses sonsuz sayda davam edəcək və hər dəfə vedrə ilə quyuya su tökdükdə, su dolu növbəti vedrəni gətirənədek quyu yenidən boşalacaq, su torpağa hopub yenidən dənizə axıb gedəcək. Beləliklə, quyu heç vaxt su ilə dolmayacaq.

2.6 və 2.7 sayılı alqoritmərdə «xörəyi qaşığa doldurub ağza aparılması» və «su dolu vedrənin quyuya boşaldılması» kimi əməliyyatlarda təkrarlanmalar baş verir. Birinci halda təkrarlanmaların sayı sonlu, ikinci halda isə sonsuzdur. Belə alqoritmlər **dövri alqoritmlər** adlanır. Qeyd etmək lazımdır ki, kompüter alqoritmləri sonlu olmalıdır, əks halda proses sonsuz davam edər və programın icrası başa çatmaz.

Beləliklə, alqoritmlər xətti, budaqlanan və dövri strukturlardan birinə, və ya bu strukturların kombinasiyasına malik ola bilərlər.

2.2. Alqoritmlərin xassələri və təsvir üsulları

Alqoritmlərin 4 əsas xassəsinə aşağıdakılar aid edilir:

- **diskretlik** – məsələnin həlli prosesinin emal mərhələlərinə bölünməsi mümkünlüyü;

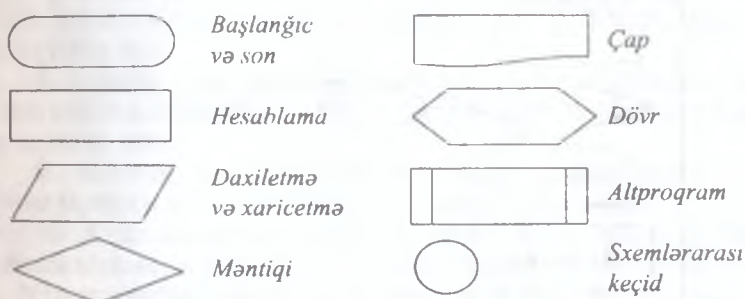
- **nəticəvilik** – məsələnin həlli üçün aparılan sonlu sayda əməliyyatdan sonra nəticə əldə edilməli və ya nəticənin olmaması haqqında məlumat verilməlidir;

- **müəyyənlik** – hər bir mərhələnin məzmununun və mərhələlərin yerinə yetirilmə ardıcılığının müəyyən olunması;

- **kütləvilik** – məsələnin həll alqoritmının ilkin verilənlərdən asılı olmaması mümkünlüyü.

Bunlardan savayı, alqoritmlərin **sadə** və **yığcam** olması, alqoritmə uyğun proqramın icrasında **operativliyin** gözlənilməsi də vacib şərtlərdəndir. Alqoritm elə qurulmalıdır ki, sonradan o, digər istifadəçilər, uzun müddət keçdikdən sonra isə hətta alqoritmı quran şəxs üçün asan qavranılsın. Alqoritmlərdə əlavə, lazım olmayan əməliyyatların aparılmasına yol verilməməlidir. Doğrudur, müasir fərdi kompüterlərin sürəti elə böyükdür ki, belə «lazımsız» bir-iki əməliyyatın proqramın icra vaxtına təsiri həddindən artıq cüzdür. Lakin onların çoxsaylı təkrar olunan dövr içərisində olması, proqramın icrasını xeyli ləngidər.

Alqoritmləri müxtəlif üsullarla təsvir etmək olar. 2.1-2.7 alqoritmləri təbii dildə təsvir edilmişdir. Bu üsuldən adətən böyük həcmli alqoritmləri ümumiləşdirilmiş şəkildə təsvir etmək üçün istifadə edirlər. Alqoritmlərin təsviri üçün ən geniş yayılmış üsul **blok-sxem** üsuludur. Belə təsvir zamanı həndəsi fiqurlardan istifadə olunur (şəkil 2.1). Bloklar bir-biri ilə istiqamətlənmiş düz xətt parçaları ilə birləşdirilir. Əgər parçada istiqamət göstərilməyibsə, onda istiqamət sxem üzrə yuxarıdan aşağıya, və ya soldan sağa qəbul olunur.



Şəkil 2.1. Alqoritmın blok-sxem üsulu ilə təsvirində istifadə olunan əsas həndəsi fiqurlar

Alqoritmlərə aid bəzi riyazi məsələləri nəzərdən keçirək. Belə məsələlər sonrakı fəsillərdə daha ətraflı öyrəniləcəkdir.

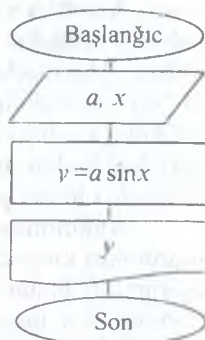
Alqoritm 2.8. İxtiyari x arqumenti üçün $y = a \sin x$ funksiyasının qiymətini tapın. a sabit ədəddir.

Həlli. Funksiyanın qiymətini hesablamaq üçün a və x dəyişənlərinin qiymətləri əməli yaddaşa daxil edilməlidir. Bu qiymətlər məlum olarsa, y funksiyasının qiymətini hesablamaq olar. Hesablamaların nəticəsi

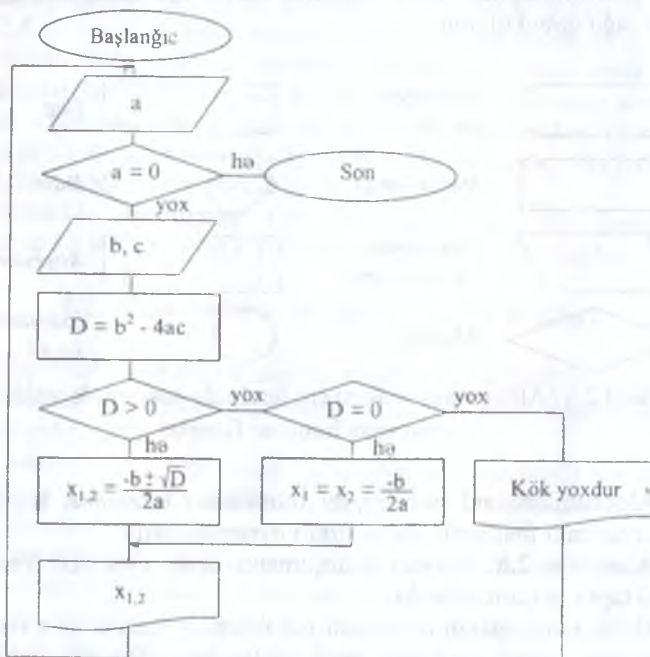
əmali yaddaşa olacaq. Bu nəticənin çap edilməsi ilə alqoritm başa çatır. Alqoritmın blok-sxem təsviri şəkil 2.2-də göstərilmişdir.

Alqoritm 2.9. a ($a \neq 0$), b və c dəyişənlərinin ixtiyari qiymətlərində $ax^2+bx+c=0$ kvadrat tənliyinin köklərini tapın.

Həlli. Dəyişənlərin müxtəlif qiymətlərində məsələnin həlli üçün dövr təşkil edilmişdir. Dövr daxilində a dəyişəninin qiyməti daxil edilir. Bu qiymət sıfırdırsa dövrədən çıxış baş verir. Əks halda, b və c dəyişənlərinin qiymətləri daxil edilir və tənliyin diskriminantı hesablanır. $D > 0$ isə x_1 və x_2 , $D = 0$ isə x_1 ($x_2 = x_1$) qiymətləri hesablanır və çap edilir. $D < 0$ isə tənliyin kökü olmaması haqqında məlumat verilir. Alqoritmın blok-sxem təsviri şəkil 2.3-də göstərilmişdir.



Şəkil 2.2



Şəkil 2.3

2.3. Laboratoriya işi №2. Alqoritmın təbii dildə və blok-sxem üsulları ilə təsviri

1. Təbii dildə təsvir edilmiş 2.1-2.7 sayılı alqoritmləri blok-sxem üsulu ilə təsvir edin.

2. Təbiətdə və gündəlik həyatınızda rast gəldiyiniz xətti, budaqlanan və dövri alqoritmlərdən hər birinə 2 misal göstərin. Onların həll alqoritmlərini təbii dildə və blok-sxem üsulu ilə təsvir edin.

3. Üç x , y və z ədədləri verilmişdir. Bu ədədlərdən ən böyüyünün tapılması alqoritmını blok-sxem üsulu ilə təsvir edin.

4. $a_1x + b_1y = c_1$ və $a_2x + b_2y = c_2$ düz xətlərinin kəsişmə nöqtəsinin koordinatlarını tapın.

5. Çevrənin radiusu verilmişdir. Bu çevrənin diametrinin, uzunluğunun və dairə sahəsinin tapılması alqoritmını blok-sxem üsulu ilə təsvir edin.

6. Rombun diaqonalları verilmişdir. Onun perimetrinin və sahəsinin tapılması alqoritmını blok-sxem üsulu ilə təsvir edin.

7. Düzgün paralelepipedin üç tilinin uzunluqları verilmişdir. Onun yan, tam səthinin sahələrinin və həcmnin tapılması alqoritmını blok-sxem üsulu ilə təsvir edin.

8. Silindrin oturacağıın radiusu və hündürlüyü verilmişdir. Onun yan, tam səthinin sahələrinin və həcmnin tapılması alqoritmını blok-sxem üsulu ilə təsvir edin.

9. Kürənin radiusu verilmişdir. Onun səthinin sahəsinin və həcmnin tapılması alqoritmını blok-sxem üsulu ilə təsvir edin.

10. N tam ədədi verilmişdir. Bu ədədin M faizinin hesablanması alqoritmını blok-sxem üsulu ilə təsvir edin. Məsələnin həllində dövr təşkil edin, N tam ədədinin daxil olunmasını və M faizinin hesablanmasını dövrün daxilində yerinə yetirin. Dövrədən çıxışı $N=0$ şərtinin ödənilməsi ilə həyata keçirin.

3. PASCAL DİLİNİN ƏSASLARI

3.1. Pascal dilinin əsas anlayışları

Pascal proqramlaşdırma dilinin ilk versiyasını 1968-ci ildə İsveçrə alimi, professor **Niklaus Virt** yaratmış və o, bu dili ilk cəmləyici qurğunu icad etmiş, görkəmli Fransa alimi Blez Paskalın şərəfinə onun adı ilə adlandırmışdır. Pascal dili azsaylı baza anlayışları əsasında qurulması, sadə sintaksisə malik olması, sadə kompilyator vasitəsilə proqram mətninin məşin kodlarına çevrilməsi mümkünlüyü ilə digər proqramlaşdırma dillərindən fərqlənir.

Görkəmli alim və pedaqoq Niklaus Virt 1968-ci ildən İsveçrənin Sürix şəhərindəki Ali Texniki Məktəbin (ETH – Eidgenoessische Technische Hochschule) informatika üzrə professorudur. 1999-cu ilin aprelindən vəzifəsindən istefa edib. Onun əsərləri Türiinq adına mükafata, proqram təminatının yaradılması sahəsində görkəmli nailiyyətlərə görə mükafata, Leonardo–da–Vinci medalına və s. layiq görülüb. O, İsveçrə və Amerika texniki akademiyalarının üzvüdür. N.Virtin əsərləri əsasən pedaqoji yönümlüdür. O, elə alqoritmik dil yaratmaq əzmində olmuşdur ki, bu dil proqramlaşdırma prosesini sadələşdirsən, bu dildə yazılan proqram mənləri qavranmaq üçün aydın və rahat olsun. Maraqlıdır ki, 1969-cu ildə kompilyatorun ilk versiyasını əvvəlcə Fortran-da, sonra isə Pascal-da yaradan E.Marme N.Virtin tələbəsi olmuşdur.



Sonralar Pascal proqramlaşdırma dili uzunmüddətli inkişaf yolu keçmiş, ona yeni genişlənmələr və imkanlar əlavə edilmişdir. Bu sahədə Amerikanın Borland International firmasının rolunu xüsusi qeyd etmək lazımdır. Pascal dilinin son versiyalarında obyekt yönümlü proqramlaşdırmanın imkanlarından geniş istifadə olunur.

Hər bir proqramlaşdırma dilində olduğu kimi, Pascal dilinin də öz əlifbası var. Bu əlifbaya aşağıdakı simvollar daxildir:

- böyük və kiçik latın hərfləri;
- 0-dan 9-dək ərəb rəqəmləri;
- +, -, *, /, :, ;, ., ,, >, <, =, (,), ', {, }, _ və s. kimi xüsusi simvollar.

Konstantalar, dəyişənlər və operatorlar proqramın əsas elementləridirlər. **Konstantalar** proqramın icrası zamanı öz qiymətlərini dəyişmir, sabit saxlayırlar. **Dəyişənlər** isə, konstantalardan fərqli olaraq, proqramın icrası zamanı müxtəlif qiymətlər ala bilirlər və bu qiymətlər dəyişənin

təyin edilmiş tipinə uyğun qəbul edilir. **Operatorlar** proqramın icra edəcəyi əməliyyatları təyin edirlər.

Pascal-proqrama və onda istifadə olunan dəyişənlərə müəyyən şərtlər daxilində ad verilir. Belə adlar **identifikatorlar** adlanırlar. İdentifikator latın hərflərindən, rəqəmlərdən və «_» simvolundan təşkil olunur və rəqəmlə başlaya bilməz. İdentifikatorda istifadə olunan simvolların sayına məhdudiyyət qoyulmur, lakin identifikasiya zamanı elə adlardan istifadə etmək tövsiyə edilir ki, bu adlar dəyişənin istifadə olunma məqsədini mənaca əks etdirdsin. İki və daha çox dəyişənə eyni ad vermək olmaz. İdentifikatorlarda eyni böyük və kiçik latın hərfləri eyni mənə daşıyır. Proqramı və dəyişənləri adlandırarkən müəyyən mənə daşıyan xüsusi təyinatlı xidməti sözlərdən istifadə etmək olmaz. Belə xidməti sözlərdən bir qismi aşağıda göstərilmişdir:

And	Do	If	Not	Set	Uses
Asm	Downto	Implementation	Object	Shl	Var
Array	Else	In	Of	Shr	While
Begin	End	Inline	Or	String	With
Case	Exports	Interface	Packed	Then	Xor
Const	File	Label	Procedure	To	
Constructor	For	Library	Program	Type	
Destructor	Function	Mod	Record	Unit	
Div	Goto	Nil	Repeat	Until	

Konstantalar və dəyişənlər üzərində aşağıdakı əməliyyatları yerinə yetirmək olar:

- hesabi: **+** (toplama), **-** (çıxma), ***** (vurma), **/** (bölmə), **div** (qalıqın atılması ilə tam bölmə), **mod** (tam bölünmədən alınan qalıq);
 - nisbət: **>** (böyük), **<** (kiçik), **>=** (böyük bərabər), **<=** (kiçik bərabər), **=** (bərabər), **<>** (bərabər deyil);
 - məntiqi: **not** (inkar), **or** (məntiqi toplama, VƏ YA), **and** (məntiqi vurma, VƏ), **xor** (istisna VƏ YA);
 - çoxluqlar: ***** (kəşimə), **+** (birləşmə), **-** (çıxma), **in** (mənsubiyyət).
- Pascal-proqramım şərti olaraq üç hissəyə bölmək olar:

Program	proqramın başlığı;
{ \$... }	kompilyatorun global direktivləri;
Uses	qoşulan kitabxanalar;
Label	global nişanlar;

Const	qlobal konstantalar;
Type	qlobal tiplər;
Var	qlobal dəyişənlər;
•	prosedur və funksiya mətnləri bölməsi;
Procedure (Function)	prosedurun (funksiyanın) başlığı;
Label	lokal nişanlar;
Const	lokal konstantalar;
Type	lokal tiplər;
Var	lokal dəyişənlər;
Begin ... End;	prosedurun (funksiyanın) əsas bloku;
•	proqramın əsas blok bölməsi.
Begin ... End.	proqramın əsas bloku.

Proqramın əsas bloku **Begin** və **End** operatorlar mötərizəsi içərisində yazılır. Bu bölmədə qarşıya qoyulan məsələnin həll algoritminə uyğun, icra olunan operatorlar ardıcılığı göstərilir. Pascal-proqramın digər bölmələri təsvir xarakterlidir. Sonuncudan savayı bütün bölmələr ehtiyac olmadıqda proqram mətnində yazılmaya bilər. Hər bir bölmənin və operatorun sonunda «;» - «nöqtəli vergül» simvolu qoyulur. Proqramın sonu «.» - «nöqtə» simvolu ilə işarə edilir və bu simvoldan sonrakı operatorları Pascal kompilyatoru qəbul etmir.

Proqramın hər hansı bir hissəsində onun icrasına heç bir təsir göstərməyən şərtlərdən istifadə etmək olar. **Şərtlər** fiqurlu mötərizələr ({ və }) içərisində yazılır və onların yazılışında dilin əlifbasına daxil olmayan istənilən simvollarından istifadə etmək olar.

3.2. Hesabi və məntiqi ifadələr, mənsub etmə operatoru

Hesabi ifadələr bir-biri ilə hesab əməlləri ilə əlaqələndirilmiş konstantalardan, dəyişənlərdən, standart funksiyalardan və kiçik mötərizələrdən təşkil olunur. Hesabi ifadələrin qiymətlərini hesablayarkən, riyaziyyatda qəbul edilmiş aşağıdakı üstünlük dərəcələri nəzərə alınmalıdır:

- mötərizələr;
- standart funksiyalar;
- soldan sağa ardıcıl vurma və bölmə;
- soldan sağa ardıcıl toplama və çıxma.

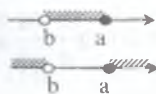
Hesabi ifadənin yazılışında iç-içə mötərizələrdən istifadə olunubsa, əməliyyatların icrası sağdan sola mötərizələr boyunca aparılır.

Hesabi ifadənin hesablanmış qiymətinin hər hansı dəyişənə verilməsi «:=» - mənsubetmə işarəsindən istifadə etməklə həyata keçirilir. Hesabi ifadələrin və mənsubetmə operatorunun yazılışına aid bəzi misalları nəzərdən keçirək. Hesabi ifadələrdə $\sin x$, $\cos x$, $\arctg x$, $|x|$, x^2 , $\sqrt{x}$ üçün standart funksiyalardan istifadə olunmuşdur.

Riyazi	Pascal
1. $y = ax + b$	$y := a * x + b;$
2. $y = \frac{a+b}{2} \sin x$	$y := (a+b)/2 * \sin(x);$
3. $y = \frac{a \ln x + b}{\cos x}$	$y := (a * \ln(x) + b) / \cos(x);$
4. $y = \frac{ a + b(\sin x + \ln x)}{2 + \cos x}$	$y := (abs(a) + b * (\sin(x) + \ln(x))) / (2 + \cos(x));$
5. $y = \ln(\ln x) + 0,5 $	$y := abs(\ln(\ln(x)) + 0.5);$
6. $y = \arctg \frac{x^2}{\sqrt{a}}$	$y := arctan(sqr(x)/sqrt(a));$

Məntiqi ifadələr nisbət və məntiqi əməliyyat işarələri ilə əlaqələndirilmiş konstantalar, dəyişənlər, standart funksiyalar və mötərizələrdən təşkil olunur. Bəzi məntiqi ifadələri nəzərdən keçirək:

Riyazi	Pascal
1. $y \geq \frac{a+b}{2} \sin x$	$y >= (a+b)/2 * \sin(x);$
2. $\sin x \cdot \cos a \neq x + a$	$\sin(x) * \cos(a) \neq x + a$
3. $x + a \sin x = a + b \cos x$	$x + a * \sin(x) = a + b * \cos(x)$
4. $b < x \leq a$	$(x > b) \text{ and } (x \leq a)$
5. $x \geq a \text{ və ya } x < b$	$(x >= a) \text{ or } (x < b)$
6. <i>Faylın sonu deyil</i>	$\text{not eof}(\text{File})$



3.3. Laboratoriya işi № 3. Hesabi və məntiqi ifadələr

Aşağıdakı riyazi funksiyaların Pascal dilində hesabi ifadələrini yazın;

1. $y = x^3 + ax^2 + b^4 \ln(|abx| + 2)$
2. $y = \frac{x+a^3}{3} + \sin(a+x)^2$
3. $y = \sin \frac{x+a^2}{2} + \cos \frac{x^2+a}{2}$
4. $y = \left(\frac{x+0,5}{2} \right)^2 + \cos(x+ab^2)$
5. $y = \frac{\cos^3(x+a) - 2(a^2-x)}{1+a^2+x^2}$
6. $y = \ln \frac{1-\sqrt{x^2+|\sin a|}}{x^2+\cos a+2}$
7. $y = \operatorname{tg} x^{2-a} + \ln(|x-a|+2)$
8. $y = \frac{a}{2} \sin x^2 + \frac{b}{2} \cos^2 x$
9. $y = a \operatorname{tg}^2 x^3 - b \operatorname{arctg} \frac{1}{x^2+1}$
10. $y = \frac{1+a \sin^3 x^2}{\operatorname{tga} - \sin^2 x + 1}$
11. $y = \frac{ax^2 + \sqrt{a^2+1}}{x^2+1}$
12. $y = \frac{\ln(a^2x^2+3)}{2+a^4x^2}$
13. $y = \frac{a + \sin x^2 + 1,5}{2+b^2|\cos x|}$
14. $y = \frac{|\cos \frac{x}{2}| + |\sin \frac{x}{2}|}{1,5 + \cos^2 \frac{x+a}{2}}$
15. $y = \sqrt[3]{(x+a)^2 + \sin^2 \frac{|x|}{2}}$
16. $y = \operatorname{arctg} \frac{1+x}{1+\sqrt{x^2+1}}$
17. $y = \ln^2 |x + \sin^2 x + 1|$
18. $y = \frac{a-x^2}{2a^4+1} \sqrt{|\cos a|}$
19. $y = \frac{ab \sin \sqrt{|x|}}{1+a^2b^2}$
20. $y = \sqrt{|a + \ln(x^4+1)|}$

Aşağıdaki məntiqi ifadələri Pascal dilində yazın:

21. $\sin^2 x < \frac{1}{1+x^2}$
22. $\cos x - \sin x \leq 0,3x$
23. $x^{2-a} < y < x^{2+a}$
24. $a \leq x - \sin x^2 \leq b$
25. $\frac{a-b \sin x}{2+x} < x \cos a$
26. $\sin a \leq \operatorname{tg} \frac{a-x}{a+x} < \cos a$
27. $\frac{3-a^2}{\sin x} > 2$ və ya $\sqrt{1-x^2} < 9a$
28. $3a^2 - b \leq x + y \leq 3a^2 + b$
29. $\sqrt{a} \sin x < a^2 x$ və ya $x^2 \cos a \geq a$
30. $\frac{a+b \sin^2 x}{2+x} < 1,5$ və ya $ax^2 + b < \sin^2 x$

3.4. Pascal-proqramın elanlar bölməsi

Pascal-proqram mətninin ilk sətirində onun başlığı yazılır. Başlıq **Program** sözü ilə başlayır və ondan sonra proqrama ad verilir. Bu sətirin proqram mətnində yazılması heç də məcburi deyildir. Lakin başlığın proqram haqqında ilk təəssürat yaratdığını, hansı məqsəd daşdığını göstərməsini nəzərə alaraq, ondan istifadə etmək tövsiyə olunur. Proqrama verilən ad identifikator olub, onun bütün xassələrini ödəməlidir.

Qlobal direktivlər bölməsi proqramın sonrakı translyasiyası zamanı kompilyatorun iş rejimini təyin etmək üçün istifadə olunur. Kompilyatora verilən bütün göstərişlərin yazılışı proqram mətnində «{\$» simvolları ilə başlayır, göstərişlərin sonu isə «{» simvolu ilə ifadə olunur.

Uses operatorunun vəzifəsi kitabxanalardakı sistem modulların proqram mətninə əlavə edilməsindən ibarətdir. Pascal-ın kitabxanalarına çoxlu modul daxil edilmişdir. Hər bir modul ayrılıqda bir proqram olub, unikal adı malikdir. **Uses** operatoru proqramda yalnız bir yerdə, proqramın bütün operatorlarının və bölmələrinin önündə yazılmalıdır. Operatorun yazılışında bütün kitabxanalar **Uses** sözündən sonra göstərilir. Məsələn,

Uses WinCrt;

Label – nişanların təsviri bölməsində bir-birindən «,» simvolu ilə ayrılan, təkrarlanmayan keçid nişanlarının adları göstərilir. Nişanlar 0-dan 9999-dək tam ədədlə, simvollar sətri və ədəd-simvol qarışığı ilə ifadə oluna bilərlər. Məsələn,

Label 3, n, m100, h1;

Const – konstantaların təsviri bölməsində proqram mətnində istifadə olunan konstantalar göstərilir. Konstantalar bir-birindən «;» simvolu ilə ayrılırlar. Məsələn,

**Const R=5; M=200;
St='Azerbaijan';**

Type – tiplərin təsviri bölməsində istifadəçilərin təyin etdikləri yeni tiplər göstərilir. Məsələn,

**Type Yazı = Record
Ad, Fam: string[15];
Telefon: string[6]**

End;

Var – qlobal dəyişənlərin təsviri bölməsində proqramda istifadə olunan dəyişənlər təsvir edilir, onlardan hər birinin tipi göstərilir. Növbəti fəsilərdə dəyişənlərin tipləri təfəsilatı ilə öyrəniləcəkdir. Hələ ki, tam dəyişənlər üçün **Integer**, həqiqi dəyişənlər üçün **Real**, məntiqi dəyişənlər

üçün isə **Boolean** tiplərini qəbul edək. Qeyd edək ki, məntiqi tip dəyişən **True** (doğru) və **False** (yalan) qiymətlərindən birini ala bilər. Məsələn,

```
Var    a, b, c: Integer;  
       x, y:   Real;  
       t:      Boolean;
```

Procedure və **Function** – prosedurlar və funksiyaların təsviri blo-
kunda istifadəçilərin yaratdıqları altproqramlar formalaşdırılır. Sonradan
əsas proqramın icrası zamanı altproqramlara giriş həm əsas proqramın
istənilən hissəsindən, həm də digər prosedur və funksiyalardan həyata
keçirilə bilər. Proqram kimi prosedur və funksiya da ad verilir. Onlar
parametrik şəkildə təsvir edilir və parametrlərin müxtəlif qiymətlərində
onları icra etmək olar. Hər bir prosedurun və funksiyanın öz lokal nişanları,
konstantaları və dəyişənləri ola bilər, onlar prosedur və ya funksiyanın
tərkibində uyğun olaraq **Label**, **Const**, **Type** və **Var** bölmələrində təsvir
olunurlar. Əsas hissə isə **Begin** və **End** operatorlar mötərizəsi içərisində
yazılır. Əsas hissənin sonunu «;» simvolu təyin edir. Qeyd etmək lazımdır
ki, funksiya vasitəsilə yalnız bir, prosedur vasitəsilə isə bir neçə çıxış
dəyişəninin qiymətini hesablamaq mümkündür:

Funksiyanın ümumi yazılış forması belədir:

```
Function F(Q1:T1; Q2:T2; ...):T;  
{Lokal dəyişənlərin təsviri}  
Begin
```

```
    P1:=...; P2:=...; ...;  
    F:=...;
```

```
End;
```

Burada, **F** – funksiyanın adı, **Q1**, **Q2**, ... giriş parametrləri, **T1**, **T2** –
bu parametrlərin tipləri, **T** – funksiyanın tipi, **P1**, **P2** – lokal dəyişənlərdir.
Funksiyanın sonunda nəticə funksiyanın adı ilə ifadə olunan dəyişənə
mənsub edilir. Funksiyadan istifadə qaydası aşağıdakı misalda göstərilmiş-
dir.

Misal 3.1. Eni *a*, uzunluğu isə enindən 3 dəfə böyük olan düzbu-
caqlının sahəsinin tapılmasını funksiya ilə istifadə etməklə yerinə yetirin.

Həlli: Proqram aşağıdakı kimi tərtib edilə bilər:

```
Program DSahe;  
Uses WinCrt;  
Var a,s: real;  
Function Sahe(m:real):real;  
Var n: real;  
Begin
```

```
    n:=3*m;      Sahe:=m*n
```

End;
{Əsas blok}

Begin

a:=5; s:=Sahe(a);

{S dəyişənin qiymətinin çap edilməsi}

End.

Prosedurun ümumi yazılış forması belədir:

Procedure F(Q₁:T₁; Q₂:T₂; ...; Var Q_n:T_n; Q_{n+1};T_{n+1}; ...);

{Lokal dəyişənlərin təsviri}

Begin

P₁:...; P₂:...; ...

Q_n:...; Q_{n+1}:...; ...

End;

Burada, F – prosedurun adı, Q₁, Q₂, ... giriş parametrləri, Q_n, Q_{n+1}, ... çıxış parametrləri, T₁, T₂, ..., T_n, T_{n+1} onların tipləri, P₁, P₂ – lokal dəyişənlərdir. Prosedura müraciət edərkən, onun adını, giriş parametrlərinin qiymətlərini və hesablamaların nəticəsini özündə saxlayan çıxış parametrlərinə uyğun dəyişənləri göstərmək lazımdır. Başqa sözlə prosedura müraciət

F(R₁, R₂, ..., R_n, R_{n+1}, ...);

kimi yerinə yetirilir. Burada R₁, R₂, ..., R_n, R_{n+1}, ... dəyişənləri Q₁, Q₂, ..., Q_n, Q_{n+1}, ... giriş və çıxış parametrlərinə uyğun ardıcılıqda qəbul edilmiş giriş və çıxış dəyişənləridir.

Misal 3.2. Eni a, uzunluğu isə enindən 3 dəfə böyük olan düzbucaqlının perimetrinin və sahəsinin tapılmasını prosedurdan istifadə etməklə yerinə yetirin.

Həlli: Program aşağıdakı kimi tərtib edilə bilər:

Program SaheVePerimetr;

Uses WinCrt;

Var a, P, S: real;

Procedure SP(m:real; Var Pd,Sd:real);

Var n: real;

Begin

n:=3*m; Pd:=2*(m+n); Sd:=m*n

End;

{Əsas blok}

Begin

a:=5; SP(a, P, S);

{P və S dəyişənin qiymətinin çap edilməsi}

End.

3.5. Daxiletmə və xaricetmə standart prosedurları

Pascal dilində daxiletmə və xaricetmə əməliyyatlarını yerinə yetirən xüsusi operator yoxdur və belə əməliyyatlar **Read**, **Readln**, **Write** və **Writeln** standart prosedurlarının köməyi ilə həyata keçirilir. Bu prosedurları digər standart prosedurlar kimi əvvəlcədən təsvir etmək lazım deyil və onlardan program mətninin istənilən yerində istifadə etmək olar.

Daxiletmə və xaricetmə prosedurları Pascal dilinin bir çox prosedurlarından fərqli xüsusiyyəti ondadır ki, bu prosedurlarda ixtiyari sayda parametr göstərmək olar. Bu parametrlər prosedurların adından sonra, kiçik mötərizələr içərisində siyahı şəklində göstərilir. Məsələn,

Writeln ('Turbo Pascal');

xaricetmə prosedurunda bir parametr,

Writeln ('Radius=', R);

prosedurunda iki parametrdən istifadə olunmuşdur. Birinci **Writeln** proseduru icra olunduğu zaman ekrana «Turbo Pascal» sözü, ikinci **Writeln** proseduru icra olunduğu zaman isə ekrana «Radius=» ifadəsi və onun ardınca əvvəlcədən təyin olunmuş Real və ya Integer tipli R dəyişənin qiyməti çıxarılacaqdır.

Verilənlərin xaric edilməsi məqsədilə **Write** prosedurundan istifadə olunubsa, onda növbəti xaricetmə əvvəlkinin ardınca yerinə yetiriləcəkdir. **Writeln** prosedurundan istifadə edildikdə isə, onun icrasından sonra yeni sətirə keçid baş verəcək, beləliklə növbəti xaricetmə yeni sətirdən yerinə yetiriləcəkdir. **Writeln** prosedurunda heç bir parametr göstərilməyibsə, xaricetmə aparılmır, yalnız yeni sətirə keçid baş verir. Məsələn,

Program Xaricetme;

Uses WinCrt;

Const R=5;

Begin

Write ('Turbo ');

Writeln('Pascal');

Write ('Baku ');

Write ('Azerbaijan');

Writeln;

Write ('Radius=',R,' Diametr=',2*R);

end.

Pprogramının icrasından sonra, ekranda aşağıdakı nəticələr alınacaq:

Turbo Pascal

Baku Azerbaijan

Radius=5 Diametr=10

Həqiqi dəyişənlərin qiymətlərinin xaric edilməsi zamanı, onlar üçün çıxış formatı göstərilə bilər. Məsələn,

Writeln (R:8:2);

xaric etmə prosedurundan istifadə olunarsa, onun icrası zamanı R dəyişəninə qiyməti üçün 8 mövqə götürüləcək. Bu mövqelərdən 2-si ədədin kəsr hissəsi, 5-i tam hissəsi, 1-i isə tam hissəni kəsr hissədən ayıran «nöqtə» üçün istifadə olunacaq.

Programın icrası zamanı verilənlərin daxil edilməsi **Read** və **Readln** prosedurlarından biri ilə həyata keçirilir. Read prosedurundan istifadə edərkən, növbəti daxil etmə prosedurundakı verilənlər kursurun vəziyyətindən, **Readln** prosedurundan istifadə edərkən isə, yeni sətirdən daxil edilir. Bu prosedurların hər birində birdəfəyə bir neçə dəyişənin qiymətini daxil etmək olar. Bu halda verilənlər bir-birindən ya «boşluq» simvolu ilə ayrılmalı, ya da hər veriləndən sonra Enter klavişi basılmalıdır. Nə qədər ki, prosedurda göstərilən bütün dəyişənlərin qiymətləri daxil edilməyib, növbəti operatora keçid baş verilməyəcək. Məsələn,

Program Daxiletmə;

Uses WinCrt;

Var a,b,x: integer;

Begin

Write ('a='); Readln (a);

Write ('b='); Readln (b);

x:=a+b;

Writeln('a+b=',x);

End.

Programın icrasından sonra aşağıdakı nəticə alınacaq:

a=5

b=8

a+b=13

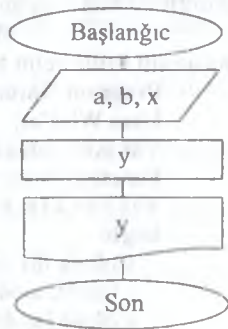
Aşağıda xətti struktura malik Pascal-programının tərtibinə dair bir neçə misal göstərilmişdir.

Misal 3.3. $y = a \sin x + \ln(b^2 + 2x^2 + 3)$ funksiyasının hesablanması üçün məsələnin həll algoritminin blok-sxemini qurun və Pascal dilində programını yazın.

Həlli: Məsələnin həll algoritminin blok-sxemi şəkil 3.1-də verilmişdir.

Program Xetti_Funksiya;

Uses WinCrt;



Şəkil 3.1

Var a, b, x, y: real;

Begin

Write ('a, b ve x = '); Readln (a,b,x);

y:=a*sin(x)+ln(sqr(b)+2*sqr(x)+3);

Writeln('y=',y:7:4);

End.

Nəticə:

a, b ve x = 4 2 7.5

y= 8.5353

Misal 3.4. Verilmiş 4 rəqəmli natural ədədin rəqəmlərinin tərs düzülüşündən alınan natural ədədi tapın.

Həlli. Fərz edək ki, $N=5387$ natural ədədi verilmişdir və onun rəqəmlərinin tərs düzülüşündən alınan natural ədədi ($M=7835$) tapmaq tələb olunur. Məsələnin həlli üçün verilmiş ədədin rəqəmləri belli olmalıdır. Ədədi 1000-ə tam bölsək, qismətdə alınan ədəd ($k1d$) verilmiş ədədin birinci rəqəmini göstərəcək: $k1d=N:1000=5387:1000=5$. Tam bölünmədən alınan qalıq ($k1m$) isə 387 olacaqdır: $k1m=387$.

Alınmış $k1m$ ədədini 100-ə tam bölsək, qismətdə alınan ədəd ($k2d$) verilmiş N ədədinin ikinci rəqəmini göstərər: $k2d=k1m:100=387:100=3$. Tam bölünmədən alınan qalıq ($k2m$) isə 87 olar: $k2m=87$.

Alınmış $k2m$ ədədini 10-ə tam bölək. Bu zaman qismətdə alınan ədəd ($k3d$) verilmiş N ədədinin üçüncü rəqəmini göstərəcək: $k3d=k2m:10=87:10=8$. Tam bölünmədən alınan qalıq ($k3m$) isə 7 olacaqdır və bu ədəd verilmiş N ədədinin dördüncü rəqəmidir: $k3m=7$.

Verilmiş N natural ədədinin rəqəmlərinin tərs düzülüşündən alınan ədədi (M) aşağıdakı düsturla tapmaq olar:

$$M=k3m*1000+k3d*100+k2d*10+k1d.$$

Baxdığımız misal üçün

$$M=7*1000+8*100+3*10+5=7835.$$

Məsələnin həlli üçün tərtib edilmiş proqram aşağıdakı kimi yazıla bilər:

Program Natural_Ters;

Uses WinCrt;

Var n,m: Integer;

Function Natural(k: Integer): Integer;

Var k1d,k1m,k2d,k2m,k3d,k3m: Integer;

Begin

k1d:=k div 1000;

k1m:=k mod 1000;

k2d:=k1m div 100;

k2m:=k1m mod 100;

k3d:=k2m div 10;

```

k3m:=k2m mod 10;
Natural:=k3m*1000+k3d*100+k2d*10+k1d;
End;
Begin
Write('4 rəqəmli N natural eded= ');
Readln(n);
m:=Natural(n);
Writeln('Rəqəmlərin tərs düzülüşü=',m:5)

```

End.

Programın icrasından alınan nəticə:

4 rəqəmli N natural eded= 5387

Rəqəmlərin tərs düzülüşü=7835

Qeyd etmək lazımdır ki, Readln prosedurunda heç bir dəyişən göstərilməzsə, onda programın icrası zamanı **dayanma** baş verir və programın icrası yalnız **Enter** klavişinin basılması ilə davam etdirilir. Readln prosedurunun bu xüsusiyyətindən programın icrası zamanı **fasilələr yaratmaq** məqsədilə istifadə etmək olar. Məsələn,

```

Write('x='); Readln(x);
Write('Enter klavishasini basin'); Readln;
Write('Programın icrası davam edir');

```

program fragmentinin icrası zamanı x dəyişənin qiyməti yaddaşa daxil edildikdən sonra, ekrana «Enter klavishasini basin» məlumatı çıxarılır və fasilə edilir. Programın icrası istifadəçinin Enter klavişini basdıqdan sonra davam etdirilir və ekrana «Programın icrası davam edir» məlumatı çıxır.

3.6. Laboratoriya işi № 4. Xətti proqramların tərtibi

1. 3 saylı laboratoriya işində verilmiş 1-20 saylı funksiyaların qiymətlərinin hesablanması üçün hər bir məsələnin həll alqoritmini qurun və programını yazın.

2. Funksiyalardan istifadə etməklə, təpəfi $a=3; 8; 12$ olan kubun səthinin sahəsini və həcmi hesablayın.

3. Prosedurdan istifadə etməklə, təpəfi $a=3; 8; 12$ olan kubun səthinin sahəsini və həcmi hesablayın.

4. Funksiyalardan istifadə etməklə, radiusu $r=2, 4, 6, 8$ olan kürənin səthinin sahəsini və həcmi hesablayın.

5. Prosedurdan istifadə etməklə, radiusu $r=2, 4, 6, 8$ olan kürənin səthinin sahəsini və həcmi hesablayın.

4. PASCAL DİLİNİN İDARƏEDİCİ KONSTRUKSIYALARI

4.1. Pascal dilinin operator qrupları

Əvvəlki fəsildə biz yalnız xətti strukturlu alqoritmlərə malik məsələlərin həlli üçün proqramların tərtibi ilə tanış olduq. Bu proqramlarda hərəsi bir dəfə olmaqla bütün operatorlar icra olunurdu. Onlarda hər hansı bir şərtin yoxlanılmasına və dövrlərin təşkilinə ehtiyac yox idi. Lakin bir çox hallarda biz budaqlanan və dövrü strukturlu alqoritmlərlə rastlaşırıq və bu fəsildə məhz belə həll alqoritmlərinə uyğun proqramların tərtib edilməsi ilə məşğul olacağıq.

Pascal dilinin operatorlarından hələlik yalnız biri ilə – mənsubetmə operatoru ilə tanış olmuşuq. Bu operatorun, eləcə də bütün digər operatorların yazılışlarının sonu «;» simvolu ilə göstərilir. Pascal dilinin operatorlarını şərti olaraq sadə və strukturlaşdırılmış kimi iki qrupa bölmək olar. Sadə operatorların yazılışında heç bir digər operatorndan istifadə olunmur. Sadə operatorlar qrupuna aşağıdakıları aid etmək olar:

- mənsubetmə operatoru (**:=**);
- şərtsiz keçid operatoru (**Goto**);
- prosuduraya müraciət operatoru.

Strukturlaşdırılmış operatorlar digər operatorlardan təşkil olunur. Bu qrupa aşağıda göstərilən operatorlar aid edilir:

- **Begin** və **End** operator mötərizələri içərisində yazılmış operatorlar ardıcılığından təşkil olunmuş tərkibli operator;
- **If**, **Case** şərt operatorları;
- **Repeat**, **While**, **For** dövr operatorları;
- **With** qoşma operatoru.

4.2. Şərtsiz konstruksiyalar

Pascal dilində şərtsiz konstruksiyalara bir operator və iki prosedur daxildir. **Goto** şərtsiz keçid operatoru proqramdakı operatorların icra olunma ardıcılığını dəyişmək üçün istifadə olunur. Bu operator vasitəsilə heç bir şərti yoxlamadan həm əvvəlki, həm də sonra gələn operatorlardan birinə keçid təşkil etmək olar. Operator ümumi şəkildə belə yazılır:

Goto <Nişan>;

Goto operatorlarında istifadə olunan bütün nişanlar proqramın **Label** – nişanların təsviri bölməsində göstərməlidir. Proqramlaşdırma

üzrə bir çox ədəbiyyatda Pascal-proqramlarda Goto operatorundan istifadə etmək tövsiyə edilmir. Buna səbəb operatorun Pascal-proqramın struktur görkəmini pozması, onun sonradan oxunması, təhlili və modifikasiyası prosesini çətinləşdirməsi hesab edilir. Buna baxmayaraq bir çox proqramçılar öz proqramlarında bu operatoru geniş istifadə edirlər. Məsələn, aşağıda verilmiş birinci proqram fraqmentinin icrasından alınan nəticə $a=5$, ikinci proqram fraqmentinin icrasından alınan nəticə isə $a=1$ olacaq.

1.

```

...
a:=1;
...
a:=5;
...
Writeln (a);
...

```

2.

```

...
Label m;
...
a:=1;
Goto m;
...
a:=5;
m: Writeln (a);
...

```

Proqramın icrası zamanı sonsuz dövr yaranmasın deyərək, diqqət yetirmək lazımdır ki, proqramda

Label 1;

1: a:=1; goto 1;

kimi və buna bənzər konstruksiya olmasın.

Bununla belə, bəzi hallarda proqramçının buraxdığı məntiqi səhv nəticəsində **sonsuz dövrlər** ilə rastlaşırıq və proqramın icrası sonsuz davam etdiyindən, kompüterlə heç bir iş aparmaq mümkün olmur. Proqramın icrasındakı sonsuz dövrü dayandırmaq üçün **Ctrl+Alt+Delete** birgə klavişlərini basmaq lazımdır. Bu zaman monitorun ekranına «Завершение работы» pəncərəsi açılır. Pəncərədəki işıqlı menyü sətirini proqramın adını göstərən sətirin üzərinə gətirib, «Завершение задачи» düyməsini basın. Ekranı proqramın adını göstərən yeni pəncərə açılacaq. Bu pəncərənin aşağı hissəsindəki «Завершение задачи» düyməsini basmaqla, proqramın icrası dayanacaqdır.

Exit prosedurunun vəzifəsi cari proqram blokunun işini vaxtından əvvəl başa çatdırmaqdan ibarətdir. Proqram bloku dedikdə proqram özü və ya onun tərkib hissəsi başa düşülür. Exit prosedurundan xüsusi halda proqramçının tərtib etdiyi prosedurlardan vaxtından əvvəl çıxışı təmin etmək üçün istifadə etmək olar. Məsələn, aşağıdakı birinci proqram fraqmentinin icrasından alınan nəticə $S=5^2=25$, ikinci proqram fraqmentinin icrasından alınan nəticə isə $S=5$ olacaqdır.

```

1. ...
  Var a,S: real;
  Procedure P(ap: real; Var sp: real);
  Begin ...
    sp:=ap;
    ...
    sp:=sqr(ap);
  End;
  Begin
    P(5,S); Writeln(S:3:0);
    ...
  End.

```

```

2. ...
  Var a,S: real;
  Procedure P(ap: real; Var sp: real);
  Begin ...
    sp:=ap; EXIT;
    ...
    sp:=sqr(ap);
  End;
  Begin
    P(5,S); Writeln(S:3:0);
    ...
  End.

```

Halt (n) – prosedurundan proqramın icrası zamanı *n* nəticə kodu ilə vaxtından əvvəl çıxışı təşkil etmək üçün istifadə olunur. Prosedurun yazılışında *n* ixtiyari tam ədəddir. Proqramın bir neçə yerində vaxtından əvvəl çıxışı təşkil etmək üçün müxtəlif nəticə kodlarını ilə **Halt** prosedurlarına müraciət etmək olar. Məsələn,

```

...
a:=-5;
Writeln('Menfi qiymet');
Halt (1);
...

```

proqram fraqmenti icra olunarkən, ekrana «Menfi qiymet» məlumatı çıxır və 1 nəticə kodu ilə proqramın icrası başa çatır.

4.3. Şərti konstruksiyalar

Şərti konstruksiyalara **If ... then ... else ...** şərt və **Case ... of** variant operatorları daxildir. If operatorunun iki yazılış forması vardır:

1) **If <məntiqi ifadə> then Begin**

<operatorlar>;

End;

Operator icra olunarkən məntiqi ifadənin qiyməti hesablanır. Bu qiymət **True** (doğru) isə **Begin** və **End** operator mötərizələri içərisində yazılmış operatorlar icra olunur, sonra isə **If** operatorundan sonrakı operatora keçid verilir. Məntiqi ifadənin qiyməti **False** (yalan) isə, **If** operatorundan sonrakı operatora keçid baş verir. Əgər məntiqi ifadənin doğru qiymətində yalnız bir operatorun icrası tələb olunursa, onda operatorun yazılışında **Begin** və **End** operator mötərizələri qoyulmur:

If <məntiqi ifadə> then <operator>;

Məsələn,

If a>0 then Begin

x:=sqr(a); y:=sin(x)

End;

If a>0 then y:=sin(sqr(a));

Fraqmentin icrası zamanı a>0 şərti ödənilərsə, onda then budağında göstərilən operator və ya operatorlar qrupu yerinə yetiriləcək.

2) If <məntiqi ifadə> then Begin

<operatorlar>;

End

else Begin

<operatorlar>;

End;

Operator icra olunarkən məntiqi ifadənin qiyməti hesablanır. Bu ifadənin qiyməti True (doğru) isə then budağındakı Begin və End operator mötərizələri içərisində yazılmış operatorlar, False (yalan) isə else budağındakı Begin və End operator mötərizələri içərisində yazılmış operatorlar icra olunur və hər iki halda if operatorundan sonrakı operatora keçid baş verir. Then və else budaqlarında yalnız bir operatorun icrası tələb olunduğu hallarda Begin və End operator mötərizələri qoyulmur. Məsələn,

If a>0 then Begin

x:=sqr(a); y:=sin(x)

End

else

Begin

x:=sin(a); y:=sqr(x)

End;

If a>0 then y:=sin(sqr(a))

else y:=sqr(sin(a));

Fraqmentin icrası zamanı a>0 şərti ödənilərsə, onda then, əks halda isə else budağında göstərilən operator və ya operatorlar qrupu yerinə yetiriləcək.

Qeyd etmək lazımdır ki, xüsusi halda If operatorunun daxilində bir və ya bir neçə If ... then ... else konstruksiyasından istifadə oluna bilər. Bu zaman nəzərə almaq lazımdır ki, iç-içə yazılmış If operatorlarında yazılmış hər bir else budağı özündən əvvəl gələn If operatoruna aid edilir. Məsələn,

If ... then If ... then ... else ...

konstruksiyasında else budağı ikinci If operatoruna aiddir. Else budağının birinci If operatoruna aid edilməsi tələb olunarsa, onda ikinci If ... then konstruksiyası Begin və End operator mötərizələri içərisində yazılmalıdır:

If ... then Begin If ... then ...; End

else ...;

Deyilənləri konkret bir misalda nəzərdən keçirək:

```

y:=0;
If x>a then
    If x>b then y:=1
    else y:=2;

```

Nəticə:

x=3, a=4, b=5 olduqda, y=0;

x=4, a=3, b=5 olduqda, y=2;

x=5, a=3, b=4 olduqda, y=1.

Həlli If şərt operatorundan istifadəyə əsaslanan bir neçə misalı nəzərdən keçirək.

Misal 4.1. $y = \sin \frac{x}{a}$ funksiyasının qiymətini hesablamalı. x və a dəyişənləri ixtiyari qiymətlər alır.

Həlli. Alqoritmə $a \neq 0$ şərti yoxlanılmalıdır və yalnız bu halda hesablama mümkündür. Məsələnin həll alqoritminin blok-sxemi şəkil 4.1-də, proqramın listinqi isə aşağıda göstərilmişdir.

```

Program P4_1;
Uses WinCrt;
Var x,a,y: real;
Begin

```

```

    Write('x='); Readln(x);
    Write('a='); Readln(a);
    if a<>0 then begin
        y:=sin(x/a);
        Writeln(y:6:4)
    end

```

End.

Misal 4.2. $y = \sin \frac{\sqrt{x}}{a}$ funksiyasının qiymətini hesablamalı. x və a dəyişənləri ixtiyari qiymətlər alır.

Həlli. Alqoritmə birinciliyi heç bir mənə daşımayan $a \neq 0$ və $x \geq 0$ şərtləri yoxlanılmalıdır. Bu səbəbdən onları **And** əməliyyatından istifadə etməklə bir məntiqi ifadə

```

y:=0;
If x>a then Begin
    If x>b then y:=1
    End
else y:=2;

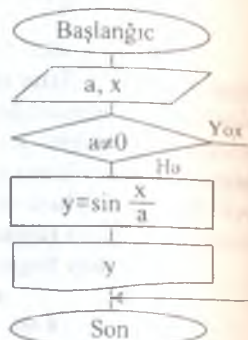
```

Nəticə:

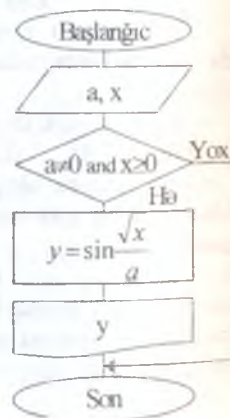
x=3, a=4, b=5 olduqda, y=0;

x=4, a=3, b=5 olduqda, y=2;

x=5, a=3, b=4 olduqda, y=1.



Şəkil 4.1



Şəkil 4.2

şəklində yazmaq olar. Məsələnin həll algoritminin blok-sxemi şəkil 4.2-də, proqramın listingi isə aşağıda göstərilmişdir.

```

Program P4_2;
Uses WinCrt;
Var a, x, y: real;
Begin
  Write('x='); Readln(x);
  Write('a='); Readln(a);
  if (a <> 0) and (x >= 0) then
    begin
      y:=sin(sqrt(x)/a); Writeln(y:6:4)
    end;
End.

```

Misal 4.3. $y = \sqrt{\sin \frac{x}{a}}$ funksiyasının

qiymətini hesablamaqlı. x və a dəyişənləri ixtiyari qiymətlər alır.

Həlli. Alqoritmdə ilk növbədə $a \neq 0$, sonra isə $\sin \frac{x}{a} \geq 0$ şərti yoxlanılmalıdır. Məsələnin

həll algoritminin blok-sxemi şəkil 4.3-də, proqramın listingi isə aşağıda göstərilmişdir.

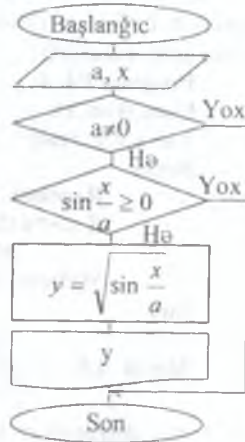
```

Program P4_3;
Uses WinCrt;
Var x,a,y: real;
Begin
  Readln(a, x);
  if a <> 0 then
    if sin(x/a) >= 0 then
      begin
        y:=sqrt(sin(x/a));
        Writeln(y:6:4)
      end;
End.

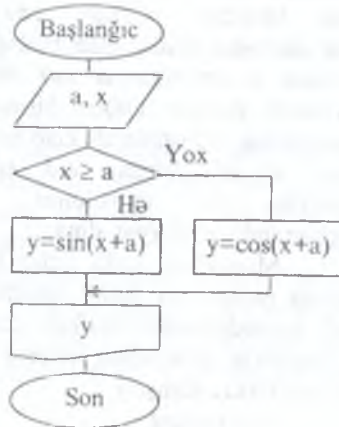
```

Misal 4.4. $y = \begin{cases} \sin(x+a) & x \geq a \\ \cos(x+a) & x < a \end{cases}$ funksiyasının qiymətini hesablamaqlı. x və a dəyişənləri ixtiyari qiymətlər alır.

Həlli. Proqram icra olunarkən a



Şəkil 4.3



Şəkil 4.4

və x dəyişənlərinin qiymətləri yaddaşa daxil edilir və $x \geq a$ şərti yoxlanılır. Əgər şərt ödənilərsə y dəyişəninin qiyməti $y = \sin(x+a)$ düsturu, əks halda isə $y = \cos(x+a)$ düsturu ilə hesablanır. Hər iki halda nəticə çap edilir. Məsələnin həll algoritminin blok-sxemi şəkil 4.4-də, proqramın listingi isə aşağıda göstərilmişdir.

```

Program P4_4;
Uses WinCrt;
Var x,a,y: real;
Begin
    Readln(a, x);
    if x >= a then y := sin(x+a)
    else y := cos(x+a);
    Writeln('y=', y:5:3)
End.

```

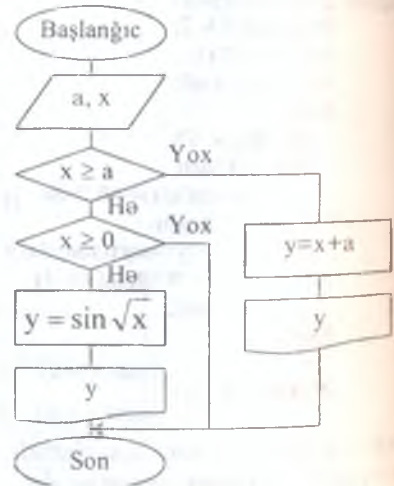
Misal 4.5. $y = \begin{cases} \sin \sqrt{x} & x \geq a \\ x + a & x < a \end{cases}$ funksiyasının qiymətini hesabla-

malı. x və a dəyişənləri ixtiyari qiymətlər alır.

Həlli. Məsələnin həll algoritminin blok-sxemi şəkil 4.5-də, proqramı isə 2 variantda tərtib edilmişdir. Birinci variantda $x \geq a$ məntiqi ifadəsinin doğru qiyməti üçün iç-içə yerləşdirilmiş If operatorundan istifadə olunmuşdur və içdəki if operatoru bloka alınmışdır. Belə hal üçün blokdan istifadə edilməsi zəruridir, əks halda, else budağının sonuncu if operatoruna aid olması qaydasını nəzərə alsaq, birinci if operatoruna aid olunmalı else budağı ikinci if operatoruna aid edilsə, beləliklə də məsələnin həll algoritmində ziddiyyət alınar.

Məsələnin həlli üçün tərtib edilmiş proqramın ikinci variantında Halt prosedurundan istifadə edilmiş və beləliklə, proqramın listingi daha yığcam forma almışdır.

1-ci variant:
Program P4_5a;
Uses WinCrt;
Var a, x, y: real;



Şəkil 4.5

```

Begin
  Readln(a,x);
  if x>=a then begin
    if x>=0 then begin
      y:=sin(sqrt(x));
      Writeln ('y=',y:5:3)
    end
  end
  else begin
    y:=x+a;
    Writeln ('y=',y:5:3)
  end;
End.

```

2-ci variant:
 Program P4_5b;
 Uses WinCrt;
 Var a, x, y: real;
 Begin
 Readln(a,x);
 if x>=a then if x>=0 then y:=sin(sqrt(x))
 else halt
 else y:=x+a;
 Writeln ('y=',y: 5: 3)
 End.

Misal 4.6. $y = \begin{cases} \sqrt{x+a} & x > \sin a \\ \frac{x+a}{x-a} & x \leq \sin a \end{cases}$ funksiyasının qiymətini hesab-

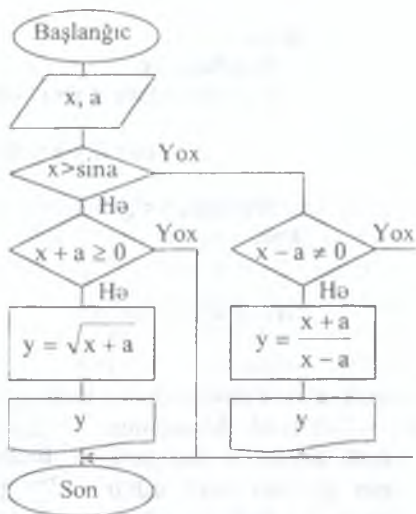
lamalı. x və a dəyişənləri ixtiyari qiymətlər alır.

Həlli. Məsələnin həlli zamanı x və a dəyişənlərinin qiymətləri daxil edilməli və x dəyişəninin qiyməti ilə $\sin a$ ifadəsinin qiyməti müqayisə olunmalıdır. Əgər $x > \sin a$ şərti ödənilərsə, onda funksiyanın qiyməti onun ifadəsindəki birinci budaqla, əks halda isə ikinci budaqla hesablanmalıdır. Lakin hər iki budaq üzrə hesablamaların aparılmasından əvvəl müəyyən şərtlər, birinci budaqda $x+a \geq 0$, ikinci budaqda isə $x-a \neq 0$ şərtləri yoxlanılmalıdır. Hesablamaların sonunda y dəyişəninin qiyməti çap edilir. Məsələnin həll alqoritmının blok-sxemi şəkil 4.6-da, proqramın listingi isə aşağıda göstərilmişdir.

```

Program P4_6;
Uses WinCrt;
Var a, x, y: real;

```



Şəkil 4.6

Begin

Readln(a, x);

if $x > \sin(a)$ then if $x+a > 0$ then $y := \text{sqr}(x+a)$

else halt

else if $x-a < 0$ then $y := (x+a)/(x-a)$

else halt;

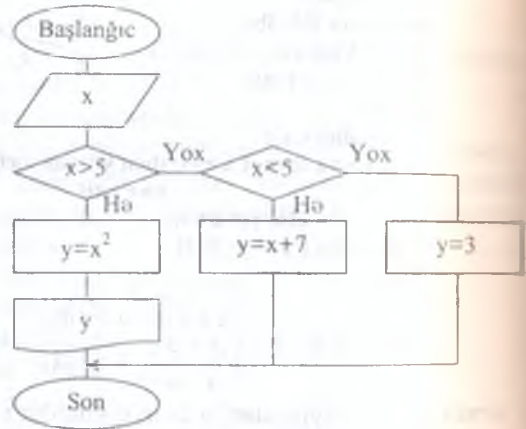
Writeln('y=', y:5:3)

End.

Misal 4.7. $y = \begin{cases} x^2 & x > 5 \\ x+7 & x < 5 \\ 3 & x = 5 \end{cases}$ funksiyanın qiymətini hesabla.

malı. x və a dəyişənləri ixtiyari qiymətlər alır.

Həlli. Məsələnin həlli zamanı x dəyişənin qiyməti daxil edilir və o, 5 ədədi ilə müqayisə olunur. Əgər $x > 5$ isə $y = x^2$, $x < 5$ isə $y = x+7$, $x = 5$ olduqla isə $y = 3$ düsturlarından biri ilə hesablama aparılır. Sonda nəticə çap edilir. Məsələnin həll algoritminin blok-sxemi şəkil 4.7-də, proqramın listinqi isə aşağıda göstərilmişdir.



Şəkil 4.7

Program P4_7;

Uses WinCrt;

Var x, y: real;

Begin

Readln(x); if $x > 5$ then $y := \text{sqr}(x)$ else if $x < 5$ then $y := x+7$ else $y := 3$;

Writeln('y=', y:5:2)

End.

Bir çox hallarda müəyyən şərt daxilində iki yox, bir neçə budak üzrə əməliyyatları yerinə yetirmək lazım gəlir. Belə hallarda If operatorundan istifadə etmək o qədər də səmərəli olmur və bu zaman Pascal dilinin variant operatoruna (Case ... of) üstünlük verilir. Operatorun yazılma forması aşağıdakı kimidir:

Case <Selektor> of

N₁: S₁;

N₂: S₂;

...

Else S;

End;

Operatorun yazılışında selektor ifadədir və o, variantlarda göstərilən qiymətləri (N₁, N₂, ...) ala bilər. N₁, N₂, ... nişan rolunu oynayan konstantalardır, onların tipi selektorun tipi ilə eyni olmalıdır. Qeyd etmək lazımdır ki, bu tip **Real** ola bilməz. Operator icra olunarkən, ifadənin qiyməti hesablanır və bu qiymət N₁, N₂, ... nişanları ilə müqayisə edilir. İfadənin qiyməti hansı nişana (N_k) uyğun gəlsə, həmin nişandakı operator (S_k) icra olunur. Əgər ifadənin qiyməti heç bir nişana uyğun gəlmirsə, onda **Else** budağında göstərilmiş operator (S) yerinə yetirilir. Operatorda **Else** budağının göstərilməsi vacib deyil. Əgər **Else** xidməti sözündən istifadə edilməyibsə və ifadənin aldığı qiymət heç bir nişana uyğun gəlmirsə, onda **Case** operatorundan sonrakı operatora keçid baş verir. Variantların sonu **End** ilə göstərilir. Əgər hər hansı nişandan sonra bir yox, bir neçə operator icra olunmalıdırsa, onda bu operatorlar **Begin** və **End** operator mötərizələri içərisində yazılırlar. Hər hansı budaq bir neçə nişana uyğun gəlsə, onda nişanlar siyahı şəklində göstərilə bilər. **Case** operatorundan istifadəyə bir misal göstərək:

Misal 4.8. Həftənin gününə görə dərs cədvilini ekrana çıxarmalı.

Həlli. Məsələnin həlli üçün programın listinginə baxaq.

Program P4_8;

Uses WinCrt;

Label L;

Const s1='İnformatika'; s2='Riyaziyyat'; s3='İngilis dili';

s4='Fizika'; s5='Kimya'; s6='Tarix';

Var N: Integer;

Begin

L: Write ('Heftenin gunu: '); Readln(N);

Case N of

1,4: Writeln (s1,', ',s2,', ',s4,', ');

2: Writeln (s3,', ',s5,', ',s1,', ');

3: Writeln (s1,', ',s6,', ',s3,', ');

5: Writeln (s5,', ',s2,', ',s4,', ');

6: Writeln ('Shenbe gunu ders yoxdur');

7: Writeln ('Bazar gunu istirahet gunudur');

Else Begin Writeln ('Gun duzgun daxil edilmeyib'); Goto L; End;

End;

End.

4.4. Laboratoriya işi № 5. Şerti və şərtsiz konstruksiyalar

Aşağıdakı riyazi funksiyaların hesablanması alqoritmlərinin blok-sxem təsvirini verin və programını tərtib edin:

1.	$y = \frac{x^2 + \cos x}{\sin^2 x}$	13.	$y = \sqrt{\frac{x - a^2 \sin x }{2 - x^2}}$
2.	$y = \frac{x - \sin(x^2 - a)}{x + a}$	14.	$y = \sqrt{\frac{x + a}{x - a}} - \ln x^2 - a $
3.	$y = \frac{\sqrt{x + a}}{\cos x}$	15.	$y = \frac{\sqrt{x \cos a} + \sqrt{a \sin x}}{x}$
4.	$y = \frac{x^3 - \ln x + a }{\sin^2 a}$	16.	$y = \frac{x^2 - a \cos \sqrt{x - a}}{\ln a }$
5.	$y = \sqrt{\operatorname{tg} x - \ln \cos(x - a) }$	17.	$y = \frac{a + x}{2} \sin \sqrt{x + a}$
6.	$y = \sqrt{\frac{x^2 \sin x - 1}{2 - a}}$	18.	$y = \frac{x - a \sqrt{x \ln x}}{x^2 \sin x}$
7.	$y = \frac{1 - x \operatorname{tg} x^2}{1,2 + \cos x}$	19.	$y = \frac{1 - \operatorname{tg} x \cos x^2}{0,1 - \sin x}$
8.	$y = \sqrt{\ln(\ln x + 1)}$	20.	$y = \sqrt{\cos(\ln x - 1)}$
9.	$y = \sqrt[4]{x \cos x + a \ln x}$	21.	$y = \sqrt[3]{ax^3 + 2a \sin x - 0,3}$
10.	$y = \frac{1 - \ln x^3}{1 - \sin \ln a}$	22.	$y = \frac{1 + \ln x}{1 - \ln^2 a} \sqrt[4]{\cos(x + a)}$
11.	$y = \begin{cases} \sin \frac{x + a}{2} & x^2 < a \\ \cos \sqrt{x - a} & x^2 \geq a \end{cases}$	23.	$y = \begin{cases} \operatorname{arctg} \frac{x + a}{\cos x} & \operatorname{tg} x < a \\ \sqrt{a \ln x } & \operatorname{tg} x \geq a \end{cases}$
12.	$y = \begin{cases} \operatorname{arctg} \frac{x^3}{\sqrt{x - a}} & x < \sqrt{a} \\ \ln(ax^2 - 2a) & x \geq \sqrt{a} \end{cases}$	24.	$y = \begin{cases} (a + \ln x)^3 & \sqrt{\sin x} \leq a \\ \frac{\sin \sqrt{ax}}{\cos(a + x)} & \sqrt{\sin x} > a \end{cases}$

25.	$y = \begin{cases} x^2 + a \sin x & x > a \\ \sin \sqrt{x} + \sqrt{\cos a} & x < a \\ \sqrt{a \sin a + \cos a^2} & x = a \end{cases}$	28.	$y = \begin{cases} 1,5 \ln^2 \frac{x-1}{x} & x < \cos x \\ \frac{x + 5 \sin a}{\cos x} & x \geq \cos x \end{cases}$
26.	$y = \begin{cases} \left \frac{x-a}{x+a} \right \cos x & x \leq \sin x \\ \ln \left(\frac{x-a}{a} \right)^2 & x > \sin x \end{cases}$	29.	$y = \begin{cases} x^2 - \sqrt{a} & x > a \\ \sqrt{x + \sin^3 a} & x < a \\ \frac{1}{a+1} & x = a \end{cases}$
27.	$y = \begin{cases} \frac{\sin(\sqrt{x+a})}{0,5 - \cos \sqrt{a}} & x < a \cos x \\ \ln \cos(x+a) & x > a \cos x \\ \sqrt[3]{x^2 + a} & x = a \cos x \end{cases}$	30.	$y = \begin{cases} ax + \ln(a+x) & x^3 > a \\ \sin \sqrt{a+x} & x^3 < a \\ \frac{\sin a}{\cos \sqrt{x}} & x^3 = a \end{cases}$

4.5. Dövri konstruksiyalar

Dövrələrin təşkili üçün Pascal dilində 3 konstruksiya təklif edilir. Dövr yaradılarkən, dövrələrin sayı əvvəlcədən məlumdursa, onda **For ... To ... Do** konstruksiyasından, məlum olmadıqda isə, **While ... Do** və **Repeat ... Until** konstruksiyalarından istifadə edilir. Lakin bu o demək deyildir ki, dövrələrin təşkili yalnız dövr operatorları vasitəsilə həyata keçirilir, başqa sözlə, dövrələri şərti və ya şərtsiz keçid operatorları ilə də yaratmaq olar. Məsələn, 1-dən 10-dək natural ədədlərin kvadratları cəmini hesablamaq üçün aşağıdakı proqram fraqmentlərindən istifadə etmək olar:

Label 10;

...

S:=0; k:=1;

10: S:=S+sqr(k);

k:=k+1;

if k<=10 then goto 10

...

Label 10,15;

...

S:=0; k:=1;

10: If k>10 then goto 15;

S:=S+sqr(k); k:=k+1;

Goto 10;

15: ...

Buna baxmayaraq, dövri konstruksiyalardan istifadə daha əlverişli və səmərəlidir. Bu konstruksiyaları ayrılıqda nəzərdən keçirək.

Dövrələrin təşkili üçün **For ... To ... Do** konstruksiyası ümumi şəkildə belə yazılır:

For P:=Q₁ To (Downto) Q₂ Do Begin S₁; S₂; ... End;

Burada **P** tam tipli dövr dəyişənidir və dövrə sayğac rolunda istifadə olunur. Ona əvvəlcə **Q₁** başlanğıc qiyməti mənsub edilir və hər bir dövrə **To** xidməti sözündən istifadə olunarsa, ona 1 əlavə edilir, **Downto** xidməti sözündən istifadə olunduqda isə, ondan 1 çıxılır. Dövr dəyişəni üzərindəki hər bir əməliyyatdan sonra, onun qiyməti dövr dəyişəninin son **Q₂** qiyməti ilə müqayisə olunur. Dövr **P** dövr dəyişəninin qiyməti **Q₂** qiymətini aşdıqda dövrədən çıxış baş verir. **S₁, S₂, ...** dövrün gövdəsini təşkil edən operatorlardır və onlar dövrün daxilində hər dəfə icra olunurlar. Dövrün gövdəsi yalnız bir operatorndan ibarət olarsa, onda **Begin** və **End** operator mötərizələrindən istifadə olunmur.

For ... To ... Do konstruksiyası vasitəsilə 1-dən 10-dək natural ədədlərin kvadrları cəminin hesablanması proqram fraqmentini aşağıdakı kimi ifadə etmək olar:

Var k,S: Integer;

...

S:=0;

For k:=1 to 10 do S:=S+sqr(k);

...

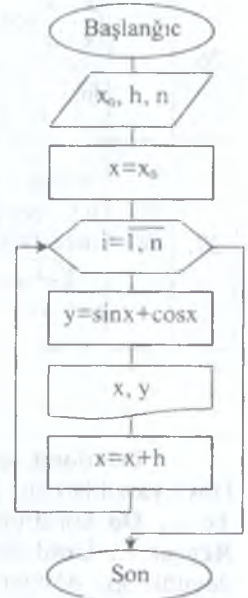
For ... To ... Do konstruksiyasından istifadəyə əsaslanmış bir neçə misalı nəzərdən keçirək.

Misal 4.9.

$y = \sin x + \cos x$ funksiyası verilmişdir. x arqumentinin $x_0=0$ başlanğıc qiymətindən $h=0,1$ addımı ilə aldığı qiymətlərində y funksiyasının $n=10$ sayda qiymətlərini hesablamaq.

Həlli. Məsələnin həll alqoritminin blok-sxemi şəkil 4.8-də, proqramın icrasından alınan nəticələr şəkil 4.9-da, proqramın listingi isə aşağıda göstərilmişdir.

Program P4_9;



Şəkil 4.8

x	y
0.0	1.0000
0.1	1.0950
0.2	1.1790
0.3	1.2510
0.4	1.3100
0.5	1.3570
0.6	1.3900
0.7	1.4090
0.8	1.4140
0.9	1.4050

Şəkil 4.9

```

Uses WinCrt;
Var i, n: Integer; x, y, x0, h: Real;
Begin

```

```

    Readln(x0, h, n);
    x:=x0;
    For i:=1 to n do begin
        y:=sin(x)+cos(x);
        Writeln(x:5:1, y:8:3);
        x:=x+h
    end;

```

End.

Misal 4.10. x_k argumentinin ixtiyari götürülmüş qiymətlərində $y_k = x_k \sin x_k$ funksiyasının qiymətlərini hesablamalı. k indeksi 1-dən 5-dək qiymət alır.

Həlli. Məsələnin həll alqoritminin blok-sxemi şəkil 4.10-da, programın icrasından alınan nəticələr şəkil 4.11-də, programın listingi isə aşağıda göstərilmişdir.

Program P4_10;

Uses WinCrt;

Var k, n: Integer;

x, y: Real;

Begin

Write('n='); **Readln(n);**

For k:=1 to n do begin

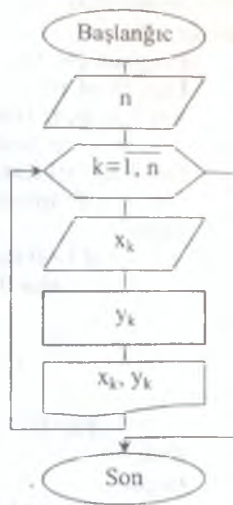
Write('k=',k:2,' x=');

Read(x); y:=x*sin(x);

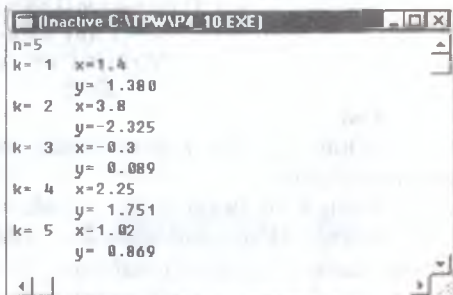
Writeln(' y=',y:6:3);

end;

End.



Şəkil 4.10



Şəkil 4.11

Misal 4.11. Tam tipli n , m və k -nın verilmiş 4 sayda qiymətlərində $x = \frac{n!(n-k)!(k+1)!}{(n-k+m)!m!}$ düsturu ilə təyin edilmiş x dəyişənin qiymətini hesablamalı. Faktorialın hesablanmasını funksiya ilə yerinə yetirməli.

Həlli: Faktorial aşağıdakı düsturla hesablanır:

$$L! = \begin{cases} 0 & L < 0 \\ 1 & L = 0 \\ 1 \cdot 2 \cdot 3 \cdot \dots \cdot L & L > 0 \end{cases}$$

Məsələnin həlli üçün tərtib edilmiş proqramın listingi aşağıda göstərilmişdir.

```

Program P4_11;
Uses WinCrt;
Var i, n, m, k: Integer;
    x: Real;
Function Ft(L: Integer): Integer;
Var i, P: Integer;
Begin
    if L<0 then P:=0
    else if L=0 then p:=1
    else Begin
        P:=1;
        For i:=1 to L do P:=P*i;
    End;
    Ft:=P;
End;
Begin
    For i:=1 to 4 do Begin
        Write ('n, m, k='); Readln(n, m, k);
        If Ft(n-k+m)*Ft(m)>0 then Begin
            x:=(Ft(n)*Ft(n-k)*Ft(k+1))/(Ft(n-k+m)*Ft(m));
            Writeln ('x=', x:6:3)    End;
        End;
    End.

```

While ... Do konstruksiyası ümumi halda aşağıda göstərilən formaya malikdir:

```

While R Do Begin S1; S2; ... End;

```

Burada **R** məntiqi ifadədir. Operator icra olunarkən dövrü olaraq məntiqi ifadənin qiyməti hesablanır. Nə qədər ki, bu ifadə **True** (doğru) qiymət alır, **Begin** və **End** operator mötərizələri içərisindəki **S₁, S₂, ...** operatorları icra olunacaqdır. Məntiqi ifadənin qiyməti **False** (yalan) olduğu halda, dövrədən çıxış baş verir və növbəti operatora keçid verilir. Dövrün gövdəsi yalnız bir operatorndan ibarət olarsa, onda **Begin** və **End** operator mötərizələrindən istifadə olunmur. Operatorun yazılışından görüldüyü kimi, məntiqi ifadə dövrün əvvəlində yazılmışdır və bu səbəbdən də **While ... Do** operatoru ön şərtli dövr operatoru adlandırılır. Qeyd etmək lazımdır ki, məntiqi ifadə elə birinci dövrə **False** qiymət ala bilər və bu zaman dövrün gövdəsinə daxil olan operatorlar bir dəfə də olsun icra olunmayacaqlar.

While ... Do konstruksiyasından istifadə etməklə 1-dən 10-dək natural ədədlərin kvadrları cəminin hesablaması proqram fraqmentini nəzərdən keçirək:

Var k, S: Integer;

...

S:=0; k:=1;

While k<=10 Do Begin

S:=S+sqr(k);

k:=k+1

End;

...

Aşağıda **While ... Do** konstruksiyasından istifadəyə əsaslanmış bir neçə misal göstərilmişdir.

Misal 4.12. Arqumentin h addımı ilə $[x_0, x_c]$ intervalına daxil olan qiymətlərində $y = x^2 \sin x$ funksiyasının qiymətlərini hesablayın.

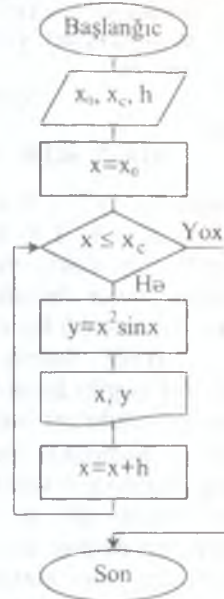
Həlli. Proqram icra olunarkən, x dəyişəninə x_0 başlanğıc, x_c son qiymətləri və h addımı daxil edilir və x dövr dəyişəninə x_0 başlanğıc qiyməti mənsub edilir. Sonra dövrün təşkili üçün məntiqi ifadənin qiyməti hesablanır. Məntiqi ifadə nə qədər ki, **True** qiymət alır, dövrün gövdəsindəki funksiyanın qiymətinin hesablaması, bu qiymətin çap edilməsi və arqumentə addım verilməsi əməliyyatları dövrü olaraq yerinə yetirilir. Dövrdən çıxışla proqramın icrası başa çatdırılır. Məsələnin həll algoritminin blok-sxemi şəkil 4.12-də, proqramın icrasından alınan nəticələr şəkil 4.13-də, proqramın listingi isə aşağıda göstərilmişdir.

Program P4_12;

Uses WinCrt;

Var x, y, x0, xc, h: Real;

Begin



Şəkil 4.12

İlkin verilənlər: 1 4 0.5	
1.0	0.84
1.5	2.24
2.0	3.64
2.5	3.74
3.0	1.27
3.5	-4.30
4.0	-12.11

Şəkil 4.13

Var k, S: Integer;

S:=0; k:=1;

Repeat

S:=S+sqr(k);

k:=k+1

Until k>10;

Repeat ... Until konstruksiyasından istifadəyə əsaslanmış bir neçə misal nəzərdən keçirək.

Misal 4.14. π ədədinin təqribi qiyməti aşağıdakı sıra ilə hesablanıla bilər:

$$\pi \cong S = 4 \cdot \left(1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \dots + (-1)^{n+1} \frac{1}{2n-1} + \dots \right).$$

Sıranın qiymətini hesablayın və π ədədi ilə müqayisə edin.

Həlli. Məsələnin həlli alqoritmində sıraya daxil olan hədlərin işarələrinin təyin olunması maraq kəsb edir. Bu məqsədlə $n \bmod 2$ ifadəsinin qiyməti hesablanır. n -in tək qiymətlərində ifadənin qiyməti 1-ə bərabərdir və sıra həddinin işarəsi müsbət götürülür. n -in cüt qiymətlərində isə ifadənin qiyməti 0 olur və bu halda sıra həddi -1 ədədinə vurulmaqla, ona mənfi işarə verilir. Sıranın hədləri müxtəlif işarəli olduğundan dövrün sonunda sıra həddinin özü yox, onun modulu ε ədədi ilə müqayisə olunur. Alqoritmın sonunda sıranın qiyməti və iterasiyaların sayı çap edilir. Məsələnin həll alqoritmının blok-sxemi şəkil 4.15-də, proqramın listingi isə aşağıda göstərilmişdir. Proqramın icrası zamanı 100001-ci iterasiyada sıra 3,14169 təqribi qiymətini almışdır.

Program P4_14;

Uses WinCrt;

Const eps=0.00005;

Var S, S1: Real; n: Integer;

Begin

n:=1; S:=0;

Repeat

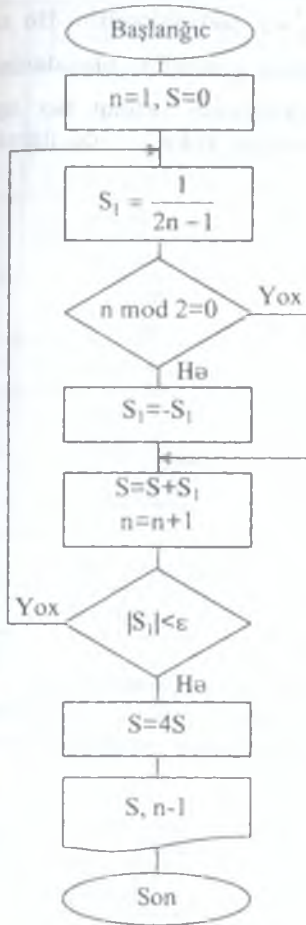
S1:=1/(2*n-1);

if n mod 2=0 then S1:=-S1; S:=S+S1; n:=n+1;

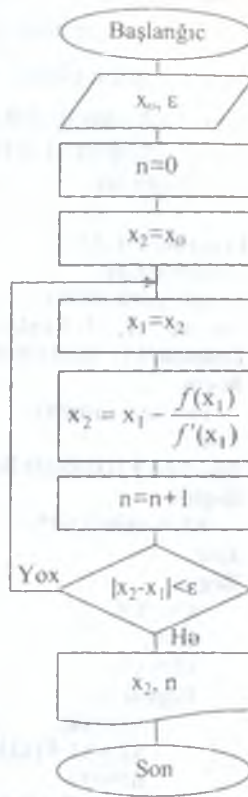
Until abs(S1)<eps;

Writeln ('Cavab=', 4*S:7:5); Writeln ('Iterasiya=', n-1)

End.



Şekil 4.15



Şekil 4.16

Misal 4.15. Məlumdur ki, $e^{-x} - \ln x = 0$ tənliyinin $(1,1; 1,9)$ intervalında bir kökü vardır. Nyuton üsulu ilə bu kökü tapın.

Həlli. Tənliyin kökünün tapılması üçün Nyuton üsulu aşağıdakı düsturdan istifadəyə əsaslanır:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

Bu düsturla tənliyin kökü üçün yaxınlaşmalar yerinə yetirilir. İterasiya prosesi o vaxt başa çatdırılır ki, $|x_{n+1} - x_n| < \varepsilon$ şərti ödənilsin. Bu zaman tənliyin kökü təqribi olaraq x_{n+1} -ə bərabər götürülür. Məsələnin həll alqoritmının blok-sxemi şəkil 4.16-da, proqramın listingi isə aşağıda göstərilmişdir. Proqramın icrası zamanı tənliyin kökü 4-cü iterasiyada 1,30980 qiymətini alır.

```

Program P4_13;
Uses WinCrt;
Const eps=0.00001;
Var x0, x1, x2: Real; n: Integer;
Function F(x:Real):Real;
Begin
    F:=exp(-x)-ln(x)
End;
Function F1(x:Real):Real;
Begin
    F1:=-exp(-x)-1/x
End;
Begin
    x0:=1.1;
    n:=0;
    x2:=x0;
    Repeat
        x1:=x2;
        x2:=x1-F(x1)/F1(x1);
        n:=n+1
    Until abs(x2-x1)<eps;
    Writeln ('Cavab=', x2:7:5);
    Writeln ('İterasiya=', n)
End.

```

4.6. Laboratoriya işi № 6. Dövri konstruksiyalar

1. Aşağıda göstərilmiş funksiyanın x arqumentinin x_0 başlanğıc qiymətlərində h addımı ilə n sayda qiymətlərini hesablayın. a, b – sabit kəmiyyətlərdir. Funksiyanın qiymətini hesablamaq mümkün olmadıqda, ekrana «Mümkün deyil» məlumatını çıxarın.

No	Funksiya	x_0	h	n	a	b
1.1.	$y = \sin^2 \frac{a+x}{2b} - \cos \sqrt{x}$	3,2	0,3	12	1,5	2,1
1.2.	$y = \sqrt{\cos \frac{a}{x}}$	0,7	0,15	10	1,25	-
1.3.	$y = \sin \sqrt{ax^2 + bx - 2ab}$	0,6	0,1	10	1,1	2
1.4.	$y = \frac{\ln \cos(x+a)}{a^2 \sin \sqrt{x}}$	1,9	0,5	8	0,7	-
1.5.	$y = \lg \frac{1-ax}{1+ax}$	1	0,5	10	1,9	-
1.6.	$y = a\sqrt{x+b} + \ln x-a $	1	1	15	-3	-
1.7.	$y = \frac{a + \sin x}{\cos x - b}$	0,7	0,3	12	3	-0,2
1.8.	$y = \sqrt{\cos \frac{a+x}{a-x}}$	0,1	0,5	18	1,1	-
1.9.	$y = \sqrt{0,5 - \cos^2 a \sin x}$	1	1,1	10	2	-
1.10.	$y = \ln \cos \sqrt{x+ab}$	0,5	0,5	20	1,1	1,3

2. x_k arqumentinin verilmiş qiymətlərində y_k funksiyanın qiymətini hesablayın. a , b – sabit kəmiyyətlərdir. Funksiyanın qiymətini hesablamaq mümkün olmadıqda, ekrana «Mümkün deyil» məlumatını çıxarın.

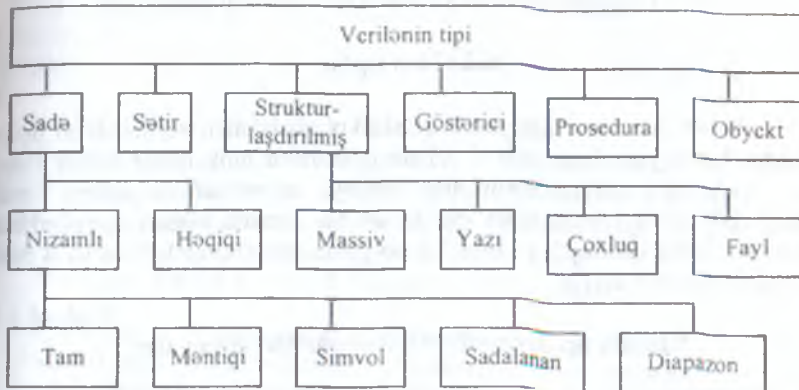
No	Funksiya	x_k	a	b	k
2.1.	$y_k = \sqrt{\cos x_k}$	{3; 5; 7; 10}	-	-	4
2.2.	$y_k = \sin^2 \frac{a+x_k}{b}$	{-3; 2; 4; 7; 5}	1,2	-2	5
2.3.	$y_k = \ln \cos x_k + \sin x_k $	{-3; -5; 0; 2; 9; 1; 3}	-	-	7
2.4.	$y_k = \lg \sqrt{\sin x_k}$	{-1; 0; 2,5; 7; 1; -2}	-	-	6
2.5.	$y_k = x_k \ln \sin x_k$	{2; 1; 0; -1; -4; 3; 7; -2}	-	-	8

6.2.	$S = \left(\frac{1}{2} + \frac{1}{3}\right) + \left(\frac{1}{2^2} + \frac{1}{3^2}\right) + \dots + \left(\frac{1}{2^n} + \frac{1}{3^n}\right) + \dots$	
6.3.	$S = \frac{1}{2} + \frac{3}{2^2} + \frac{5}{2^3} + \dots + \frac{2n-1}{2^n} + \dots$	
6.4.	$S = \frac{1}{1 \cdot 2} + \frac{1}{2 \cdot 3} + \frac{1}{3 \cdot 4} + \dots + \frac{1}{n(n+1)} + \dots$	
6.5.	$S = \frac{1}{1 \cdot 4} + \frac{1}{4 \cdot 7} + \dots + \frac{1}{(3n-2)(3n+1)} + \dots$	
6.6.	$S = \frac{1!}{1} + \frac{2!}{2^2} + \frac{3!}{3^3} + \dots + \frac{n!}{n^n} + \dots$	
6.7.	$S = \frac{2 \cdot 1!}{1} + \frac{2^2 \cdot 2!}{2^2} + \frac{2^3 \cdot 3!}{3^3} + \dots + \frac{2^n \cdot n!}{n^n} + \dots$	
6.8.	$S = \frac{(1!)^2}{2} + \frac{(2!)^2}{2^2} + \frac{(3!)^2}{2^3} + \dots + \frac{(n!)^2}{2^n} + \dots$	
6.9.	$S = \frac{1}{\sqrt{2}} + \frac{1}{2\sqrt{3}} + \frac{1}{3\sqrt{4}} + \dots + \frac{1}{n\sqrt{n+1}} + \dots$	
6.10.	$S = \frac{1}{\sqrt{1 \cdot 3}} + \frac{1}{\sqrt{3 \cdot 5}} + \dots + \frac{1}{\sqrt{(2n-1)(2n+1)}} + \dots$	
6.11.	$S = \sum_{n=1}^{\infty} \left(\frac{1 + \cos n}{2 + \cos n} \right)^{2n - \ln n}$	6.16. $S = \sum_{n=1}^{\infty} \frac{\cos 3^n}{n^2}$
6.12.	$S = \sum_{n=1}^{\infty} \frac{\cos(3n) - \cos[3(n+1)]}{n}$	6.17. $S = \sum_{n=1}^{\infty} \frac{\sin 5n}{2^n}$
6.13.	$S = \sum_{n=1}^{\infty} \frac{n^2 [\sqrt{2} + (-1)^n]^n}{3^n}$	6.18. $S = \sum_{n=1}^{\infty} \frac{n^2}{(2 + 1/n)^n}$
6.14.	$S = \sum_{n=1}^{\infty} \frac{\sqrt{n+2} - \sqrt{n-2}}{n^2}$	6.19. $S = \sum_{n=1}^{\infty} \frac{n^5}{2^n + 3^n}$
6.15.	$S = \sum_{n=1}^{\infty} \left(\frac{1}{\sqrt{n}} - \sqrt{\ln \frac{n+1}{n}} \right)$	6.20. $S = \sum_{n=1}^{\infty} \frac{\cos^2 \frac{n\pi}{3}}{2^n}$

5. VERİLƏNLƏRİN SADƏ TIPLƏRİ

5.1. Verilənlərin tiplərinin struktur təsviri

Turbo Pascal-da bütün verilənlər, o cümlədən konstantalar, dəyişənlər, funksiyalar və s. öz tipləri ilə xarakterizə olunurlar. Tip verilənin ala biləcəyi qiymətlər çoxluğunu, onunla aparılan mümkün əməliyyatları təyin edir və verilənin məşındaxili təsvir formatını müəyyənləşdirir. Verilənlərin tipləri şəkil 5.1-də göstərilmişdir.



Şəkil 5.1. Verilənlərin tipləri

Sadə tiplərə nizamlı və həqiqi tiplər daxildir. Nizamlı tipli verilənlərdən hər biri sonlu sayda mümkün qiymətlər ala bilər. Bu qiymətləri müəyyən qaydada nizamlamaq mümkündür və bu qiymətlərdən hər birinə müəyyən tam ədəd – qiymətin nömrəsi qarşı qoyulur. Həqiqi tip verilənlər həqiqi ədədlərin məşındaxili təsvir formatı ilə müəyyən edilir. Bu verilənlərin qiymətlərinin sayı sonlu olsa da, o qədər böyükdür ki, onlara sıra nömrəsi qarşı qoymaq mümkün deyil.

Sadə tipli dəyişənlərdən fərqli olaraq, **strukturlaşdırılmış tip** dəyişənlər çox sayda elementdən təşkil olunurlar. Başqa sözlə, strukturlaşdırılmış tip konstantanın və ya dəyişənin birdən çox komponenti olur. Hər bir komponent özü də strukturlaşdırılmış tipli ola bilər. Bu o deməkdir ki, strukturlaşdırılmış tiplərdə **ic-icəlik** xüsusiyyəti vardır və bu xüsusiyyətin dərinliyinə heç bir məhdudiyyət qoyulmur. Lakin, nəzərə

almaq lazımdır ki, bu tip dəyişənlərin məşindəxili təsviri zamanı baytlarla uzunluğu 65520 ədədini aşmamalıdır.

Turbo Pascal-da 4 sayda strukturlaşdırılmış tiptən istifadə olunur: massivlər, yazılar, çoxluqlar və fayllar. Turbo Pascal-in standart Pascal ilə uyumasını təmin etmək üçün strukturlaşdırılmış tip dəyişənini təsvir edərkən, ondan əvvəl **Packed** xidməti sözündən istifadə etməyə imkan verilir. Bu söz ilə kompilyatora strukturlaşdırılmış tip dəyişənlər üçün ayrılan yaddaşa mümkün qədər qənaət etmək göstərişi çatdırılır. Lakin, Turbo Pascal faktiki olaraq bu sözə məhəl qoymur və strukturlaşdırılmış tip verilənlərin yaddaşa optimal yerləşdirilməsini özü təmin edir.

5.2. Tam tiplər

Nizamlı tiplərə tam, məntiqi, simvol, sadalanan və diapazon tiplər daxildir. Bu dəyişənlərə cədvəl 5.1-də göstərilən funksiyalar tətbiq oluna bilər. Tam tipli dəyişənlərin ala biləcəyi qiymətlər diapazonu onun məşindəxili təsvir formatından asılıdır və bu formata əsasən dəyişənlərin qiymətləri üçün yaddaşa cədvəl 5.2-də göstəriləni kimi 1, 2 və ya 4 bayt uzunluqlu sahələr ayrılır.

Cədvəl 5.1

Nizamlı tip dəyişənlərə tətbiq olunan funksiyalar

Nö	Adı	Təyinatı
1	Dec	Dəyişənin qiymətini verilmiş kəmiyyət qədər azaldır
2	Inc	Dəyişənin qiymətini verilmiş kəmiyyət qədər artırır
	Odd	Arqumentin tək ədəd olmasını yoxlayır
4	Pred	Nizamlı tip dəyişənin qiymətləri siyahısından cari elementdən əvvəlki elementi qaytarır
5	Succ	Nizamlı tip dəyişənin qiymətləri siyahısından cari elementdən sonrakı elementi qaytarır
6	Ord	Nizamlı tip dəyişənin qiymətinə uyğun sıra nömrəsini qaytarır
7	Chr	Göstərilən sıra nömrəsinə uyğun simvolu qaytarır

Tam ədəd parametrli funksiyalar və prosedurlarla işləyərkən tiplərin iç-içəlik xassəsini nəzərə almaq lazımdır. Başqa sözlə dəyişənin tipi kimi Integer qəbul edilərsə, onu Byte, ShortInt və ya Word ilə, ShortInt tipini Byte tipi ilə əvəz etmək olar və s. Hökmün tərsi isə doğru deyil, yəni Word tipinin Integer tipi ilə əvəz olunması və s. mümkün olmaya da bilər. Cədvəl 5.3-də tam tipli dəyişənlərə tətbiq olunan

funksiyalar və prosedurlar göstərilmişdir. Onları ayrı-ayrılıqda nəzərdən keçirək:

Cədvəl 5.2

Tam tipli dəyişənlərin məhsındaxili təsvir formatları və qiymətləri diapazonu

Tam tip	Uzunluğu, bayt	Qiymətləri diapazonu
Byte	1	0 ... 255
ShortInt	1	-128 ... 127
Word	2	0 ... 65 535
Integer	2	-32 768 ... 32 767
LongInt	4	-2 147 483 648 ... 2 147 483 647

- **Abs(x: Byte/.../LongInt): Byte/.../LongInt;**

Funksiya icra olunarkən arqumentin modulunu qaytarır. X arqumenti ShortInt, Integer və ya LongInt tiplərindən biri ilə təsvir olunur. Byte və Word tipləri istisna olunmasa da, funksiya üçün onlar məna daşımır. Nəticə funksiyanın tipinə uyğun götürülür. Məsəl:

Var x1,y1: ShortInt; x2,y2: Integer; x3,y3: LongInt;

```

***
x1:=-100;      y1:=Abs(x1);  {Nəticə:      y1:=100}
x2:=-32000;    y2:=Abs(x1);  {Nəticə:      y2:=32000}
x3:=-1500000;  y3:=Abs(x1);  {Nəticə:      y3:=1500000}
x2:=1500;      y2:=Abs(x1);  {Nəticə:      y2:=1500}

```

- **Chr(x: Byte): Char;**

Funksiya icra olunarkən ASCII kodlarda verilmiş Byte tipli x ədədini bu koda uyğun simvola çevirir. Məsəl:

Var x: Byte; s: Char;

```

***
x:=77; s:=Chr(x);      {Nəticə:      s='M'}
x:=85; s:=Chr(x);      {Nəticə:      s='U'}
x:=50; s:=Chr(x);      {Nəticə:      s='2'}

```

- **Ord(x: Char): Byte;**

Funksiya icra olunarkən x simvolunun ASCII-cədvəldəki uyğun kodunu qaytarılır. Məsəl:

Var x: Char; y: Byte;

```

***
x:='M'; y:=Ord(x);      {Nəticə:      y=77}
x:='U'; y:=Ord(x);      {Nəticə:      y=85}
x:='2'; y:=Ord(x);      {Nəticə:      y=50}

```

Tam tip dəyişənlərə tətbiq olunan funksiyalar və prosedurlar

No	Adı	Təyinatı
Funksiyalar		
1	Abs	x ədədinin modulunu qaytarır. Formatı: Abs(x: Byte/.../LongInt): Byte/.../LongInt;
2	Chr	ASCII koda uyğun simvolu qaytarır. Formatı: Chr(x: Byte): Char;
3	Ord	Simvola uyğun ASCII kodu qaytarır. Formatı: Ord(x: Char): Byte;
4	Hi	x arqumentinin yerləşdiyi sol baytı qaytarır. Formatı: Hi(x: Word/Integer): Byte;
5	Lo	x arqumentinin yerləşdiyi sağ baytı qaytarır. Formatı: Lo(x: Word/Integer): Byte;
6	Odd	x arqumenti tək ədəd olduqda, True qiymətini qaytarır. Formatı: Odd(x: Byte/.../LongInt): Boolean;
7	Random	[0, x-1] intervalında bərabər paylanmış təsadüfi ədədi qaytarır. Formatı: Random(x: Byte/.../Integer): Byte/.../Integer;
8	Sqr	Arqumentin kvadratını qaytarır. Formatı: Sqr(x: Byte/.../LongInt): Byte/.../LongInt;
9	Swap	Yarımsözdə sağ və sol baytların qiymətlərini dəyişir. Formatı: Swap(x: Word/Integer): Word/Integer;
Prosedurlar		
10	Dec	x ədədinin qiymətini y vahid azaldır. Formatı: Dec(x: Byte/.../LongInt, y: Byte/.../LongInt);
11	Inc	x ədədinin qiymətini y vahid artırır. Formatı: Inc(x: Byte/.../LongInt, y: Byte/.../LongInt);
12	Randomize	Təsadüfi ədədlər generatorunun aktivləşdirilməsi. Formatı: Randomize;

- Hi(x: Word/Integer): Byte;

Funksiya icra olunarkən, x arqumentinin 16-lıq rəqəmlərlə ifadə olunmuş qiymətinin yaddaşda yerləşdiyi yarımsözün sol baytı qaytarılır. İzahat: onluq 270 ədədinin 16-lıq rəqəmlərlə ekvivalenti $10E_{16}$ kimidir ($E_{16} = 14_{10}$; $270_{10} = 10E_{16} = 1 * 16^2 + 0 * 16 + 14$). Yaddaşda bu ədəd

01	0E
----	----

kimi təsvir olunur. Misal,

x:=254; y:=Hi(x);

{ 254₁₀=0FE₁₆; Nəticə: y=0 }

x:=270; y:=Hi(x);

{ 270₁₀=10E₁₆; Nəticə: y=1 }

x:=403; y:=Hi(x); { 403₁₀=193₁₆; Nəticə: y=1}

- Lo(x: Word/Integer): Byte;

Funksiya icra olunarkən, x arqumentinin 16-lıq rəqəmlərlə ifadə olunmuş qiymətinin yaddaşda yerləşdiyi yarımsözün sağ baytı qaytarılır. Hi funksiyasına bax. Misal,

x:=254; y:=Lo(x); {254₁₀=0FE₁₆; FE₁₆=15*16+14=254; Nəticə: y=254}

x:=270; y:=Lo(x); {270₁₀=10E₁₆; 0E₁₆=0*16+14=14; Nəticə: y=14}

x:=403; y:=Lo(x); {403₁₀=193₁₆; 93₁₆=9*16+3=147; Nəticə: y=147}

- Odd(x: Byte/ .../LongInt): Boolean;

Funksiya icra olunarkən, x ədədinin tək və ya cüt ədəd olması yoxlanılır. Əgər x ədədi təkdirsə, funksiya True, cütdürsə False qiymətini qaytarır. Misal,

Var x: Integer; y: Boolean;

...

x:=204; y:=Odd(x); {Nəticə: y=False}

x:=341; y:=Odd(x); {Nəticə: y=True}

- Random(x: Byte/.../Integer): Byte/.../Integer;

Funksiya icra olunarkən, [0, x-1] intervalında bərabər paylanmış təsadüfi ədəd qaytarılır. Funksiyadan istifadə olunmamışdan qabaq Randomize prosedurunu icra etmək tövsiyə olunur (Bax: Randomize). Misal,

Var x, y: Word;

...

x:=100;

y:=Random(x); {Nəticə: y=76}

y:=Random(x); {Nəticə: y=32}

- Sqr(x: Byte/ .../LongInt): Byte/ .../LongInt;

Funksiya icra olunarkən, arqumentin kvadratı qaytarılır. Misal,

Var x, y: Word;

...

x:=75; y:=Sqr(x); {Nəticə: y=5625}

- Swap(x: Word/Integer): Word/Integer;

Funksiya icra olunarkən x arqumentinin 16-lıq ekvivalentinin yaddaşda yerləşdiyi yarımsözün sağ və sol baytlarının qiymətləri öz yerlərini qarşılıqlı olaraq dəyişdirir. Onluq 759 ədədinin 16-lıq ekvivalenti 2F7₁₆ kimidir və yaddaşda yarımsözdə yerləşir:

02	F7
----	----

Funksiyanın icrasından sonra, sağ və sol baytların qiymətləri qarşılıqlı yerlərini dəyişirlər:

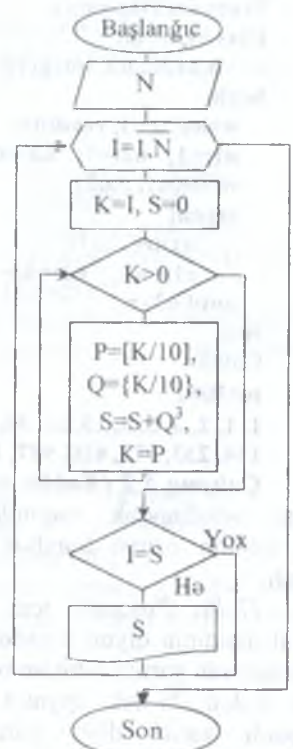
4) $I=371=1^3+7^3+3^3=S$ olduğundan bu ədəd cavab kimi qəbul edilir.

Programın həll algoritminin blok-sxemi şəkil 5.3-də, listinqi və icrasından alınan cavablar isə aşağıda göstərilmişdir.

```

Program KubCem;
Uses WinCrt;
var I,N,K,S,P,Q: integer;
begin
    write('N='); readln(N);
    for I:=1 to N do
        begin
            K:=I; S:=0;
            while K>0 do begin
                P:=K div 10;
                Q:=K mod 10;
                S:=S+Q*Q*Q;
                K:=P;
            end;
            if I=S then
                Writeln('Cavab=',S:5);
            end;
        end.
    Cavab:
    N=3000
    Cavab= 1
    Cavab= 153
    Cavab= 370
    Cavab= 371
    Cavab= 407

```



Şəkil 5.3

Çalışma 5.3 (Dost ədədlər). İki natural ədəddən birinin özündən savayı bütün bölənlərinin (vahid də daxil olmaqla) cəmi digərinə bərabədirsə, bu ədədlər dost ədədlər adlanır. Qiymətləri 3000-dən böyük olmayan dost ədədləri tapmalı.

Həlli: 220 və 284 ədədləri dost ədədlərdir, çünki 220 ədədinin özündən savayı bütün bölənlərinin cəmi 284-ə, və tərsinə, 284 ədədinin özündən savayı bütün bölənlərinin cəmi 220-yə bərabərdir:

220 ədədi üçün - $1+2+4+5+10+11+20+22+44+55+110=284$;

284 ədədi üçün - $1+2+4+71+142=220$.

Məsələnin şərti daxilində 3000-dən böyük olmayan ($T=3000$) natural dost ədədləri tapmaq üçün, iç-içə yerləşən iki dövrədən istifadə

olunur. Hesablamalarda təkrarlamalara yol verilməməsindən ötrü, dostluq münasibətinin təyini üçün aparılan müqayisə zamanı birinci ədədin ikincidən kiçik olması şərti nəzərə alınmışdır. Xarici dövrdəki dövr dəyişəni (I) birinci ədədi ifadə edir və 1-dən (T-1)-dək natural qiymət alır. Daxili dövrdəki dövr dəyişəni (J) isə müqayisə üçün ikinci ədədi ifadə edir və hər bir xarici dövrün icrası zamanı I natural ədədindən sonrakı (I+1) ədədindən T-dək natural qiymətləri qəbul edir.

İstənilən natural ədədin özündən savayı böləni bu ədədin yarısından böyük ola bilməz. Proqramın icra vaxtına qənaət etmək üçün bu hökm nəzərə alınır və daxili dövrə hər bir J ədədi üçün onun bölənləri 1-dən $[J/2]$ natural ədədləri içərisində axtarılır. J ədədinin özündən savayı bölənlərinin cəmi S1 dəyişənində toplanır. J ədədi üçün bölənlərinin cəminin tapılması prosesi başa çatdıqdan sonra, cəmi özündə saxlayan S1 dəyişəninə qiymətinin I dəyişəninə qiymətinə bərabər olması şərti yoxlanılır. Bu qiymətlər bir-birindən fərqlidirsə, onda I və J ədədləri dost ola bilməz. Bu halda J dəyişəni daxili dövrə öz qiymətindən sonrakı növbəti qiyməti alır və daxili dövr davam edir. Əks halda I və J ədədləri arasında dostluq münasibətinin olması ehtimalı var və bu münasibət aqaloji olaraq I ədədi üçün yoxlanılır. I ədədinin özündən savayı bölənlərinin S2 cəmi J ədədinə bərabər olarsa, I və J ədədləri dost ədədlərdir.

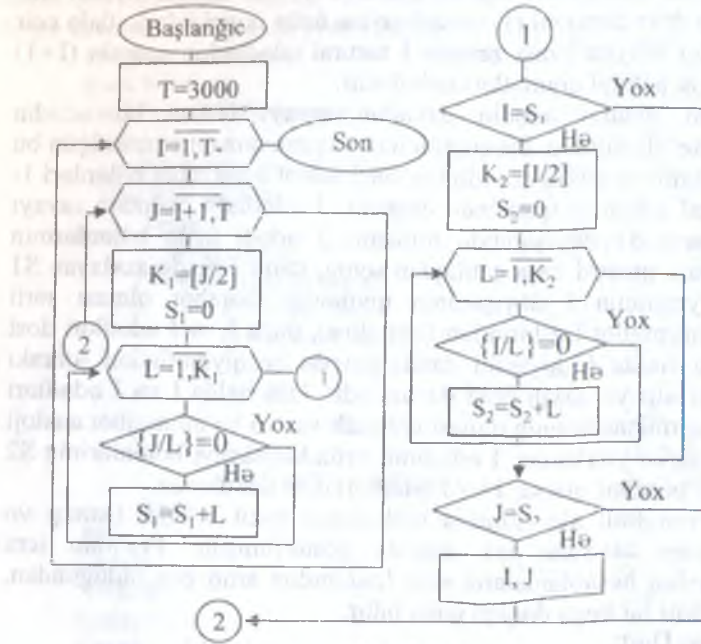
Proqramın həll alqoritminin blok-sxemi şəkil 5.4-də, listingi və icrasından alınan cavablar isə aşağıda göstərilmişdir. Proqram icra olunarkən aparılan hesablamaların sayı həddindən artıq çox olduğundan, onun icra müddəti bir neçə dəqiqə çəkə bilər.

```

Program Dost;
Uses WinCrt;
const T=3000;
var I,J,L,K1,K2,S1,S2: integer;
begin
  for I:=1 to T-1 do
    for J:=I+1 to T do begin
      K1:=J div 2;  S1:=0;
      for L:=1 to K1 do if (J mod L)=0 then S1:=S1+L;
      if I=S1 then begin
        K2:=I div 2;  S2:=0;
        for L:=1 to K2 do if (I mod L)=0 then S2:=S2+L;
        if J=S2 then writeln(I, ' və ', J)
        end;
      end;
    end;
end.

```

Cavab:
220 ve 284
1184 ve 1210
2620 ve 2924



Şəkil 5.4

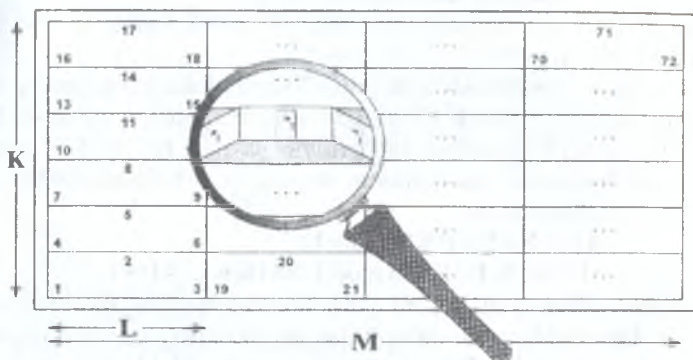
5.3. Məntiqi tiplər

Məntiqi dəyişənlər Boolean tipi ilə təsvir edilir və **True** – "doğru", **False** – "yalan" qiymətlərindən birini alır. Məntiqi tip **verilənlər** üçün aşağıdakı ifadələr doğrudur:

Ord(False) = 0; Ord(True) = 1;
Succ(False) = True; Pred(True) = False;
False < True;

Çalışma 5.4 (Bina). Yasamal rayonundakı çoxmərtəbəli binalardan birinə qonaq getmək üçün həmin binaya yaxınlaşmışıq. Həyətdə dayanıb binaya nəzər saldıqdan sonra, binadakı blokların və mərtəbələrin

sayı bizə məlum oldu. Blokun hər mərtəbəsində yerləşən mənzillərin sayını bilərək, qonaq gedəcəyimiz mənzilin nömrəsinə görə blokun və mərtəbənin nömrəsini təyin etməli (şəkil 5.5).



Şəkil 5.5

Həlli: Fərz edək ki, yaxınlaşdığımız bina M sayda blokdan və K sayda mərtəbədən ibarətdir. Blokun hər bir mərtəbəsində L sayda mənzil yerləşir. Deməli, binada $M \cdot K \cdot L$ sayda mənzil var. Xüsusi halda, bina 4 bloklu, 6 mərtəbəlidirsə və hər blokun mərtəbəsində 3 mənzil yerləşirsə ($M=4, K=6, L=3$), onda binada $4 \cdot 6 \cdot 3=72$ sayda mənzil olar. Əgər, bizə verilən mənzilin N nömrəsi $M \cdot K \cdot L$ hasilini aşmırsa, onda hər şey öz qaydasındadır və biz qonaqlıqdan qalmayacağıq. Əks halda, geri qayıtmaq olar, çünki binada bizə verilən N nömrəli mənzil ola bilməz!

İlk növbədə biz tələb olunan mənzilin hansı blokda yerləşdiyini təyin edək. Aydındır ki, hər bir blokda binanın mərtəbələrinin sayının blokun mərtəbəsidəki mənzillərin sayına hasili qədər mənzil var, yəni binanın hər blokundakı mənzillərin sayı $K \cdot L$ qədərdir. Bizim misal üçün hər blokda mənzillərin sayı $6 \cdot 3=18$ olar. Gedəcəyimiz mənzilin N nömrəsi blokda mənzillərin $K \cdot L$ sayına qalıqsız bölünürsə, alınan T qisməti bizə lazım olan blokun nömrəsidir. Qeyd etmək lazımdır ki, sonuncu hesablamada bölmə əməliyyatından alınan qalıq yalnız o zaman 0 qiymət alar ki, gedəcəyimiz mənzilin nömrəsi blokun sonuncu mərtəbəsidəki sonuncu mənzil olsun. Bizim misalda, blokda sonuncu mənzil nömrələri (18, 36, 54, 72) bir blokda mənzillərin sayının (18) misilləri olduğundan, bölmə əməliyyatından alınan qalıq 0 olacaqdır. Digər hallarda, yəni bölmə əməliyyatından alınan qalıq 0-dan böyükdürsə,

onda T qismətinin üzərinə bir vahid əlavə edilməlidir. Blok nömrəsinin təyini aşağıdakı operatorlarla tapıla bilər:

T:=N div (K*L);
if N mod (K*L)>0 then T:=T+1;

Bizim misalda N=47 qəbul etsək, onda $47:18=2$ (qalıq 11) və blokun nömrəsi $2+1=3$ olar.

Beləliklə, hesablayıb tapdığımız T sayılı bloka yaxınlaşmaq olar!

Mərtəbənin nömrəsi (U), gedəcəyimiz mənzilin nömrəsi (N) ilə T-1 sayda əvvəlki bloklardakı mənzillərin sayının $((T-1)*K*L)$ fərqi blokun mərtəbəsindəki mənzillərin sayına (L) bölünməsindən alınan qismət kimi hesablanı bilər:

U:=(N-(T-1)*K*L) div L;
if (N-(T-1)*K*L) mod L>0 then U:=U+1;

Bölmə əməliyyatı zamanı qalıq 0-ra bərabərsə, mərtəbə nömrəsi tapılmışdır. Əks halda, yəni qalıq 0-dan böyükdürsə, qismətin üzərinə bir vahid əlavə edilməlidir. Bizim misalda əvvəlki bloklardakı mənzillərin sayı $(3-1)*6*3$ olduğundan, $U=(47-36):3=3$ (qalıq 2), $U=3+1=4$. Deməli, axtardığımız 47 sayılı mənzil 4-cü mərtəbədədir. Beləliklə, mərtəbə nömrəsi də bizə məlum oldu. Artıq bloka daxil olub, mənzilə yaxınlaşmaq olar!

Proqramda d məntiqi tipli dəyişənindən sonsuz dövrün təşkili üçün istifadə olunmuşdur. İstifadəçi proqramın icrası zamanı tələb olunan mənzilin hansı blokda və hansı mərtəbədə yerləşdiyini bilmək üçün istədiyi sayda mənzil nömrəsi daxil edə bilər. Dövrlərin sayına proqramda heç bir məhdudiyyət qoyulmamışdır. d məntiqi dəyişəni nə qədər True qiymətinə malik isə, dövr davam edəcəkdir. Lakin o, proqramın icrasını başa çatdırmaq istəsə, buna imkan var. Bu məqsədlə "Mənzilin nömrəsi?" sualına sıfır və ya mənfi qiymət daxil etmək kifayətdir. Bununla d məntiqi dəyişəninə False qiyməti mənsub edilir və dövrdən çıxış baş verir.

Məsələnin həlli üçün aşağıdakı proqramdan istifadə etmək olar:

```
Program Bina;  
Uses WinCrt;  
var M,K,L,N,T,U: integer;  
d: boolean;  
begin  
write('Blokların sayı: '); readln (M);  
write('Mərtəbələrin sayı: '); readln (K);  
write('Blok və mərtəbədəki mənzillərin sayı: '); readln (L);  
d:=true;  
while d do begin
```

```

write('Menzilin nomresi: '); read(N);
if N<=0 then d:=false
else if N>M*K*L then write('Ele nomre yoxdur!')
else begin
    T:=N div (K*L);
    if N mod (K*L)>0 then T:=T+1;
    U:=(N-(T-1)*K*L) div L;
    if (N-(T-1)*K*L) mod L>0 then U:=U+1;
    write('Blok=',T:2,' Mertebe=',U:2);
    end; writeln;
end;

```

end.

Cavab:

Blokların sayı: 4

Mertebelerin sayı: 6

Blok və mertebedeki menzillərin sayı: 3

Menzilin nomresi: 27

Blok=2 Mertebe=3

Menzilin nomresi: 47

Blok=3 Mertebe=4

Menzilin nomresi: 100

Ele nomre yoxdur!

Menzilin nomresi: 72

Blok=4 Mertebe=6

Menzilin nomresi: 0

5.4. Simvol tipləri

Simvol dəyişənləri Char tipi ilə təsvir edilir. Hər bir simvola [0,255] intervalında bir tam ədəd qarşı qoyulur və bu ədəd simvolun ASCII (American Standard Code for Information Interchange) kodlarda məşin daxili təsviri üçün istifadə olunur. ASCII standart kodları baytın 7 bitində yerləşir və 0-dan 127-dək 128 sayda simvolu əhatə edir. Baytın 8-ci bitindən istifadə etməklə simvolların sayını 2 dəfə artırmaq və 256 sayda simvolu kodlaşdırmaq olar. 128-dən 255-dək kodlu simvollar müxtəlif tip fərdi kompüterlərdə müxtəlif ola bilər. Standart ASCII kodlardan bir qismi cədvəl 5.4-də verilmişdir.

ASCII-cədvəldə 0-dan 31-dək kodlar xüsusi təyinatlıdır. Daxiletmə və xaricətmə əməliyyatlarında onlar müstəqil mənə daşıyırlar. Onlardan bəzilərini nəzərdən keçirək:

Standart ASCII kodları və onlara qarşı qoyulan simvollar

Kod	Sim.	Kod	Sim.	Kod	Sim.	Kod	Sim.	Kod	Sim.	Kod	Sim.
32	aralıq	48	0	64	@	80	P	96	*	112	p
33	!	49	1	65	A	81	Q	97	a	113	q
34	"	50	2	66	B	82	R	98	b	114	r
35	#	51	3	67	C	83	S	99	c	115	s
36	\$	52	4	68	D	84	T	100	d	116	t
37	%	53	5	69	E	85	U	101	e	117	u
38	&	54	6	70	F	86	V	102	f	118	v
39	'	55	7	71	G	87	W	103	g	119	w
40	(	56	8	72	H	88	X	104	h	120	x
41	)	57	9	73	I	89	Y	105	i	121	y
42	*	58	:	74	J	90	Z	106	j	122	z
43	+	59	;	75	K	91	[	107	k	123	{
44	,	60	<	76	L	92	\	108	l	124	
45	-	61	=	77	M	93	]	109	m	125	}
46	.	62	>	78	N	94	^	110	n	126	~
47	/	63	?	79	O	95	_	111	o	127	□

7 – Zəng. Simvolun ekranda əks olunması səs signalı ilə müşayiət olunur;

9 – Üfüqi tabulyasiya. Simvolun ekranda əksi zamanı kursurun mövqesi 8-in misli+1 vahid (9, 17, 25, 33, ...) sürüşür;

10 – Növbəti sətirə keçid. Simvolun ekranda əksi alındıqdan sonra, qalan simvollar növbəti sətirin həmin mövqesindən başlayaraq ekrana çıxarılır;

11- Şaquli tabulyasiya. Ekranı çıxarılarkən xüsusi işarə ilə əvəz olunur;

12 – Yeni səhifənin önünədək sürüşmə; printerə çıxış zamanı yeni səhifə formalaşdırır, ekrana əks olunarkən isə, xüsusi işarə ilə əvəz olunur;

13 – Karetkanın qaytarılması. Enter klavişini sıxarkən daxil edilir, read və readln ilə daxiletmə zamanı daxiletmə əməliyyatını təyin edir və daxiletmə buferində yerləşdirilmir. Xaricetmə zamanı "Cari sətirin başlanğıcından xaricetməni davam etmək" əmrini icra edir;

26 – Faylın sonu. Klaviaturada Ctrl+Z klavişini sıxmaqla daxil edilir, xaricetmə zamanı xüsusi işarə ilə əvəz olunur;

27 – İşin sonu. Klaviaturadan ESC klavişini sıxmaqla daxil edilir, xaricetmə zamanı xüsusi işarə ilə əvəz olunur və s.

Cədvəl 5.5-də simvol tipli verilənlərlə işləyən funksiyalar verilmişdir. Aşağıda bu funksiyalara misallar göstərilmişdir:

Cədvəl 5.5

Simvol tipli dəyişənlərə tətbiq olunan funksiyalar

Nö	Adı	Təyinatı
1	Chr	ASCII koda uyğun simvolu qaytarır. Formatı: Chr(x: Byte): Char;
2	Ord	Simvola uyğun ASCII kodu qaytarır. Formatı: Ord(x: Char): Byte;
3	UpCase	Latın kiçik hərflərini böyük hərflərə çevirir. Formatı: UpCase(x: Char): Char;
4	Pred	ASCII-cədvəldə x simvolundan əvvəlki simvolu qaytarır. Formatı: Pred(x: Char): Char;
5	Succ	ASCII-cədvəldə x simvolundan sonrakı simvolu qaytarır. Formatı: Succ(x: Char): Char;

```
var x,y: Char; t:Byte;
```

```
...
```

```
x:='b'; y:=Pred(x); {Nəticə: y='a'}
y:=Succ(x); {Nəticə: y='c'}
y:=UpCase(x); {Nəticə: y='B'}
t:=Ord(x); {Nəticə: t=98}
y:=Chr(ord(x)); {Nəticə: y='b'}
```

5.5. Sadalanan tiplər

Sadalanan tip dəyişənləri ala biləcəyi qiymətlərin sadalanma ardıcılığı ilə təsvir edilir. Dəyişənin hər bir qiyməti ümumi qəbul edilmiş identifikatorla adlandırılır və kiçik mötərizələr daxilində siyahıda yerləşdirilir. Məsələn,

Type TFesil=(yaz, yay, payız, gish);

Sadalanan tip dəyişənin hər bir qiymətinə təsvir olunduğu siyahıdakı sıra nömrəsi qarşı qoyulur. Siyahıdakı birinci qiymət 0 saylı, ikinci qiymət 1 saylı və s. hesab edilir. Sadalanan dəyişənlərə Ord, Pred və Succ

funksiyalarını tətbiq etmək olar. Bu funksiyaların formatları və təyinatı cədvəl 5.6-da göstərilmişdir.

Cədvəl 5.6

Sadalananan tipli dəyişənlərə tətbiq olunan funksiyalar

Nö	Adı	Təyinatı
1	Ord	Qiyəmetin siyahıdakı sıra nömrəsini qaytarır. Formatı: Ord(x: <Sadalananan tip>): Byte;
2	Pred	Siyahıda x qiyəmetindən əvvəlki qiyəmeti qaytarır. Formatı: Pred(x: <Sadalananan tip>): <Sadalananan tip>;
3	Succ	Siyahıda x qiyəmetindən sonrakı qiyəmeti qaytarır. Formatı: Succ(x: <Sadalananan tip>): <Sadalananan tip>;

Sadalananan dəyişənlərlə mənsubətmə, müqayisə əməliyyatları aparmaq mümkündür, lakin onları daxil və xaric etmək olmaz. Misal:

Var Fesil: TFesil; t: Byte; b: boolean;

```

...
t:=Ord(yaz);           {Nəticə: t=0}
t:=Ord(payiz);         {Nəticə: t=2}
if yaz < yaz then b:=true else b:=false; {Nəticə: b=True}
Fesil:=Pred(yaz);      {Nəticə: Fesil=yaz}
if Fesil < yaz then b:=true else b:=false; {Nəticə: b=True}
Fesil:=Succ(yaz);      {Nəticə: Fesil=payiz}
if Fesil < yaz then b:=true else b:=false; {Nəticə: b=False}

```

5.6. Diapazon tipləri

Diapazon tip baza tip olan digər tipin alt hissəsi olub, dəyişənin ala biləcəyi qiymətlərin sərhədləri ilə təsvir olunur. Baza tipi kimi Integer, Boolean, Char və sadalananan tiplərdən biri ola bilər. Məsələn,

Type

NumHefte = 1..7;

Elifba = 'A'..'Z';

Diapazon tipli dəyişəni Type bölməsində göstərməyərək, birbaşa Var bölməsində də təsvir etmək olar:

Var

NumHefte: 1..7;

Elifba: 'A'..'Z';

Təsviretmə zamanı ".." bir simvol kimi qəbul edilir və nöqtələrarası "aralıq" simvoluna yol verilmir. Diapazonda sağda yerləşən yuxarı sərhəd, solda yerləşən aşağı sərhəddən böyük olmalıdır. Diapazon

tipli dəyişənlər üçün Ord funksiyasından istifadə etmək olar, lakin Pred və Succ funksiyalarının tətbiqi mümkün deyil. Misal,

Type

TAy = (yanv, fevr, mart, apre, may, iyun,
iyul, avqu, sent, okty, noya, deka);
TYay=iyun..avqu;

Var

Ay: TAY; Yay: TYay; t:Word;

Yay:=avqu;

t:=ord(Yay);

{Nəticə:7}

5.7. Həqiqi tiplər

Nizamlı tiplərdən fərqli olaraq, həqiqi tip verilənlər məşindəxili formatdan asılı olaraq müəyyən dəqiqliklə təyin olunurlar. Turbo Pascal-da həqiqi tipin 5 müxtəlifliyi vardır və hər bir tip müəyyən diapazona və dəqiqliyə malikdir (cədvəl 5.7). Bu tiplər içərisində kəsr və eksponensial hissəsi olmayan Comp tipi xüsusi maraq kəsb edir. Comp tipli dəyişənin qiyməti yaddaşda 8 yanaşı baytda yerləşərək, 19..20 sayda qiymətli onluq rəqəmlərdən və işarədən ibarət olur. Bu tip digər həqiqi tiplərlə uyuşur, onlarla bütün həqiqi əməliyyatları aparmaq mümkündür, bütün riyazi funksiyalardan istifadə etmək olar və s.

Cədvəl 5.7

Həqiqi tip dəyişənlərin məşindəxili təsvir formatları
və qiymətləri diapazonu

Həqiqi tip	Uzunluğu, bayt	Qiymətli rəqəmlərin sayı	Qiymətləri diapazonu
Real	6	11 - 12	2,9E-39 - 1,7E+38
Single	4	7 - 8	1,5E-45 - 3,4E+38
Double	8	15 - 16	5,0E-324 - 1,7E+308
Extended	10	19 - 20	3,4E-4932 - 1,1E+4932
Comp	8	19 - 20	-9,2E+18 - 9,2E+18

Cədvəl 5.8-də həqiqi tip dəyişənlərə tətbiq olunan funksiyalar göstərilmişdir. Onlardan bir qismini nəzərdən keçirək:

- Int(x: Real): Real;

Funksiya icra olunarkən argumentin kəsr hissəsi tamamilə atılır və tam hissəsi qaytarılır. Misal:

Həqiqi tip dəyişənlərə tətbiq olunan funksiyalar

No	Adı	Təyinatı
1	Abs	x ədədinin modulunu qaytarır. Formatı: Abs(x: Real): Real;
2	ArcTan	x ədədinin arktangensini qaytarır. Formatı: ArcTan(x: Real): Real;
3	Sin	x ədədinin sinusunu qaytarır. Formatı: Sin(x: Real): Real;
4	Cos	x ədədinin kosinusunu qaytarır. Formatı: Cos(x: Real): Real;
5	Exp	x ədədinin eksponentasını qaytarır. Formatı: Exp(x: Real): Real;
6	Ln	x ədədinin natural loqarifmini qaytarır. Formatı: Ln(x: Real): Real;
7	Sqr	x ədədinin kvadratını qaytarır. Formatı: Sqr(x: Real): Real;
8	Sqrt	x ədədinin kvadrat kökünü qaytarır. Formatı: Sqrt(x: Real): Real;
9	Int	x ədədinin tam hissəsini qaytarır. Formatı: Int(x: Real): Real;
10	Frac	x ədədinin kəsr hissəsini qaytarır. Formatı: Frac(x: Real): Real;
11	Pi	π ədədini qaytarır. Formatı: Pi: Real;
12	Trunc	x ədədinin tam hissəsini qaytarır. Formatı: Trunc(x: Real): Real/LongInt;
13	Round	x ədədinin tam hissəsini yuvarlaqlaşdırmaqla qaytarır. Formatı: Round(x: Real): Real/LongInt;

Var a, r: Real;

...

a:=3.62; r:=Int(a); {Nəticə: r=3}

a:=-2.1; r:=Int(a); {Nəticə: r=-2}

- Frac(x: Real): Real;

Funksiya icra olunarkən argumentin tam hissəsi tamamilə atılır və kəsr hissəsi qaytarılır. Məsəl:

Var a, r: Real;

...

a:=3.62; r:=Frac(a); {Nəticə: r=0.62}

```
a:=-2.1;      r:=Frac(a);      {Nəticə:      r=-0.1}
```

```
- Trunc(x: Real): Real/LongInt;
```

Funksiya icra olunarkən arqumentin kəsir hissəsi tamamilə atılır və tam hissəsi qaytarılır. Nəticə LongInt tipli götürülə bilər. Məsəl:

```
Var a: Real; r: LongInt;
```

```
***
a:=3.62;      r:=Trunc(a);      {Nəticə:      r=3}
```

```
a:=-2.1;      r:=Trunc(a);      {Nəticə:      r=-2}
```

```
- Round(x: Real): Real/LongInt;
```

Funksiya icra olunarkən arqument tam hissəyədək riyazi üsulla yuvarlaqlaşdır və onun tam hissəsi qaytarılır. Nəticə LongInt tipli götürülə bilər. Məsəl:

```
Var a: Real; r: LongInt;
```

```
***
a:=3.12;      r:=Round(a);      {Nəticə:      r=3}
```

```
a:=3.62;      r:=Round(a);      {Nəticə:      r=4}
```

```
a:=-2.1;      r:=Round(a);      {Nəticə:      r=-2}
```

```
a:=-2.71;     r:=Round(a);      {Nəticə:      r=-3}
```

```
- Pi: Real;
```

Funksiya icra olunarkən π ədədinin qiyməti qaytarılır. Məsəl:

```
Var a: Real;
```

```
***
a:=Pi;      {Nəticə:      a=3.14159}
```

Çalışma 5.5 (Detal). 5 fəhlə birlikdə 180 detal hazırladı. Onlardan biri bütün hazırlanmış detalların 1/6 hissəsi sayda, 2-ci və 3-cü fəhlə uyğun olaraq 20 və 25 faizi qədər, 4-cü fəhlə 13 detal, 5-ci fəhlə isə yerdə qalan sayda detal hazırladı. Fəhlələrin hazırladığı detalların sayını müəyyən etməli və nəticələri diaqram formasında təsvir etməli.

Həlli. Diaqramın qurulması üçün 136 kodundan istifadə edilmişdir. Proqramın həlli üçün proqramın listingi aşağıda, icrasından alınan nəticələr isə şəkil 5.6-də göstərilmişdir.

```
Program Detal;
```

```
uses WinCrt;
```

```
var i,d : integer;
```

```
f1,f2,f3,f4,f5: Real;
```

```
begin
```

```
d:=180;
```

```
f1:=d/6; f2:=d*20/100; f3:=d*30/100;
```

```
f4:=13; f5:=d-(f1+f2+f3+f4);
```

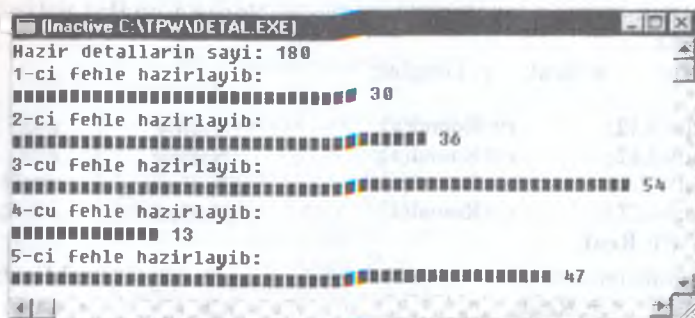
```
writeln('Hazir detallarin sayi: ',d);
```

```
writeln('1-ci fehle hazirlayib: ');
```

```

for i:=1 to Round(f1) do write(chr(136)); writeln(' ',Round(f1));
writeln('2-ci fehle hazirlayib: ');
for i:=1 to Round(f2) do write(chr(136)); writeln(' ',Round(f2));
writeln('3-cu fehle hazirlayib: ');
for i:=1 to Round(f3) do write(chr(136)); writeln(' ',Round(f3));
writeln('4-cu fehle hazirlayib: ');
for i:=1 to Round(f4) do write(chr(136)); writeln(' ',Round(f4));
writeln('5-ci fehle hazirlayib: ');
for i:=1 to Round(f5) do write(chr(136)); writeln(' ',Round(f5));
end.

```



Şəkil 5.6. "Detal" proqramının icrasından alınan nəticələr

5.8. Laboratoriya işi № 7. Verilənlərin sadə tipləri

Aşağıdakı məsələlərin alqoritmlərinin blok-sxemlərini qurun və proqramlarını tərtib edin. 1-8 sayılı tapşırıqlarda Randomize prosedurundan istifadə edin.

1. Verilmiş N natural ədədinin rəqəmlərinin sayını tapın.
2. Verilmiş N natural ədədinin rəqəmlərinin cəmini tapın.
3. Verilmiş N natural ədədinin soldan birinci rəqəmini tapın.
4. Verilmiş beşrəqəmli N natural ədədin tək sayılı mərtəbələrində duran rəqəmlərinin cəmindən, cüt sayılı mərtəbələrindəki rəqəmlərinin cəmini çıxın.

5. Dördədək rəqəmi olan N natural ədədinin palindrom ədəd olmasını müəyyənləşdirin. Palindrom ədədlərə misallar: 7337, 9669, 1111, 0220, 1001 və s.

6. Dördədək rəqəmi olan N natural ədədinin tərkibində üç eyni rəqəmin olduğunu müəyyənləşdirin. Məsələn, 7707, 2022, 0005, 0111, ...

7. Verilmiş N natural ədədinin bütün rəqəmlərinin bir-birindən fərqli olduğunu müəyyənləşdirin.

8. Verilmiş 7 rəqəmli N natural ədədinin tərkibindəki 3 və 8 rəqəmlərini atıb, qalan rəqəmlərin mərtəbə ardıcılığını gözləməklə yeni alınmış ədədi göstərin. Məsələn, 3768398 ədədindən 769 ədədi alınmalıdır.

9. Rəqəmləri cəminin kubu ədədin özünə bərabər olan dörd rəqəmli ədədləri tapın.

10. Rəqəmləri cəminin kubu ədədin kvadratına bərabər olan ikirəqəmli natural ədədi tapın.

11. Natural k dəyişəninin 1-dən 100-dək qiymətlərində hesablanmış $ke^{\cos^2(k+1)}$ ədədlərinin cəmini və bu ədədlərdən ən böyüyünü tapın.

12. $x_1=y_1=1$, $x_i = x_{i-1} + \frac{y_{i-1}}{i^2}$, $y_i = y_{i-1} + \frac{x_{i-1}}{i}$ düsturları ilə təyin edilmiş $x_1, x_2, \dots, x_{20}$ və $y_1, y_2, \dots, y_{20}$ ədədləri ardıcılıqlarını göstərin.

13. $u_1=1$, $u_2=u_1$, $u_3=u_1+u_2$, $u_4=u_1+u_2+u_3, \dots, u_n=u_1+u_2+u_3+\dots+u_{n-1}$ qaydası ilə götürülmüş ədədi ardıcılığın ilk 20 həddini tapın.

14. Onluq say sistemində verilmiş 56, 131, 546, 939, 1024 ədədlərini ikilik say sistemində ifadə edin.

15. Dövr təşkil edib aşağıdakı pilləli kəsrlərin qiymətlərini hesablayın:

$$\begin{array}{c} \frac{1}{1 + \frac{1}{2 + \frac{1}{3 + \frac{1}{4 + \frac{1}{\dots + \frac{1}{99 + \frac{1}{100}}}}}}} \end{array} \quad \begin{array}{c} \frac{1}{1 + \frac{2}{1 + \frac{3}{1 + \frac{4}{1 + \frac{5}{\dots + \frac{1}{99 + \frac{1}{100}}}}}}} \end{array}$$

16. Onluq say sistemində verilmiş 444, 871, 1945, 2002, 4911 ədədlərini onaltılıq say sistemində ifadə edin.

17. Onluq say sistemində təsadüfi götürülmüş dörd rəqəmli natural ədədi ikilik, səkkizlik və onaltılıq say sistemlərində ekvivalentlərini tapın.

18. Təsadüfi qəbul edilmiş və 127-dən böyük olmayan 20 sayda ədədlərdən hər birinə ASCII kodlar cədvəlində latın hərfi qarşı durarsa, bu hərfdən əvvəlki hərfi, həmin hərfin özünü və bu hərfdən sonrakı hərfi

ekrana çıxarmalı. Əgər tapılan hərfdən əvvəlki və ya sonrakı simvol latın hərfi deyilsə, onun yerinə “#” simvolunu yazmalı.

19. Təsadüfi qəbul edilmiş və 127-dən böyük olmayan 50 sayda ədədlərdən hər birinə ASCII kodlar cədvəlində kiçik latın hərfi qarşı duranları seçin, onlara uyğun kiçik latın hərflərini böyük latın hərflərinə çevirin və ardıcıl düzün.

20. Təsadüfi qəbul edilmiş və 127-dən böyük olmayan 100 sayda ədədlərdən ayrılıqda neçəsinə ASCII kodlar cədvəlində «b», «h», «n» və «y» simvolları uyğun gəlir. Cavabı aşağıdakı kimi formalaşdırın:

«b» - ... eded, «h» - ... eded, «n» - ... eded, «y» - ... eded.

21. Standart ASCII kodları və onlara qarşı qoyulan simvolları istifadə edərək, böyük və kiçik latın hərflərini əlifba sırası ilə düzün: Aa Bb Cc Dd ... Yy Zz.

22. Hesablayın:

22.1.	$S = \sum_{i=1}^9 i \sum_{j=i+1}^{10} (j-i)^2$	22.6.	$S = \sum_{i=1}^{15} \prod_{j=i+1}^{16} \sin(j-i) $
22.2.	$S = \sum_{i=1}^{10} \sum_{j=1}^{10} \cos(i^2 - j)$	22.7.	$S = \sum_{i=1}^6 i \prod_{j=1}^6 e^{(-1)^j j + j}$
22.3.	$S = \prod_{i=1}^{10} \cos i \prod_{j=1}^{10} \frac{\sin j}{i+j}$	22.8.	$S = \sum_{i=1}^{15} \frac{1}{i} \sum_{j=1}^{10} (-1)^j \cos j$
22.4.	$S = \sum_{i=1}^{15} \frac{(-1)^i i \cos i^2 }{i+1} \prod_{j=1}^8 \frac{ i-j }{j+1}$	22.9.	$S = \prod_{i=1}^7 i \sum_{j=1}^8 \frac{i+j}{j}$
22.5.	$S = \sum \frac{\sin i^2}{1 + \cos^2 i} \sum_{j=1}^{10} \sin(i-j)^2$	22.10.	$S = \prod_{i=1}^8 \sum_{j=1}^9 \sin(i+j) $

6. VERİLƏNLƏRİN SƏTİR TIPLƏRİ

6.1. Sətir tipli verilənlərin təsviri

Turbo Pascal-da sözləri və cümlələri təşkil edən simvolları emal etmək üçün verilənlərin **STRING** – sətir tipindən istifadə olunur. Sətir tipli verilənlər iki apostrof işarəsi (') arasında simvollar ardıcılığı ilə göstərilir. Məsələn, 'Akademiya', 'Odlar_Yurdu', '712_saylı_qrup' və s. Sətir tipli dəyişənlərə müxtəlif qiymətlər mənsub etmək, nisbət əməliyyatlarından istifadə etməklə onları müqayisə etmək, onlarla daxiletmə və xaric etmə əməliyyatları aparmaq, dəyişənə daxil olan simvolların axtarışını təşkil etmək, bir neçə dəyişənin qiymətlərini bitişdirmək, onlarla bu kimi digər əməliyyatları yerinə yetirmək mümkündür.

İki sətirin müqayisəsi bu sətirlərə daxil olan və eyni mövqelərdə yerləşən simvolların uyğun ASCII kodları üzrə ardıcıl olaraq soldan-sağa aparılır. Müqayisə o vaxta qədər davam edir ki, sətirdə eyni mövqedə duran simvollar bir-birindən fərqli olsun. Kiçik koda malik simvolu özündə saxlayan sətir kiçik hesab edilir. Əgər iki sətirdən birinin uzunluğu digərindən kiçikdirsə və kiçik sətirin simvolları uzun sətirin uyğun mövqelərində duran simvolları ilə eynidirsə, onda uzun sətir böyük hesab edilir. Məsələn,

```
'Turbo_Pascal' > 'Turbo_Basic',  
'Qar' < 'Qaraca',  
'714' > '712' və s.
```

Boş sətir heç bir simvolu özündə saxlamır və aralarında heç bir simvol olmayan qoşa apostrof işarəsi ilə (") ifadə olunur. Sətir dəyişəni

```
var <dəyişənin adı>: string;
```

və ya

```
var <dəyişənin adı>: string [n];
```

kimi təsvir edilir. Sonuncu yazılışda n – sətir dəyişəni üçün ayrılmış baytların sayını göstərir, yəni sətir dəyişəni maksimum n sayda simvoldan ibarət ola bilər. Sətir dəyişəninin təsvirində n göstərilməyibsə, onda sətirin uzunluğu 255 simvol qəbul edilir. Dəyişənə n saydan çox simvollar ardıcılığı mənsub edilərsə, sağdan artıq simvollar atılır və dəyişən ilk n sayda simvolu qəbul edir. Məsələn,

```
var s: string [6];
```

```
...  
s:='Barselona';
```

{Nəticə:

'Barsel'}

Sətir tipli dəyişənə n sayda elementdən ibarət birölçülü massiv kimi də baxmaq olar və bu dəyişənin istənilən simvoluna dəyişənin adı və indeksdə simvolun sətirdəki mövqesi göstərilməklə müraciət etmək olar. Məsələn, s[2]='a', s[3]='r', s[5]='e' və s. Massivin 0 indeksli elementində sətirin uzunluğu yerləşdirilir: s[0]=6. Bu element ASCII kodlarla ifadə olunduğundan, onun qiymətinə yalnız ord funksiyası ilə baxmaq olar. Sonuncu funksiya ASCII-cədvəldə simvola uyğun ədədi tapmaq üçün istifadə olunur:

```
var    s: string [6];
      t: longint;
      ...
      s:='Barselona';      {Nəticə:      s='Barsel'}
      t:=ord(s[0]);        {Nəticə:      t=6}
```

Sətir tipli dəyişən əvvəlcə n uzunluqlu elan edilsə də, proqramın icrası zamanı bu uzunluğu massivin 0 indeksli elementində dəyişiklik aparmaqla artırmaq və ya azaltmaq olar:

```
s[0]:=chr(8);      {Nəticə:      s='Barselon'}
t:=ord(s[0]);      {Nəticə:      t=8}.
```

Turbo Pascalda bir neçə sətiri bir-biri ilə bitişdirmək mümkündür. Bu əməliyyat konkatenasiya əməliyyatı adlanır və əməliyyat "+" işarəsi ilə təsvir olunur. Bitişdirilmiş sətirlər hər hansı sətir dəyişəninə mənsub edilir:

```
s, s1, s2: string;
...
s:='Compares_'+'two_'+'strings';
      {Nəticə:      s='Compares_two_strings'}
s1:='Copies_one_string';
s2:='to_another';
s:=s1+'_' +s2;    {Nəticə:      s='Copies_one_string_to_another'}
```

6.2. Sətir tipli verilənlərin emalı

Sətir tipli verilənləri emal etmək üçün Turbo Pascal-da bir sıra standart funksiya və prosedurlardan istifadə olunur (cədvəl 6.1). Bu funksiyaları və prosedurları ayrılıqda nəzərdən keçirək.

- **Copy(s: string,<mövqe>,<uzunluq>: integer):string;**

Funksiya icra olunarkən sətir dəyişəninə göstərilən mövqesindən başlayaraq verilmiş uzunluqda fraqment ayrılır. Funksiyanın yazılışında 3 sayda parametr göstərməlidir: s - ayrılacaq fraqmenti özündə saxlayan sətir; <mövqe> - ayrılacaq fraqmentin ilk simvolunun sətirdəki mövqeyi,

<uzunluq> - ayrılacaq fraqmentdəki simvolların sayı. Sətirdən ayrılmış fraqment digər dəyişənə mənsub edilə bilər. Misal:

```
var metn: string; metn1,metn2: string[15];
```

```
...
```

```
metn:= 'Mirze_Elekber_Sabir_<Hophopnamenin>_muellifidir';
```

```
metn1:=copy(metn,15,5); {Nəticə: 'Sabir'}
```

```
metn2:=copy(metn,22,6); {Nəticə: 'Hophop'}
```

Cədvəl 6.1

Sətir tipli verilənlərlə işləyən funksiyalar və prosedurlar

Nö	Adı	Təyinatı
Funksiyalar		
1	Copy	Sətirdən fraqment ayrılır. Formatı: Copy(s: string, <mövqe>, <uzunluq>: integer): string;
2	Pos	Sətirdə fraqmentin axtarışı aparılır. Formatı: Pos(fraq, s: string): byte;
3	Length	Sətrin cari uzunluğunu təyin edir. Formatı: Length(s: string): byte;
4	Concat	Bir neçə sətri bitişdirir. Formatı: Concat(s ₁ , s ₂ , ..., s _n : string): string;
5	UpCase	Sətrə daxil olan kiçik hərfi böyük hərfə çevirir. Formatı: UpCase(s: string): string;
Prosedurlar		
6	Delete	Sətirdən fraqment silinir. Formatı: Delete(s: string, <mövqe>, <uzunluq>: integer);
7	Insert	Sətrə göstərilən mövqedən yeni sətir əlavə edilir. Formatı: Insert(fraq,s: string, <mövqe>, <uzunluq>: integer);
8	Str	Arqumentin ədədi qiymətini sətir ifadəsinə çevirir. Formatı: Str(x[:<uz>[:<kh>]]): real or integer, st: string);
9	Val	Arqumentin sətir ifadəsini ədədi qiymətə çevirir. Formatı: val(st: string, x: real or integer, kod: integer);

- Pos(fraq, s: string): byte;

Funksiya icra olunarkən, fraq sətir fraqmentinin s sətrinə daxil olması şərti yoxlanılır. s sətrində fraq fraqmentinə rast gəlinirsə, onda funksiyanın aldığı ədədi qiymət fraq fraqmentinin ilk simvolunun s sətrindəki mövqesiinə bərabər olur. Əks halda, funksiya 0 qiymətini qəbul edir. Misal:

```
var s: string; s1,s2,s3: string[6]; t1,t2,t3: byte;
```

```
...
```

```

s:='Yunanistan';
s1:='an'; t1:=pos(s1,s); {Nəticə: Movqe: 4}
s2:='and'; t2:=pos(s2,s); {Nəticə: Movqe: 0}
s3:='un'; t3:=pos(s3,s); {Nəticə: Movqe: 2}

```

Fraq fraqmentinin s sətirində bir neçə yerdə təkrar olunduğu hallarda belə, pos funksiyası həmişə ilk rast gəlinən fraqmentin birinci simvolunun mövəsini göstərən ədədi qiyməti alır. Məsələn, 'an' fraqmenti s sətirində iki dəfə rast gəlinir, lakin pos funksiyası təkrar icra olunsa belə, yenə də birinci rast gəlinən sətir fraqmentinin mövqesini qaytarır:

```

t1:=pos('an',s); {Nəticə: t1=4}
t2:=pos('an',s); {Nəticə: t2=4}

```

Fraqmentin s sətrinə daxil olması şərti yoxlanılarkən, kiçik və böyük yazılmış eyni hərflər, onların ASCII kodları eyni olmadığından, müxtəlif qəbul edilir. Məsələn, "a" və "A" hərfləri müqayisə zamanı müxtəlif simvollar kimi başa düşülür:

```

t1:=pos('an',s); {Nəticə: t1=4}
t2:=pos('An',s); {Nəticə: t2=0}

```

- **Length(s: string): byte;**

Funksiya sətir dəyişəninin cari (faktiki) uzunluğunu təyin edir. Dəyişənin təsvirində göstərilmiş maksimal uzunluq bu funksiya ilə tapıla bilməz. Məsələn:

```

var metn: string; t: byte;

```

```

...
metn:='Xezer_denizi'; t:=length(metn); {Nəticə: t=12}
metn:='Odlar_Yurdu_Universiteti'; t:=length(metn); {Nəticə: t=24}

```

Qeyd. Sətrin sonunda boşluq simvolları (" ") qoyulmuşsa, bu simvollar da sətirin uzunluğunda nəzərə alınır:

```

metn:='Odlar_Yurdu_Universiteti_'; t:=length(metn); {Nəticə: t=24}

```

- **Concat(s₁, s₂, ..., s_n): string;**

Funksiya icra olunarkən s₁, s₂, ..., s_n sətirləri ardıcıl olaraq bir-birinə bitişir. Bitişmədən əmələ gələn s sətiri sətir tipli olur. Məsələn:

```

var s: string; s1,s2,s3: string[10];

```

```

...
s1:='aero'; s2:='port'; s3:='plan';
s:=concat(s1,s2); {Nəticə: s='aeroport'}
s:=concat(s1,s3); {Nəticə: s='aeroplan'}

```

- **UpCase(s: string): string;**

Funksiya s sətrin yalnız hər hansı bir simvolu üçün tətbiq edilə bilər və icra olunarkən sətərə daxil olan kiçik hərfi böyük hərfə çevirir. Bu səbəbdən də, sətirin bütün kiçik hərfli simvollarını böyük hərf simvollarına

çevirmək üçün dövr təşkil etmək tələb olunur. Qeyd etmək lazımdır ki, rus və azərbaycan literləri üçün UpCase funksiyası yaramır. Misal,

```
var sk,sb: string; i: integer;
```

```
***
```

```
sk:='azerbaycan';
```

```
for i:=1 to length(sk) do sb[i]:=UpCase(sk[i]);
```

```
{Nəticə: sk='azerbaycan', sb='AZERBAYCAN'}
```

```
- Delete(s: string, <mövqe>, <uzunluq>: integer);
```

Prosedur icra olunarkən s sətrindən prosedurda göstərilən mövqedən başlayaraq göstərilən sayda simvollar ardıcılığı silinir. Prosedurda göstərilən mövqe sətrin uzunluğundan böyükdürsə, onda sətirdə heç bir silinmə əməliyyatı aparılmır. Prosedurda göstərilən silinməli olan simvolların sayı, göstərilən mövqedən başlayaraq sətrin sonunadək olan sətir simvollarının sayından böyükdürsə, onda həmin mövqedən başlayaraq sətrin sonunadək olan simvolların hamısı silinir. Misal:

```
var s: string;
```

```
***
```

```
s:='qartal'; {Nəticə: s='qartal'}
```

```
delete(s,9,1); {Nəticə: s='qartal'}
```

```
delete(s,4,1); {Nəticə: s='qaral'}
```

```
delete(s,5,1); {Nəticə: s='qara'}
```

```
delete(s,2,8); {Nəticə: s='q'}
```

```
- Insert(fraq, s: string, <mövqe>, <uzunluq>: integer);
```

Prosedur icra olunarkən, fraq fraqmenti s sətrinin prosedurda göstərilən mövqesindən sətrə əlavə edilir.

```
var s1,s2: string;
```

```
***
```

```
s1:='qar'; {Nəticə: s1='qar' }
```

```
s2:='ta'; {Nəticə: s2='ta' }
```

```
insert(s2, s1, 3); {Nəticə: s1='qatar'}
```

Əgər sətrə əlavə edildikdən sonra yeni alınmış s sətrinin uzunluğu ilkin təsvir edilmiş uzunluqdan böyükdürsə, onda artıq simvollar sağdan başlayaraq ləğv edilir. Misal:

```
var s3: string[5];
```

```
***
```

```
s3:='baca'; {Nəticə: s3='baca'}
```

```
insert('la', s3, 3); {Nəticə: s3='balac', sonuncu 'a' simvolu ləğv edilib}
```

```
- Str(x[:<uz>[:<kh>]]: real or integer, st: string);
```

Prosedur tam və ya həqiqi x ədədini simvollar sətrinə çevirmək üçün istifadə olunur. Prosedurun yazılışında ədədin sətirdəki <uz> uzunluğu və həqiqi ədədlər üçün <kh> kəsr hissəsinə uyğun sətirdəki mövqelərin sayı göstərilə bilər.

Uzunluqlar prosedura aşkar şəkildə göstərilməyibsə, onda susmaya görə, sətirin uzunluğu tam ədədin cari uzunluğuna bərabər götürülür, həqiqi ədədlər isə simvol sətirdə eksponensial formada təsvir olunurlar.

Tam x ədədi üçün prosedura uzunluq göstərilərsə və ədəddəki rəqəmlərin sayı bu uzunluqdan kiçikdirsə, onda sətirdə soldan artıq qalan mövqelərə boşluq simvolları əlavə edilir. Əgər tərsinə, ədədin rəqəmlərinin sayı sətir üçün ayrılmış simvolların sayından böyükdürsə, onda avtomatik olaraq sətir soldan artıq mövqelərin sayı qədər genişləndirilir. Həqiqi ədədlər üçün də, sətirdə ədədin tam və kəsr hissələrini ayıran nöqtə simvolunun qoyuluşu nəzərə alınmaqla, yuxarıda deyilənlər doğrudur. Sətirdə kəsr hissə üçün ayrılmış mövqelərin sayı ədədin kəsr hissəsindəki rəqəmlərin sayından azdırsa, ədəd əvvəlcə yuvarlaqlaşdırılır, sonra isə sətir formalaşır. Əks halda, sətirdə kəsr hissə üçün artıq mövqelər sağdan qiymətsiz sıfırlarla doldurulur.

Sətirin uzunluğu dəyişənin təsviri zamanı aşkar şəkildə verilibsə, kəsr hissə prosedura verilmiş mövqelərə görə əvvəlcə yuvarlaqlaşdırılır, sonra isə ədədin tam hissəsinin sətirdə tamamilə yerləşdirilməsi şərti ilə, kəsr hissə sağdan kəsilir. Misal:

```
var s: string; s1: string[6]; x: integer; y: real;
```

```
...
```

```
x:=76; y:=295.765844;
```

str(x,s);	{Nəticə:	s='76'
str(x:3,s);	{Nəticə:	s='_76'
str(x:5,s);	{Nəticə:	s='__76'
str(y,s);	{Nəticə:	s='2.9576584400E+02'
str(y:7:2,s);	{Nəticə:	s='_295.77'
str(y:7:3,s);	{Nəticə:	s='295.766'
str(y:3:1,s);	{Nəticə:	s='295.8'
str(y:7:2,s1);	{Nəticə:	s1='295.77'
str(y:8:5,s1);	{Nəticə:	s1='295.76'

- Val(st: string, x: real or integer, kod: integer);

Prosedur st sətirdəki simvolları x dəyişənin tipindən asılı olaraq tam və ya həqiqi ədədə çevirmək üçün istifadə olunur. Çevirmə doğru başa çatıbsa, nəticə x dəyişəninə mənsub edilir və kod dəyişəni 0 qiymətini alır. Əks halda, st sətirdə səhv simvollarla rast gəlinibsə, ilk rast

gəlinən səhv simvolun mövqesinin qiyməti kod dəyişəninə mənsub edilir, x dəyişənin qiyməti isə 0 olur. st sətirdə ədədin sol hissəsində boşluq simvollarından istifadə oluna bilər. Ədədin sağdakı boşluq simvollarını isə prosedur səhv kimi qəbul edir və bu halda ədəddən sonrakı ilk boşluq simvolunun mövqesi kod dəyişəninə mənsub edilir, x dəyişənin qiyməti isə 0 olur. Misal:

```
var s: string; x, kod: integer; y: real;
```

```
...
s:='65'; val(s,x,kod); {Nəticə: x=65 kod=0}
s:='72.36'; val(s,y,kod); {Nəticə: y=72.36 kod=0}
s:='__7.9'; val(s,y,kod); {Nəticə: y=_7.90 kod=0}
s:='7.9__'; val(s,y,kod); {Nəticə: y=_0.00 kod=4}
s:='60$__'; val(s,x,kod); {Nəticə: x=0 kod=3}
```

Çalışma 6.1 (Faiz). Cümlədə istifadə olunan hərflərin cümlənin uzunluğuna nəzərən faizlərini tapmalı.

Həlli. Məsələnin həlli üçün ilk növbədə cümlədə işlənən hərflər müəyyənləşdirilməlidir. Nəzərə almaq lazımdır ki, cümlədə durğu işarələri və hərflərin təkrarlanmaları ola bilər.

Məqsədə nail olmaq üçün proqramda t sətir dəyişəni götürülmüş və ona boş sətir mənsub edilmişdir (t:= ""). Sonra yaddaşa daxil edilən cümlənin uzunluğuna (m=length(s)) görə dövr təşkil edilmişdir. Dövrün daxilində cümlənin hər bir i (i ≤ m) saylı simvolu ayrılır (t1:=copy(s,i,1)), onun durğu işarəsi olmaması (pos(t1,d)=0) və proqramın icrası zamanı ona yeni hərflərin bitişdirilməsi ilə genişlənən t sətrinə daxil olmaması (pos(t1,t)=0) şərtləri yoxlanılır. Hər iki şərt doğru qiymət alarsa, yeni ayrılmış t1 simvolu hərfdirsə və bu hərf t sətir dəyişəninə yoxdursa, onda t1 simvolu t sətrinə sağdan bitişir (concat(t,t1)). Dövrün sonunda t sətir dəyişəni formalaşır və o, təkrarlanmama şərti nəzərə alınmaqla s sətir dəyişəninə cümlənin bütün hərflərindən ibarət olur.

Növbəti mərhələdə t sətirinin n uzunluğuna (n=length(t)) görə dövr təşkil olunur. Dövrün daxilində t sətrinə daxil olan hər bir hərfin s sətirində neçə dəfə rast gəlinməsi hesablanır. t sətirinin hər bir hərfi



(copy(t,i,1)), bu hərfin s sətirindəki təkrarlanmaları (k) və cümlənin uzunluğuna nəzərən rastolma faizi (fz) çap edilir. Dəyişənlərin qiymətlərini 5 sayda sütunda çap etmək üçün 1 dəyişəndən saygac kimi istifadə edilib. Bu dəyişənin qiyməti hər daxili dövrə 5-ə bölünür, bölmədən alınan qalıq 0 olarsa, yəni 5-ə tam bölmə baş verərsə, onda çap üçün yeni sətərə keçid baş verir.

Məsələnin həlli üçün tərtib edilmiş proqramın listingi və proqramın icrasından alınan nəticələr aşağıda göstərilmişdir.

```

Program Faiz;
Uses WinCrt;
const d='..!''?;';
var s, t: string;
    t1, t2: string[1];
    i, j, l, m, n, k: integer;
    fz: real;
begin
s:='XALQIMIZIN ANA KITABI "DEDE QORQU'D DASTANIDIR.';
t:=''; m:=length(s);
for i:=1 to m do begin
    t1:=copy(s,i,1);
    if (pos(t1,t)=0) and (pos(t1,d)=0) then t:=concat(t,t1);
    end;
n:=length(t); l:=0;
for i:=1 to n do begin    k:=0;
    for j:=1 to m do
        if copy(s,j,1)=copy(t,i,1) then k:=k+1;
        fz:=k*100/m;
        write(' ',copy(t,i,1),'-',k,'-',fz:4:1,'%');
        l:=l+1;
        if (l mod 5)=0 then writeln;
    end;
end.

```

Cavab:

X-1- 2.1%	A-6-12.8%	L-1- 2.1%	Q-3- 6.4%	I-7-14.9%
M-1-2.1%	Z-1- 2.1%	N-3- 6.4%	K-1-2.1%	T-2- 4.3%
B-1-2.1%	D-5-10.6%	E-2- 4.3%	O-1- 2.1%	R-2- 4.3%
U-1-2.1%	S-1- 2.1%			

6.3. Laboratoriya işi № 8. Verilənlərin sətir tipləri

1-20 sayılı tapşırıqlarda S1, S2 və S3 sətir tipli dəyişənləri verilmişdir və onlar aşağıdakı qiymətləri alırlar:

S1 = 'Sherefñise xanim.'

S2 = 'Fireng qizlari bashlarina orten tesekler mene lazim deyil.'

S3 = 'Al, Parijde onların başlarına ort ki, Qarabaqdan onların havasına yellənib ucursan.'

1. S1, S2 və S3 sətirlərindən hər birinin uzunluğunu tap. Bu sətirlərdən hansı ən uzun qiymətə malikdir?

2. S1 sətirindəki kiçik latın hərflərini böyük latın hərfləri ilə əvəz et. S1 və S2 sətirlərinin sonuna «aralıq» simvolu əlavə et. S1, S2, S3 sətirlərini bitişdirərək yeni sətir formalaşdır.

3. S3 sətirindəki bütün «a» simvollarını «A» simvolu ilə əvəz et və yenidən sətiri əvvəlki vəziyyətinə qaytar. Qaytarma zamanı birinci simvulun «A» olmasını nəzərə al.

4. S1, S2 və S3 sətirlərinin hər birində neçə hərf simvolu vardır və bu saylar hər bir uyğun sətirdəki hərf simvolların neçə faizini təşkil edir?

5. S3 sətirindəki sözlərin sayını müəyyənləşdir və onları, aralarında «vergül» və aralıq ssimvolları olmaqla, bir-birinin ardınca ekrana çıxar.

6. S3 sətirindəki ən uzun sözü müəyyən et. Bu söz neçə simvoldan təşkil olunub?

7. S3 sətirindəki ilk rast gələn «onların» sözünü «Fireng qızların» sözü ilə əvəz et.

8. S3 sətirindəki «Qarabaqdan» sözündən sonra «Parije» sözünü əlavə et.

9. S2 sətirində ən çox işlənən hərfi təyin et. Bu hərf sətirdə neçə dəfə işlənmişdir?

10. S2 sətirində istifadə olunmayan hərfləri siyahı şəklində göstər.

11. S2 sətirində birinci sözlə sonuncu sözün yerlərini dəyişin.

12. S2 sətirində birinci sözlə ikinci sözün, üçüncü sözlə dördüncü sözün və s. qaydada yerlərini dəyişin. Cümlədə sözlərin sayı təkdirsə, sonuncu sözü öz yerində saxlayın. Durğu işarələrini nəzərə almayın və onlar öz yerlərində saxlanılsın.

13. S3 sətirində sətirində «o» hərfi ilə başlayan sözləri tapın.

14. S3 sətirindəki ikinci sözə daxil olan bütün hərflərin cümlənin digər sözlərində rast gəlib-gəlməməsini müəyyənləşdirin və bu hərflərdən hər birinin digər sözlərdə neçə dəfə istifadə olunduğunu təyin edin.

15. S3 sətində ardıcıl iki eyni hərfin olmasını müəyyənləşdirin. Belə qoşa eyni hərfi olan söz tapılsa, onu ekrana çıxarın.

16. S3 sətində «o» hərfi ilə başlayan sözləri silin və silinmiş sözlərin sayını təyin edin.

17. S3 sətində böyük latın hərflərini uyğun kiçik hərflərlə, kiçik hərfləri isə böyük hərflərlə əvəz edin. Dürğu işarələrini olduğu kimi saxlayın.

18. S2 sətində saitlərin sayı ən çox olan sözü tapın.

19. S2 sətində samitlərin sayı ən az olan sözü tapın.

20. S3 sətində uzunluğu ən uzun olan sözlə uzunluğu ən qısa olan sözün yerlərini dəyişin. Belə sözlərin sayı bir neçədirsə, cümlədə ilk rast gələnı götürün.

21. Tərkibində ardıcıl bir neçə «aralıq» simvolu olan S sətindən artıq «aralıq» simvollarını silin. Silinmiş simvolların sayını göstərin.

22. $A1 = '5 * 77 = ?'$, $A2 = '75 - 43 = ?'$, $A3 = '972 / 3 = ?'$ və $A4 = '4 + 38 = ?'$ sətir tipli dəyişənləri verilmişdir. Hesablamaları aparıb, hər bir dəyişəndə «?» simvolunun yerinə hesablamanın nəticəsini yazın.

7. MASSİVLƏR

7.1. Massivlərin təsviri və emalı

Massivlər strukturlaşdırılmış tip verilənlərinin təsvir formalarından biridir. **Massiv** – müəyyən ardıcılıqla nömrələnməyi bir sıra eyni tipli verilənlərin bir adla adlandırılan təsvir formasıdır. Məsələn, bir ildə ali məktəbin eyni ixtisasına daxil olan tələbələrin bir qismini bir tədris qrupunu təşkil edir və qrup nömrəsi ilə adlandırılır (6331, 6332 və s.), və ya kinoteatrın sarı divarlı "Sarı" zalında oturmaqlar sıra nömrələri boyunca düzülür və eyni vaxtda eyni filmə baxmaq istəyən şəxslər bu kreslolar üzrə yerləşdirilir.

Massiv elementinin nömrəsi **indeks** adlanır. İndekslərin sayı bir, və ya bir neçə ola bilər. Bir indekslə təsvir olunan massiv **birölçülü**, iki indekslə təsvir olunan massivlər **ikiölçülü** və s. adlanırlar. Ali riyaziyyatda birölçülü massivlərə **vektor**, ikiölçülü massivlərə isə **matris** anlayışları uyğun gəlir. Beləliklə, massivin hər bir elementinə massivin ümumi qəbul edilmiş adını və elementin massivdəki indeksini göstərərək müraciət etmək olar.

Məsələn, tələbələrin qrup jurnalı qrup nömrəsi ilə adlandırılır, jurnalın daxilindəki siyahıda isə hər bir tələbənin adına bir sıra nömrəsi qarşı qoyulur. Bu misalda tələbələrin adlarından ibarət bir sıra eyni tipli verilənlər birölçülü massiv kimi təsvir olunmuşdur. Massivdəki verilənlərin emalı zamanı hər bir tələbənin adına onun qrup və siyahıdakı sıra nömrələri ilə müraciət etmək olar. Kino biletlərinin üzərində qrupun adı ilə yanaşı, oturmaqların sıra və yer nömrələri də göstərilir. Verilənlərin belə təsvirində ikiölçülü massivdən istifadə olunmuşdur və massivin hər bir elementi iki indekslə – oturmaqların zaldakı yerini təyin edən sıra və yer nömrəsi ilə təyin olunmuşdur. Bilet alan şəxs zalda öz biletinə uyğun yeri çox asanlıqla tapa bilər.

Massiv **Array** xidməti sözü vasitəsilə təsvir edilir:

<Tipin adı> : Array [<İndeksələr>] of <Tip>;

Burada, <Tipin adı> - identifikatordur, <İndeksələr> - **of** <Tip>; **LongInt** və baza tipi **LongInt**-dən fərqli olan ixtiyari nizamlı tip ilə təsvir oluna bilər, <Tip> isə Turbo Pascal-ın ixtiyari tipidir.

Massivlərin təsvirində indekslər üçün **ada** diapazon tipdən istifadə edilir, indeksin ala biləcəyi aşağı və yuxarı **sə**dlərini göstərir. Məsələn,

x: Array [1..20] of Real;

15. S3 sətrində ardıcıl iki eyni hərfin olmasını müəyyənləşdirin. Belə qoşa eyni hərfi olan söz tapılsa, onu ekrana çıxarın.

16. S3 sətrində «o» hərfi ilə başlayan sözləri silin və silinmiş sözlərin sayını təyin edin.

17. S3 sətrində böyük latın hərflərini uyğun kiçik hərflərlə, kiçik hərfləri isə böyük hərflərlə əvəz edin. Dürğu işarələrini olduğu kimi saxlayın.

18. S2 sətrində saitlərin sayı ən çox olan sözü tapın.

19. S2 sətrində samitlərin sayı ən az olan sözü tapın.

20. S3 sətrində uzunluğu ən uzun olan sözlə uzunluğu ən qısa olan sözün yerlərini dəyişin. Belə sözlərin sayı bir neçədirsə, cümlədə ilk rast gələnə götürün.

21. Tərkibində ardıcıl bir neçə «aralıq» simvolu olan S sətrindən artıq «aralıq» simvollarını silin. Silinmiş simvolların sayını göstərin.

22. $A1=5*77=?$, $A2=75-43=?$, $A3=972/3=?$ və $A4=4+38=?$ sətir tipli dəyişənləri verilmişdir. Hesablamaları aparıb, hər bir dəyişəndə «?» simvolunun yerinə hesablamanın nəticəsini yazın.

7. MASSİVLƏR

7.1. Massivlərin təsviri və emalı

Massivlər strukturlaşdırılmış tip verilənlərin təsvir formalarından biridir. **Massiv** – müəyyən ardıcılıqla nömrələnmiş bir sıra eyni tipli verilənlərin bir adla adlandırılan təsvir formasıdır. Məsələn, bir ildə ali məktəbin eyni ixtisasına daxil olan tələbələrin bir qismi bir tədris qrupunu təşkil edir və qrup nömrəsi ilə adlandırılır (631, 633 və s.), və ya kinoteatrın sarı divarlı "Sarı" zalında oturmaqlar sıra və sütunlar boyunca düzülür və eyni vaxtda eyni filmə baxmaq istəyən şəxslər bu kreslolar üzrə yerləşdirilir.

Massiv elementinin nömrəsi **indeks** adlanır. İndekslərin sayı bir, və ya bir neçə ola bilər. Bir indekslə təsvir olunan massivlər **birölçülü**, iki indekslə təsvir olunan massivlər **ikiölçülü** və s. adlanırlar. Ali riyaziyyatda birölçülü massivlərə **vektor**, ikiölçülü massivlərə isə **matris** anlayışları uyğun gəlir. Beləliklə, massivin hər bir elementinə massivin ümumi qəbul edilmiş adını və elementin massivdəki indeksini göstərməklə müraciət etmək olar.

Məsələn, tələbələrin qrup jurnalı qrup nömrəsi ilə adlandırılır, jurnalın daxilindəki siyahıda isə hər bir tələbənin adına bir sıra nömrəsi qarşı qoyulur. Bu misalda tələbələrin adlarından ibarət sətir tipli verilənlər birölçülü massiv kimi təsvir olunmuşdur. Massivdəki verilənlərin emalı zamanı hər bir tələbənin adına onun qrup və siyahıdakı say nömrələri ilə müraciət etmək olar. Kino biletlərinin üzərində zalın adı ilə yanaşı, oturmaqların sıra və yer nömrələri də göstərilir. Verilənlərin belə təsvirində ikiölçülü massivdən istifadə olunmuşdur və massivin hər bir elementi iki indekslə – oturmaqın zaldakı yerini təyin edən sıra və yer nömrəsi ilə təyin olunmuşdur. Bilet alan şəxs zalda öz biletinə uyğun yeri çox asanlıqla tapa bilər.

Massiv **Array** xidməti sözü vasitəsilə təsvir olunur:

<Tipin adı> : Array [<İndeksələr>] of <Tip>;

Burada, <Tipin adı> - identifikatordur, <İndeksələr> - LongInt və baza tipi LongInt-dən fərqli olan ixtiyari nizamlı tip ilə təsvir oluna bilər, <Tip> isə Turbo Pascal-ın ixtiyari tipidir.

Massivlərin təsvirində indekslər üçün adətən diapazon tipdən istifadə edilir, indeksin ala biləcəyi aşağı və yuxarı sərhədlərini göstərir. Məsələn,

x: Array [1..20] of Real;

15. S3 sətirində ardıcıl iki eyni hərfin olmasını müəyyənləşdirin. Belə qoşa eyni hərfi olan söz tapılsa, onu ekrana çıxarın.

16. S3 sətirində «o» hərfi ilə başlayan sözləri silin və silinmiş sözlərin sayını təyin edin.

17. S3 sətirində böyük latın hərflərini uyğun kiçik hərflərlə, kiçik hərfləri isə böyük hərflərlə əvəz edin. Dürğu işarələrini olduğu kimi saxlayın.

18. S2 sətirində saitlərin sayı ən çox olan sözü tapın.

19. S2 sətirində samitlərin sayı ən az olan sözü tapın.

20. S3 sətirində uzunluğu ən uzun olan sözlə uzunluğu ən qısa olan sözün yerlərini dəyişin. Belə sözlərin sayı bir neçədirsə, cümlədə ilk rast gələnə götürün.

21. Tərkibində ardıcıl bir neçə «aralıq» simvolu olan S sətirindən artıq «aralıq» simvollarını silin. Silinmiş simvolların sayını göstərin.

22. $A1=5*77=?$, $A2=75-43=?$, $A3=972/3=?$ və $A4=4+38=?$ sətir tipli dəyişənləri verilmişdir. Hesablamaları aparıb, hər bir dəyişəndə «?» simvolunun yerinə hesablamının nəticəsini yazın.

7. MASSİVLƏR

7.1. Massivlərin təsviri və emalı

Massivlər strukturlaşdırılmış tip verilənlərin təsvir formalarından biridir. **Massiv** – müəyyən ardıcılıqla nömrələnmiş bir sıra eyni tipli verilənlərin bir adla adlandırılan təsvir formasıdır. Məsələn, bir ildə ali məktəbin eyni ixtisasına daxil olan tələbələrin bir qismi bir tədris qrupunu təşkil edir və qrup nömrəsi ilə adlandırılır (631, 633 və s.), və ya kinoteatrın sarı divarlı "Sarı" zalında oturmaqlar sıra və sütunlar boyunca düzülür və eyni vaxtda eyni filmə baxmaq istəyən şəxslər bu kreslolar üzrə yerləşdirilir.

Massiv elementinin nömrəsi **indeks** adlanır. İndekslərin sayı bir, və ya bir neçə ola bilər. Bir indekslə təsvir olunan massivlər **birölçülü**, iki indekslə təsvir olunan massivlər **ikiölçülü** və s. adlanırlar. Ali riyaziyyatda birölçülü massivlərə **vektor**, ikiölçülü massivlərə isə **matris** anlayışları uyğun gəlir. Beləliklə, massivin hər bir elementinə massivin ümumi qəbul edilmiş adını və elementin massivdəki indeksini göstərməklə müraciət etmək olar.

Məsələn, tələbələrin qrup jurnalı qrup nömrəsi ilə adlandırılır, jurnalın daxilindəki siyahıda isə hər bir tələbənin adına bir sıra nömrəsi qarşı qoyulur. Bu misalda tələbələrin adlarından ibarət sətir tipli verilənlər birölçülü massiv kimi təsvir olunmuşdur. Massivdəki verilənlərin emalı zamanı hər bir tələbənin adına onun qrup və siyahıdakı say nömrələri ilə müraciət etmək olar. Kino biletlərinin üzərində zalın adı ilə yanaşı, oturmaqların sıra və yer nömrələri də göstərilir. Verilənlərin belə təsvirində ikiölçülü massivdən istifadə olunmuşdur və massivin hər bir elementi iki indekslə – oturmağın zaldakı yerini təyin edən sıra və yer nömrəsi ilə təyin olunmuşdur. Bilet alan şəxs zalda öz biletinə uyğun yeri çox asanlıqla tapa bilər.

Massiv **Array** xidməti sözü vasitəsilə təsvir olunur:

<Tipin adı> : Array [<İndeksələr>] of <Tip>;

Burada, <Tipin adı> - identifikatordur, <İndeksələr> - LongInt və baza tipi LongInt-dən fərqli olan ixtiyari nizamlı tip ilə təsvir oluna bilər, <Tip> isə Turbo Pascal-ın ixtiyari tipidir.

Massivlərin təsvirində indekslər üçün adətən diapazon tipdən istifadə edilir, indeksin ala biləcəyi aşağı və yuxarı sərhədlərini göstərir. Məsələn,

x: Array [1..20] of Real;

15. S3 sətrində ardıcıl iki eyni hərfin olmasını müəyyənləşdirin. Belə qoşa eyni hərfi olan söz tapılsa, onu ekrana çıxarın.

16. S3 sətrində «o» hərfi ilə başlayan sözləri silin və silinmiş sözlərin sayını təyin edin.

17. S3 sətrində böyük latın hərflərini uyğun kiçik hərflərlə, kiçik hərfləri isə böyük hərflərlə əvəz edin. Dürğu işarələrini olduğu kimi saxlayın.

18. S2 sətrində saitlərin sayı ən çox olan sözü tapın.

19. S2 sətrində samitlərin sayı ən az olan sözü tapın.

20. S3 sətrində uzunluğu ən uzun olan sözlə uzunluğu ən qısa olan sözün yerlərini dəyişin. Belə sözlərin sayı bir neçədirsə, cümlədə ilk rast gələnə götürün.

21. Tərkibində ardıcıl bir neçə «aralıq» simvolu olan S sətrindən artıq «aralıq» simvollarını silin. Silinmiş simvolların sayını göstərin.

22. $A1=5*77=?$, $A2=75-43=?$, $A3=972/3=?$ və $A4=4+38=?$ sətir tipli dəyişənləri verilmişdir. Hesablamaları aparıb, hər bir dəyişəndə «?» simvolunun yerinə hesablamanın nəticəsini yazın.

7. MASSİVLƏR

7.1. Massivlərin təsviri və emalı

Massivlər strukturlaşdırılmış tip verilənlərin təsvir formalarından biridir. **Massiv** – müəyyən ardıcılıqla nömrələnmiş bir sıra eyni tipli verilənlərin bir adla adlandırılan təsvir formasıdır. Məsələn, bir ildə ali məktəbin eyni ixtisasına daxil olan tələbələrin bir qismi bir tədris qrupunu təşkil edir və qrup nömrəsi ilə adlandırılır (631, 633 və s.), və ya kinoteatrın sarı divarlı "Sarı" zalında oturmaqlar sıra və sütunlar boyunca düzülür və eyni vaxtda eyni filmə baxmaq istəyən şəxslər bu kreslolar üzrə yerləşdirilir.

Massiv elementinin nömrəsi **indeks** adlanır. İndekslərin sayı bir, və ya bir neçə ola bilər. Bir indekslə təsvir olunan massivlər **birölçülü**, iki indekslə təsvir olunan massivlər **ikiölçülü** və s. adlanırlar. Ali riyaziyyatda birölçülü massivlərə **vektor**, ikiölçülü massivlərə isə **matris** anlayışları uyğun gəlir. Beləliklə, massivin hər bir elementinə massivin ümumi qəbul edilmiş adını və elementin massivdəki indeksini göstərməklə müraciət etmək olar.

Məsələn, tələbələrin qrup jurnalı qrup nömrəsi ilə adlandırılır, jurnalın daxilindəki siyahıda isə hər bir tələbənin adına bir sıra nömrəsi qarşı qoyulur. Bu misalda tələbələrin adlarından ibarət sətir tipli verilənlər birölçülü massiv kimi təsvir olunmuşdur. Massivdəki verilənlərin emalı zamanı hər bir tələbənin adına onun qrup və siyahıdakı say nömrələri ilə müraciət etmək olar. Kino biletlərinin üzərində zalın adı ilə yanaşı, oturmaqların sıra və yer nömrələri də göstərilir. Verilənlərin belə təsvirində ikiölçülü massivdən istifadə olunmuşdur və massivin hər bir elementi iki indekslə oturmağın zaldakı yerini təyin edən sıra və yer nömrəsi ilə təyin olunmuşdur. Bilet alan şəxs zalda öz biletinə uyğun yeri çox asanlıqla tapa bilər.

Massiv **Array** xidməti sözü vasitəsilə təsvir olunur:

<Tipin adı> : Array [<İndeksələr>] of <Tip>;

Burada, <Tipin adı> - identifikatordur, <İndeksələr> - LongInt və baza tipi LongInt-dən fərqli olan ixtiyari nizamlı tip ilə təsvir oluna bilər, <Tip> isə Turbo Pascal-ın ixtiyari tipidir.

Massivlərin təsvirində indekslər üçün adətən diapazon tipdən istifadə edilir, indeksin ala biləcəyi aşağı və yuxarı sərhədlərini göstərir. Məsələn,

x: Array [1..20] of Real;

15. S3 sətirində ardıcıl iki eyni hərfin olmasını müəyyənləşdirin. Belə qoşa eyni hərfi olan söz tapılsa, onu ekrana çıxarın.

16. S3 sətirində «o» hərfi ilə başlayan sözləri silin və silinmiş sözlərin sayını təyin edin.

17. S3 sətirində böyük latın hərflərini uyğun kiçik hərflərlə, kiçik hərfləri isə böyük hərflərlə əvəz edin. Dürğu işarələrini olduğu kimi saxlayın.

18. S2 sətirində saitlərin sayı ən çox olan sözü tapın.

19. S2 sətirində samitlərin sayı ən az olan sözü tapın.

20. S3 sətirində uzunluğu ən uzun olan sözle uzunluğu ən qısa olan sözün yerlərini dəyişin. Belə sözlərin sayı bir neçədirsə, cümlədə ilk rast gələnı götürün.

21. Tərkibində ardıcıl bir neçə «aralıq» simvolu olan S sətirindən artıq «aralıq» simvollarını silin. Silinmiş simvolların sayını göstərin.

22. $A1=5*77=?$, $A2=75-43=?$, $A3=972/3=?$ və $A4=4+38=?$ sətir tipli dəyişənləri verilmişdir. Hesablamaları aparıb, hər bir dəyişəndə «?» simvolunun yerinə hesablamının nəticəsini yazın.

7. MASSİVLƏR

7.1. Massivlərin təsviri və emalı

Massivlər strukturlaşdırılmış tip verilənlərin təsvir formalarından biridir. **Massiv** – müəyyən ardıcılıqla nömrələnmiş bir sıra eyni tipli verilənlərin bir adla adlandırılan təsvir formasıdır. Məsələn, bir ildə ali məktəbin eyni ixtisasına daxil olan tələbələrin bir qismi bir tədris qrupunu təşkil edir və qrup nömrəsi ilə adlandırılır (631, 633 və s.), və ya kinoteatrın sarı divarlı "Sarı" zalında oturmaqlar sıra və sütunlar boyunca düzülür və eyni vaxtda eyni filmə baxmaq istəyən şəxslər bu kreslolar üzrə yerləşdirilir.

Massiv elementinin nömrəsi **indeks** adlanır. İndekslərin sayı bir, və ya bir neçə ola bilər. Bir indekslə təsvir olunan massivlər **birölçülü**, iki indekslə təsvir olunan massivlər **ikiölçülü** və s. adlanırlar. Ali riyaziyyatda birölçülü massivlərə **vektor**, ikiölçülü massivlərə isə **matris** anlayışları uyğun gəlir. Beləliklə, massivin hər bir elementinə massivin ümumi qəbul edilmiş adını və elementin massivdəki indeksini göstərməklə müraciət etmək olar.

Məsələn, tələbələrin qrup jurnalı qrup nömrəsi ilə adlandırılır, jurnalın daxilindəki siyahıda isə hər bir tələbənin adına bir sıra nömrəsi qarşı qoyulur. Bu misalda tələbələrin adlarından ibarət sətir tipli verilənlər birölçülü massiv kimi təsvir olunmuşdur. Massivdəki verilənlərin emalı zamanı hər bir tələbənin adına onun qrup və siyahıdakı say nömrələri ilə müraciət etmək olar. Kino biletlərinin üzərində zalın adı ilə yanaşı, oturmaqların sıra və yer nömrələri də göstərilir. Verilənlərin belə təsvirində ikiölçülü massivdən istifadə olunmuşdur və massivin hər bir elementi iki indekslə oturmağın zaldakı yerini təyin edən sıra və yer nömrəsi ilə təyin olunmuşdur. Bilet alan şəxs zalda öz biletinə uyğun yeri çox asanlıqla tapa bilər.

Massiv **Array** xidməti sözü vasitəsilə təsvir olunur:

<Tipin adı> : Array [<İndeksələr>] of <Tip>;

Burada, <Tipin adı> - identifikatordur, <İndeksələr> - LongInt və baza tipi LongInt-dən fərqli olan ixtiyari nizamlı tip ilə təsvir oluna bilər, <Tip> isə Turbo Pascal-ın ixtiyari tipidir.

Massivlərin təsvirində indekslər üçün adətən diapazon tipdən istifadə edilir, indeksin ala biləcəyi aşağı və yuxarı sərhədlərini göstərir. Məsələn,

x: Array [1..20] of Real;

15. S3 sətrində ardıcıl iki eyni hərfin olmasını müəyyənləşdirin. Belə qoşa eyni hərfi olan söz tapılsa, onu ekrana çıxarın.

16. S3 sətrində «o» hərfi ilə başlayan sözləri silin və silinmiş sözlərin sayını təyin edin.

17. S3 sətrində böyük latın hərflərini uyğun kiçik hərflərlə, kiçik hərfləri isə böyük hərflərlə əvəz edin. Dürğu işarələrini olduğu kimi saxlayın.

18. S2 sətrində saitlərin sayı ən çox olan sözü tapın.

19. S2 sətrində samitlərin sayı ən az olan sözü tapın.

20. S3 sətrində uzunluğu ən uzun olan sözlə uzunluğu ən qısa olan sözün yerlərini dəyişin. Belə sözlərin sayı bir neçədirsə, cümlədə ilk rast gələnı götürün.

21. Tərkibində ardıcıl bir neçə «aralıq» simvolu olan S sətrindən artıq «aralıq» simvollarını silin. Silinmiş simvolların sayını göstərin.

22. $A1=5*77=?$, $A2=75-43=?$, $A3=972/3=?$ və $A4=4+38=?$ sətir tipli dəyişənləri verilmişdir. Hesablamaları aparıb, hər bir dəyişəndə «?» simvolunun yerinə hesablamının nəticəsini yazın.

7. MASSİVLƏR

7.1. Massivlərin təsviri və emalı

Massivlər strukturlaşdırılmış tip verilənlərin təsvir formalarından biridir. **Massiv** – müəyyən ardıcılıqla nömrələnmiş bir sıra eyni tipli verilənlərin bir adla adlandırılan təsvir formasıdır. Məsələn, bir ildə ali məktəbin eyni ixtisasına daxil olan tələbələrin bir qismi bir tədris qrupunu təşkil edir və qrup nömrəsi ilə adlandırılır (631, 633 və s.), və ya kinoteatrın sarı divarlı "Sarı" zalında oturaqlar sıra və sütunlar boyunca düzülür və eyni vaxtda eyni filmə baxmaq istəyən şəxslər bu kreslolar üzrə yerləşdirilir.

Massiv elementinin nömrəsi **indeks** adlanır. İndekslərin sayı bir, və ya bir neçə ola bilər. Bir indekslə təsvir olunan massivlər **birölçülü**, iki indekslə təsvir olunan massivlər **ikiölçülü** və s. adlanırlar. Ali riyaziyyatda birölçülü massivlərə **vektor**, ikiölçülü massivlərə isə **matris** anlayışları uyğun gəlir. Beləliklə, massivin hər bir elementinə massivin ümumi qəbul edilmiş adını və elementin massivdəki indeksini göstərməklə müraciət etmək olar.

Məsələn, tələbələrin qrup jurnalı qrup nömrəsi ilə adlandırılır, jurnalın daxilindəki siyahıda isə hər bir tələbənin adına bir sıra nömrəsi qarşı qoyulur. Bu misalda tələbələrin adlarından ibarət sətir tipli verilənlər birölçülü massiv kimi təsvir olunmuşdur. Massivdəki verilənlərin emalı zamanı hər bir tələbənin adına onun qrup və siyahıdakı say nömrələri ilə müraciət etmək olar. Kino biletlərinin üzərində zalın adı ilə yanaşı, oturaqların sıra və yer nömrələri də göstərilir. Verilənlərin belə təsvirində ikiölçülü massivdən istifadə olunmuşdur və massivin hər bir elementi iki indekslə oturağın zaldakı yerini təyin edən sıra və yer nömrəsi ilə təyin olunmuşdur. Bilet alan şəxs zalda öz biletinə uyğun yeri çox asanlıqla tapa bilər.

Massiv **Array** xidməti sözü vasitəsilə təsvir olunur:

<Tipin adı> : Array [<İndeksələr>] of <Tip>;

Burada, <Tipin adı> - identifikatordur, <İndeksələr> - LongInt və baza tipi LongInt-dən fərqli olan ixtiyari nizamlı tip ilə təsvir oluna bilər, <Tip> isə Turbo Pascal-ın ixtiyari tipidir.

Massivlərin təsvirində indekslər üçün adətən diapazon tipdən istifadə edir, indeksin ala biləcəyi aşağı və yuxarı sərhədlərini göstərir. Məsələn,

x: Array [1..20] of Real;

15. S3 sətində ardıcıl iki eyni hərfin olmasını müəyyənləşdirin. Belə qoşa eyni hərfi olan söz tapılsa, onu ekrana çıxarın.

16. S3 sətində «o» hərfi ilə başlayan sözləri silin və silinmiş sözlərin sayını təyin edin.

17. S3 sətində böyük latın hərflərini uyğun kiçik hərflərlə, kiçik hərfləri isə böyük hərflərlə əvəz edin. Durğu işarələrini olduğu kimi saxlayın.

18. S2 sətində saitlərin sayı ən çox olan sözü tapın.

19. S2 sətində samitlərin sayı ən az olan sözü tapın.

20. S3 sətində uzunluğu ən uzun olan sözə uzunluğu ən qısa olan sözün yerlərini dəyişin. Belə sözlərin sayı bir neçədirsə, cümlədə ilk rast gələnə götürün.

21. Tərkibində ardıcıl bir neçə «aralıq» simvolu olan S sətindən artıq «aralıq» simvollarını silin. Silinmiş simvolların sayını göstərin.

22. $A1 = '5 * 77 = ?'$, $A2 = '75 - 43 = ?'$, $A3 = '972 / 3 = ?'$ və $A4 = '4 + 38 = ?'$ sətir tipli dəyişənləri verilmişdir. Hesablamaları aparıb, hər bir dəyişəndə «?» simvolunun yerinə hesablamaların nəticəsini yazın.

MASSIVLƏR

7.1. Massivlərin təsviri və emalı

Massivlər strukturlaşdırılmış tip verilənlərin bir formalarından biridir. **Massiv** – müəyyən ölçüdə nömrələndirilmiş bir sıra eyni tipli verilənlərin bir adla adlandırılmasıdır. Beləliklə, bir ildə ali məktəbin eyni ixtisasına daxil tələbələrin bir qrupu bir tədris qrupunu təşkil edir və qrup nömrəsi adlandırılır (631 və s.), və ya kinoteatrın sarı divarlı "Sarı" zalda oturmaqlar sırası üzrə kreslolar boyuncə düzülür və eyni vaxtda eyni filmə baxmaq istəyən şəxslər bu kreslolar üzrə yerləşdirilir.

Massiv elementinin **nömrə indeks** adlanır. İndeks – elementlərin sayı bir, və ya bir neçə ola bilər. Bir indeksə təsvir olunan massiv **birölçülü**, iki indekslə təsvir olunan massiv **ikiölçülü** və s. adlanır. Ali riyaziyyatda birölçülü massivlərə **vektor**, ikiölçülü massivlərə isə **matris** anlayışları uyğun gəlir. Beləliklə, massivlərin bir elementinə **indeks** ümumi qəbul edilmiş adını və elementin mədəki indeksini **indeks** ilə müraciət etmək olar.

Məsələn, tələbələrin qrupu jurnalı qrup nömrəsi ilə adlandırılır, jurnalın daxilindəki siyahıda hər bir tələbənin bir sıra nömrəsi qarşı qoyulur. Bu misalda tələbələr adlarından ibarət bir sıra tipli verilənlər birölçülü massiv kimi təsvir olunur. Massivdəki elementlərin emalı zamanı hər bir tələbənin adına qrup və siyahı nömrəsi əlavə edilərək, yəni nömrələri ilə müraciət etmək olar. Kino teatrının üzərində oturmaqların adı ilə yanaşı, oturmaqların sıra və yer nömrələri də göstərilir. Və beləliklə, tələbələr belə təsvirində ikiölçülü massivdən istifadə olunmuşdur və massivlərin hər bir elementi iki indekslə – oturmaqların zal və sıra nömrələrini təyin edən indekslərlə – yer nömrəsi ilə təyin olunmuşdur. Bilet alan şəxs öz biletinə uyğun kresloları tapmaqla öz yerini çox asanlıqla tapa bilər.

Massiv **Array** xidməti vasitəsilə təsvir olunur.

<Tipin adı>: [<İndekslər>] of

Burada, <Tipin adı> - identifikator, <İndekslər> - LongInt və baza tipi LongInt-dən fərqli olan ixtisaslaşdırılmış tip ilə təsvir edilə bilər, <Tip> isə Turbo Pascal-m ixtiyari tipdir.

Massivlərin təsvirində indekslər üçün **ad** və **başlanğıc** diapazon tipdən istifadə edilir, indeksin aşağı və yuxarı limitlərini göstərir. Məsələn,

x: [1..20] of Real;

Bu yazılışda x dəyişəni 20 sayda həqiqi ədəddən ibarət birölçülü massiv kimi təsvir olunmuşdur. Massivin təsvirində <Tip> digər massiv ilə də ifadə oluna bilər:

x: Array [1..20] of Array [1..15] of Real;

Sonuncu yazılışı daha yığcam şəkildə belə yazmaq olar:

x: Array [1..20, 1..15] of Real;

Turbo Pascal-da bir massivin bütün elementlərini birdəfəyə eyni tipli digər massivə mənimsətmək olar. Məsələn,

Var x,y: Array [1..10] of Real;

...

y:=x;

Fraqmentin icrasından sonra x və y massivlərinin uyğun indeksli qiymətləri eyni olacaqdır. Lakin, massivlərin adları üzərində müqayisə əməliyyatı aparmaq mümkün deyil, yəni

if x=y then ...

konstruksiyası icra olunmazdır. İki massivin bərabərliyi onların bütün uyğun elementlərinin ardıcıl müqayisəsi ilə yoxlanıla bilər. Məsələn,

Var x, y : Array [1..10] of Real;

t: Boolean;

i: Integer;

...

t:=True;

for i:=1 to 10 do

if x[i]<>y[i] then t:=False;

if t=True then writeln('x və y bərabərdir')

else writeln('x və y bərabər deyil');

Gündəlik fəaliyyətimizdə daima massivlərin emalı ilə əlaqədar bir sıra məsələlər ilə qarşılaşırıq. Belə emal növlərinə aşağıdakıları göstərmək olar:

- hər hansı əlamətə görə sayın hesablanması. Məsələn, tədris qrupunda oğlan və ya qız tələbələrin sayı, fakültədə əlaçı tələbələrin sayı, məktəbdəki təmirə ehtiyacı olan dərslər otaqlarının sayı və s.;
- hər hansı əlamətə görə verilənlərin cəminin və ya hasilinin hesablanması. Məsələn, şagirdin dərslərinin alınması üçün tələb olunan pul məbləği, tələbələrin orta qiymətlərinin hesablanması üçün hər bir tələbənin aldığı qiymətlərin cəmi, bankda bir iş günündə köçürülən məbləğlərin cəmi və s.;
- verilənlər içərisində ən böyük (maksimum) və ya ən kiçik (minimum) qiymətə malik olan veriləni tapmaq. Məsələn, ən

yüksək tədris göstəricisinə malik tələbənin tapılması, ən kiçik boylu şagirdin seçilməsi, ən çox kapitalla malik maliyyə müəssisəsinin təyin olunması və s.;

- verilənləri hər hansı əlamətə görə nizamlamaq. Məsələn, bədən tərbiyəsi dərslərində şagirdlərin və ya tələbələrin boy sırasına görə, dərslərə davamiyyət jurnallarında şagirdlərin və ya tələbələrin ad siyahılarının əlifbaya görə, televiziya və radio verilişləri proqramlarının dinləyicilərin tələbatlarına görə düzülüşləri və s. Belə məsələlərin həlli ilə bağlı bəzi alqoritmləri və çalışmaları nəzərdən keçirək.

Çalışma 7.1 (Say). 20 sayda təsadüfi götürülmüş ədəddən $[10, 30]$ intervalına düşən ədədlərin sayını tapmalı.

Həlli: Random funksiyasından istifadə edərək, birölçülü X massivinin elementlərini təyin edək. Qiymətləri tələb olunan intervala düşən elementlərin sayını tapmaq üçün k dəyişəni qəbul edib, ona ilkin sıfır qiymətini mənsub edək.

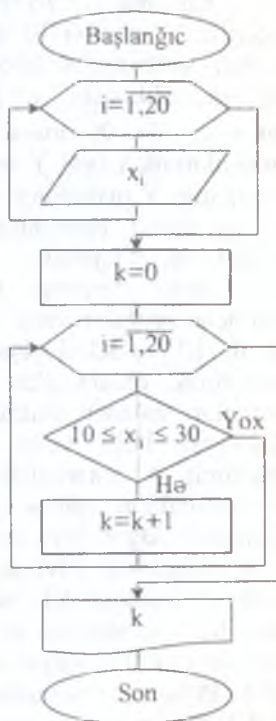
Məsələnin həlli üçün dövrü olaraq X massivinin hər bir elementinin verilmiş intervala düşməsi yoxlanılmalıdır. Əgər element verilmiş intervala daxildirsə, onda k dəyişəninin qiymətinin üzərinə bir vahid əlavə edilməlidir. Əks halda dövr davam etdirilir. Dövrün sonunda k dəyişəninin qiyməti çap edilir. Qeyd etmək lazımdır ki, xüsusi halda, k dəyişəni öz ilkin sıfır qiymətini saxlaya da bilər. Belə hal o vaxt baş verər ki, massivin heç bir elementinin qiyməti tələb olunan intervala düşməsin.

Məsələnin həll alqoritminin blok-sxemi şəkil 7.1-də, tərtib edilmiş proqramın listinqi isə aşağıda göstərilmişdir.

```

Program Say;
Uses WinCrt;
Var i,k: Integer; x: array [1..20] of Integer;
Begin
    Randomize;

```



Şəkil 7.1

```

Write('Ededler: ');
For i:=1 to 20 do Begin x[i]:=Random(70); Write(x[i]:3) End;
k:=0;
For i:=1 to 20 do if (x[i]>=10) and (x[i]<=30) then k:=k+1;
Writeln;
Writeln('k=',k:2)

```

End.

Cavab:

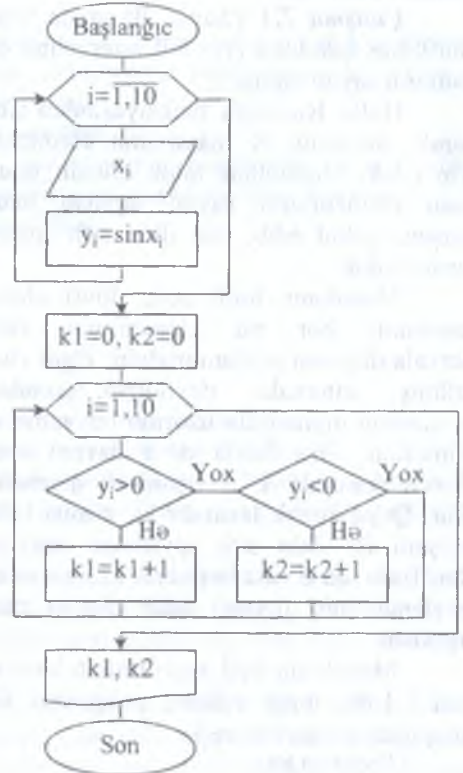
Ededler: 62 12 5 1 28 27 50 14 61 32 56 56 24 39 12 43 36 30 3 30
k=8

Çalışma 7.2 (Sayılar).

Birölçülü X massivi 10 sayda təsadüfi götürülmüş ədədlərdən təşkil edilmişdir. $y = \sin x$ funksiyası ilə X massivin elementlərindən yeni Y massivi yaradılır. Y massivin müsbət və mənfi elementlərinin ayrılıqda sayım tapmalı.

Həlli. Sayların tapılması üçün saygac rolunu oynayan iki $k1$ və $k2$ dəyişənləri götürülmüş, onlara ilkin sıfır qiymətləri mənsub edilmişdir. Məsələnin həlli üçün yeni yaradılmış Y massivin hər bir elementinin işarəsi təyin edilməlidir. Əgər dövr daxilində Y massivin növbəti elementi müsbətdirsə $k1$, mənfidirsə $k2$ dəyişənin qiymətinin üzərinə bir vahid əlavə edilir. Proqramın sonunda $k1$ və $k2$ dəyişənlərinin qiymətləri çap edilir.

Məsələnin həll algoritminin blok-sxemi şəkil 7.2-də, tərtib edilmiş proqramın icrasından alınan nəticələr şəkil 7.3-də, proqramın listinqi isə aşağıda göstərilmişdir.



Şəkil 7.2

```

Program Saylar;
Uses WinCrt;
Var i, k1, k2: Integer;
    x: array [1..10] of Integer;
    y: array [1..10] of Real;
Begin
  Randomize;
  for i:=1 to 10 do Begin
    x[i]:=Random(20);
    y[i]:=sin(x[i]);
    Writeln(x[i]:3, ' ', y[i]:6:4)
  End;
  k1:=0; k2:=0;
  for i:=1 to 10 do Begin
    if y[i]>0 then k1:=k1+1;
    if y[i]<0 then k2:=k2+1
  End;
  Writeln('k1=', k1:2);
  Writeln('k2=', k2:2)
End.

```

[[inactive C:\TPW\SAYLAR.EXE]]

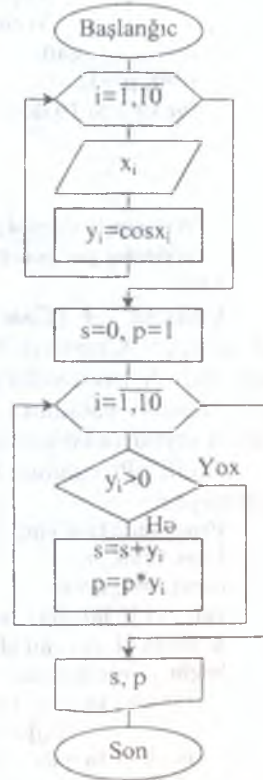
9	0.4121
10	-0.5440
16	-0.2879
18	-0.7510
17	-0.9614
8	0.0000
12	-0.5366
8	0.9894
13	0.4202
11	-1.0000
k1=	3
k2=	6

Şəkil 7.3

Çalışma 7.3 (Cəm və hasil). Birölçülü X massivi təsadüfi götürülmüş 10 ədəddən təşkil edilmişdir. $y = \cos x$ funksiyası ilə X massivin elementlərindən yeni Y massivi yaradılır. Y massivin müsbət elementlərinin cəmini və hasilini tapın.

Həlli: Birölçülü Y massivin müsbət elementlərinin cəmi üçün S dəyişəni qəbul edilmiş və ona ilkin qiymət kimi sıfır qiyməti, bu elementlərin hasilini üçün isə P dəyişəni qəbul edilmiş və ona ilkin qiymət kimi vahid mənsub edilmişdir. Məsələnin həlli üçün yeni yaradılmış Y massivin hər bir elementinin işarəsi təyin edilməlidir. Əgər dövr daxilində Y massivin növbəti elementi müsbətdirsə, bu elementin qiyməti S dəyişəninə elavə edilir, P dəyişəninə isə vurulur. Programın sonunda S və P dəyişənlərinin qiymətləri çap edilir.

Məsələnin həll algoritminin blok-sxemi şəkil 7.4-də, tərtib edilmiş programın icrasından



Şəkil 7.4

alınan nəticələr şəkil 7.5-də, proqramın listingi isə aşağıda göstərilmişdir.

```

Program CemHasil;
Uses WinCrt;
Var i: Integer;
    s, p: Real;
    x: array [1..10] of Integer;
    y: array [1..10] of Real;
Begin
    Randomize;
    for i:=1 to 10 do begin
        x[i]:=Random(20);
        y[i]:=cos(x[i]);
        Writeln(x[i]:3, ' ', y[i]:6:4)
    end;
    s:=0; p:=1;
    for i:=1 to 10 do if y[i]>0 then Begin
        s:=s+y[i];
        p:=p*y[i]
    End;
    Writeln('s=',s:6:4);
    Writeln('p=',p:6:4)
End.

```

1	0.5403
3	-0.9900
8	-0.1455
13	0.9074
12	0.8439
13	0.9074
9	-0.9111
17	-0.2752
10	-0.8391
15	-0.7597
s=3.1990	
p=0.3754	

Şəkil 7.5

Çalışma 7.4 (Cəm və say). n sayda sətir və m sayda sütundan ibarət ikiölçülü X massivi 30-dan böyük olmayan müsbət ədədlərdən təşkil olunmuşdur. X massivin elementlərindən $y=\sin(x)$ funksiyası vasitəsilə yeni Y massivi yaradılır. Y massivin hər bir sətiri üçün müsbət elementlərin sayını və cəmini tapmalı.

Həlli: Proqramın listingi və icrasından alınan cavablar aşağıda göstərilmişdir.

```

Program MasCem;
Uses WinCrt;
const n=4; m=6;
var i,j,k: integer; s: real;
    x: array [1..n,1..m] of integer; y: array [1..n,1..m] of real;
begin
    randomize;
    for i:=1 to n do for j:=1 to m do begin
        x[i,j]:=random(30); y[i,j]:=sin(x[i,j]); end;
    for i:=1 to n do begin
        writeln;
        for j:=1 to m do write(' ',x[i,j]:2); write(' ');
        for j:=1 to m do write(' ',y[i,j]:5:2);
    end;

```

```

writeln; writeln;
for i:=1 to n do begin
    k:=0; s:=0;
    for j:=1 to m do if y[i,j]>0 then begin
        k:=k+1; s:=s+y[i,j]; end;
    writeln('i=',i, ' k=',k, ' s=',s:5:2);
end;

```

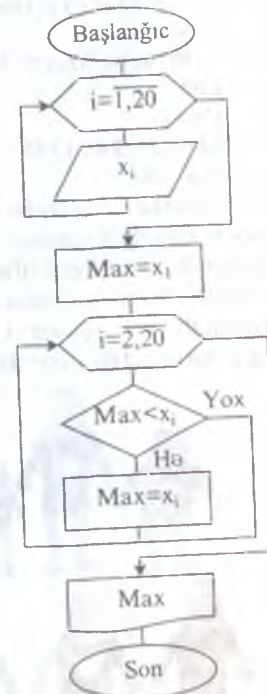
Cavablar:

9	27	13	26	6	23	0.41	0.96	0.42	0.76	-0.28	-0.85
4	0	1	2	12	20	-0.76	0.00	0.84	0.91	-0.54	0.91
25	4	2	20	18	18	-0.13	-0.76	0.91	0.91	-0.75	-0.75
6	12	27	24	7	27	-0.28	-0.54	0.96	-0.91	0.66	0.96
I=1	k=4		s=2.55								
I=2	k=3		s=2.66								
I=3	k=2		s=1.82								
I=4	k=3		s=2.57								

Çalışma 7.5 (Maksimum). Birölçülü X massivi 20 sayda təsadüfi götürülmüş elementdən təşkil edilmişdir. X massivin ən böyük qiymətə malik elementini təyin edin.

Həlli. Proqramın icrası zamanı ilk növbədə X massivin elementlərinə təsadüfi götürülmüş qiymətlər mənsub edilir. Maksimal qiymətə malik elementi tapmaq üçün həll algoritmində X massivin birinci elementi maksimal element kimi qəbul edilir və bu qiymət Max dəyişəninə mənsub edilir. Dövr dəyişənin 2-dən 20-dək qiymətlərində massivin uyğun elementləri dövr olaraq Max dəyişəni ilə müqayisə edilir. Hər bir müqayisə zamanı massivin uyğun elementi Max dəyişənin qiymətindən böyük olarsa, Max dəyişəninə massivin bu elementin qiyməti mənsub edilir. Dövrün sonunda Max dəyişəni maksimal qiyməti özündə saxlayacaqdır.

Qeyd etmək lazımdır ki, massivin minimal qiymətə malik elementin tapılması da anoloji alqoritmlə həyata keçirilə bilər, belə ki, bu zaman müqayisə zamanı «kiçik» işarəsini «böyük» işarəsi ilə əvəz etmək lazımdır.



Şəkil 7.6

Məsələnin həll alqoritminin blok-sxemi şəkil 7.6-da, tərtib edilmiş programın listingi isə aşağıda göstərilmişdir.

```

Program Maksimum;
Uses WinCrt;
Var i, Max: Integer;
    x: Array [1..20] of Integer;
Begin
    Randomize;
    For i:=1 to 20 Do Begin
        x[i]:=Random(70);
        Write(x[i], ' ')
    End;
    Max:=x[1];
    For i:=2 to 20 do
        if Max<x[i] then Max:=x[i];
    WriteLn;
    WriteLn('Max=',Max)
End.

```

End.

Cavab:

54 53 59 3 42 13 58 25 67 4 58 57 24 64 46 66 69 55 27 47

Max=69

İnsanlar rastlaşdıqları məsələlərin həllinə yönəlmiş müxtəlif prosesləri həyata keçirmək üçün verilənlərin müxtəlif əlamətlərə görə nizamlı düzülüşlərindən istifadə edirlər. Bu kimi məsələlərin kompüter həlli üçün verilənlərin hər hansı əlamətə görə nizamlanması alqoritmi işlənilməli və həyata keçirilməlidir. Nizamlama alqoritmi müxtəlif üsullarla tərtib edilə bilər. Bu üsullardan birini nəzərdən keçirək.



A1, A2, A3, A4, A5 dəyişənləri uyğun olaraq 7, 3, 9, 2, 6 qiymətlərinə malikdirlər. Bu dəyişənlərə verilmiş ədədlərin artma ardıcılığı ilə götürülmüş qiymətlərini mənsub etmək üçün aşağıda göstərilən mərhələlər yerinə yetirilə bilər.

1-ci mərhələ. A1 dəyişənin qiyməti özündən sonrakı dəyişənlərin (A2, A3, A4, A5) qiymətləri ilə ardıcıl olaraq müqayisə edilir. Bu dəyişənin qiyməti müqayisə olunan dəyişənin qiymətindən böyük olarsa, qiymətlərin yerdəyişməsi üçün əməliyyatlar aparılır. Əks halda qiymətlərin yerdəyişməsi baş vermir və müqayisə olunan dəyişənlər öz əvvəlki qiymətlərini saxlayırlar. 1-ci mərhələdən sonra A1 dəyişənin qiyməti ən kiçik olur.

A1	A2	A3	A4	A5
7	3	9	2	6

A1 > A2 olduğundan A1 və A2 dəyişənləri öz qiymətlərini bir-biri ilə dəyişir

A1	A2	A3	A4	A5
3	7	9	2	6

A1 < A3 olduğundan A1 və A3 dəyişənlərinin qiymətləri öz yerlərində qalır

A1	A2	A3	A4	A5
3	7	9	2	6

A1 > A4 olduğundan A1 və A4 dəyişənləri öz qiymətlərini bir-biri ilə dəyişir

A1	A2	A3	A4	A5
2	7	9	3	6

A1 < A5 olduğundan A1 və A5 dəyişənlərinin qiymətləri öz yerlərində qalır

A1	A2	A3	A4	A5
2	7	9	3	6

2-ci mərhələ. A2 dəyişənin qiyməti özündən sonrakı dəyişənlərin (A3, A4, A5) qiymətləri ilə ardıcıl olaraq müqayisə olunur. 1-ci mərhələyə analogi olaraq müqayisə olunan dəyişənlərin qiymətləri arasında yerdəyişmə həyata keçirilir.

A2 < A3 olduğundan A2 və A3 dəyişənlərinin qiymətləri öz yerlərində qalır

A1	A2	A3	A4	A5
2	7	9	3	6

A2 > A4 olduğundan A2 və A4 dəyişənləri öz qiymətlərini bir-biri ilə dəyişir

A1	A2	A3	A4	A5
2	3	9	7	6

A2 < A5 olduğundan A2 və A5 dəyişənlərinin qiymətləri öz yerlərində qalır

A1	A2	A3	A4	A5
2	3	9	7	6

3-cü mərhələ. A3 dəyişənin qiyməti özündən sonrakı dəyişənlərin (A4, A5) qiymətləri ilə ardıcıl olaraq müqayisə olunur. 1-ci mərhələyə analogi olaraq müqayisə olunan dəyişənlərin qiymətləri arasında yerdəyişmə həyata keçirilir.

A3>A4 olduğundan A3 və A4 dəyişənləri öz qiymətlərini bir-biri ilə dəyişir

A1	A2	A3	A4	A5
2	3	7	9	6

A3>A5 olduğundan A3 və A5 dəyişənləri öz qiymətlərini bir-biri ilə dəyişir

A1	A2	A3	A4	A5
2	3	6	9	7

4-cü mərhələ. A4 dəyişənin qiyməti özündən sonrakı dəyişənin (A5) qiyməti ilə müqayisə olunur. 1-ci mərhələyə analogi olaraq dəyişənlərin qiymətləri arasında yerdəyişmə həyata keçirilir.

A4>A5 olduğundan A4 və A5 dəyişənləri öz qiymətlərini bir-biri ilə dəyişir

A1	A2	A3	A4	A5
2	3	6	7	9

Nizamlama alqoritmində mərhələlərin icrası zamanı iç-içə yerləşdirilmiş 2 dövrədən istifadə olunmuşdur. Xarici dövr birinci dəyişəndən (A1) başlayır və sonrakı dəyişənlər də daxil olmaqla sonuncudan əvvəlki dəyişənədək (A2, A3, A4) davam edir. Daxili dövr isə hər dəfə i saylı xarici dövr dəyişənindən (A_i) sonrakı dəyişəndən (A_{i+1}) başlayır və sonuncu dəyişənədək (A5) davam edir.

Müqayisə olunan dəyişənlərin qiymətlərinin dəyişdirilməsi zamanı adətən üçüncü dəyişəndən istifadə edirlər. Məsələn, k1 və k2 dəyişənlərinin qiymətlərinin qarşılıqlı dəyişdirilməsi zamanı m köməkçi dəyişənindən istifadə olunmaqla, aşağıdakı əməllər ardıcılığı yerinə yetirilir:

```

...          k1=3  k2=7  m=...
m:=k1        k1=3  k2=7  m=3
k1:=k2        k1=7  k2=7  m=3
k2:=m         k1=7  k2=3  m=3

```

Bekə bir sual qarşıya çıxır: "İki dəyişənin qiymətlərinin yerdəyişməsinə üçüncü dəyişəndən istifadə etmədən yerinə yetirmək olarmı?". Bu sual ətrafında düşünməyə dəyər və oxucuya tövsiyə olunur ki, bir qədər vaxtını cavabın tapılmasına sərf etsin.

Sualın cavabı müsbətdir aşağıda göstərilmişdir.

...
 $k1 = k1 + k2$ $k1 = 10$ $k2 = 7$
 $k2 = k1 - k2$ $k1 = 10$ $k2 = 3$
 $k1 = k1 - k2$ $k1 = 7$ $k2 = 3$

Çalışma 7.6 (Nizamlama). Hər biri 100 ədədindən kiçik olan 20 sayda mənfi olmayan tam ədədi artma ardıcılığı ilə düzməli.

Həlli: Tərtib edilmiş proqramda ilkin verilənlərin seçilməsi zamanı randomizə təsadüfi ədədlər generatorundan istifadə edilmişdir. $X[20]$ massivinin elementlərinin qiymətləri $\text{random}(100)$ funksiyasından istifadə olunmaqla 0-dan 100-dək təsadüfi ədədlər siyahısından seçilir. Proqramın həll algoritminin blok-sxemi şəkil 7.7-də, listingi və icrasından alınan nəticələr isə aşağıda, göstərilmişdir.

Program Nizamlama;

Uses WinCrt;

const n=20;

var i,j,m: integer;

x: array [1..n] of integer;

begin

writeln('Nizamlamadan evvel');

randomize;

for i:=1 to n do begin

x[i]:=random(100); write('x[i];
end;

writeln;

for i:=1 to n-1 do for j:=i+1 to n do

if x[i]>x[j] then begin

m:=x[i]; x[i]:=x[j]; x[j]:=m
end;

writeln('Nizamlamadan sonra'); **for i:=1 to n do write('x[i];**

end.

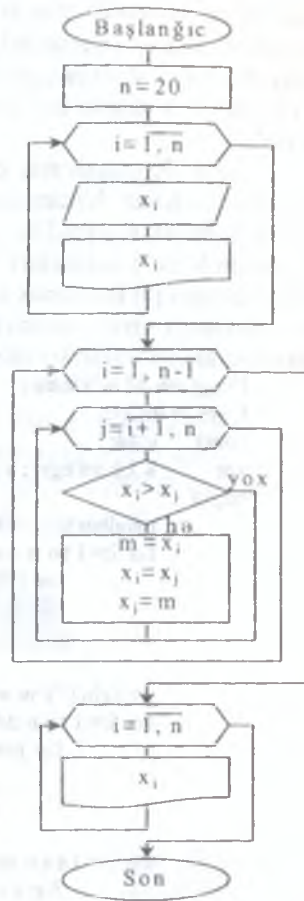
Cavablar:

Nizamlamadan evvel

10 66 50 19 57 24 60 26 66 59 91 81 1 2 73 22 24 52 49 32

Nizamlamadan sonra

1 2 10 19 22 24 24 26 32 49 50 52 57 59 60 66 66 73 81 91



Şəkil 7.7

Çalışma 7.7 (Matrisdə nizamlama). Qiymətləri vahiddən kiçik və mənfi olmayan ədədlərdən təşkil olunmuş, $N \times N$ sayda elementdən ibarət ikiölçülü X massivi verilmişdir. Elementləri massivin $i, j \leq N$ indeksli x_{ij} elementlərindən $y_{ij} = \sin(x_{ij}) + \cos(x_{ij})$ funksiyası ilə hesablanmaqla alınan yeni ikiölçülü Y massivin hər bir sətri üçün elementləri artan ardıcılıqda düzməli.

Həlli: X massivin elementləri $\text{random}(1000)/1000$ ifadəsi vasitəsilə təyin olunur. Nizamlama əməliyyatının Y massivin hər bir sətri üçün aparılması məqsədilə i dövr dəyişənli xarici dövr təşkil olunur. Programda k və j indeksləri i -ci sətrin k və j saylı sütunlarında yerləşən elementləri müqayisə etmək üçün istifadə olunmuşdur. Bu indekslər daxili dövr dəyişənlərinin qiymətlərini təyin edir. Programın listinqi və icrasından alınan nəticələr aşağıda göstərilmişdir.

```
Program MasNizam;
Uses WinCrt;
const n=4;
var k,i,j: integer; x,y: array [1..n,1..n] of real; m: real;
begin
  randomize; writelnt('X massivi');
  for i:=1 to n do begin
    for j:=1 to n do begin
      x[i,j]:=random(1000)/1000; write(' ',x[i,j]:5:3);
    end; writelnt;
  end;
  writelnt('Y massivi, ilkin');
  for i:=1 to n do begin
    for j:=1 to n do begin
      y[i,j]:=sin(x[i,j]); write(' ',y[i,j]:5:3);
    end; writelnt;
  end;
  for i:=1 to n do
    for k:=1 to n-1 do
      for j:=k+1 to n do
        if y[i,k]>y[i,j] then begin
          m:=y[i,k]; y[i,k]:=y[i,j]; y[i,j]:=m;
        end;
      end;
    end;
  writelnt('Y massivi, nizamlı');
  for i:=1 to n do begin for j:=1 to n do write(' ',y[i,j]:5:3);
  writelnt; end;
end;
```

end.

Cavablar:

X massivi

0.903	0.798	0.669	0.234
0.555	0.074	0.681	0.267
0.519	0.597	0.333	0.251
0.839	0.039	0.964	0.589

Y massivi, ilkin

0.785	0.716	0.620	0.232
0.527	0.074	0.630	0.264
0.496	0.562	0.327	0.248
0.744	0.039	0.821	0.556

Y massivi, nizamlı

0.232	0.620	0.716	0.785
0.074	0.264	0.527	0.630
0.248	0.327	0.496	0.562
0.039	0.556	0.744	0.821

Çalışma 7.8 (Şəhərlərarası məsafə). Dekart koordinat sisteminə n ($n > 5$) sayda şəhər öz koordinatları ilə göstərilmişdir. Bu şəhərlərin hər birindən digərlərinə yol çəkilmişdir və bu yollar düzxətli qəbul edilir. Hansı iki şəhər arasındakı yolun ən uzun olduğunu təyin edin.



Həlli. Koordinat sisteminə n sayda şəhər göstərilmişdir. Bu şəhərləri aşağıdakı kimi adlandırmaq:

$A_1(x_1, y_1), A_2(x_2, y_2), A_3(x_3, y_3), \dots, A_n(x_n, y_n)$.

İxtiyari iki i və j saylı şəhərlər arasındakı məsafə riyaziyyatdan bəlli

$$d^2 = (x_1 - x_2)^2 + (y_1 - y_2)^2$$

düsturu ilə tapıla bilər. Məsələnin həlli üçün bütün şəhərlərarası məsafələr hesablanmalı və onlardan ən böyüyü seçilməlidir. Şəhərlərarası d məsafələri D matrisi ilə təsvir oluna bilər:

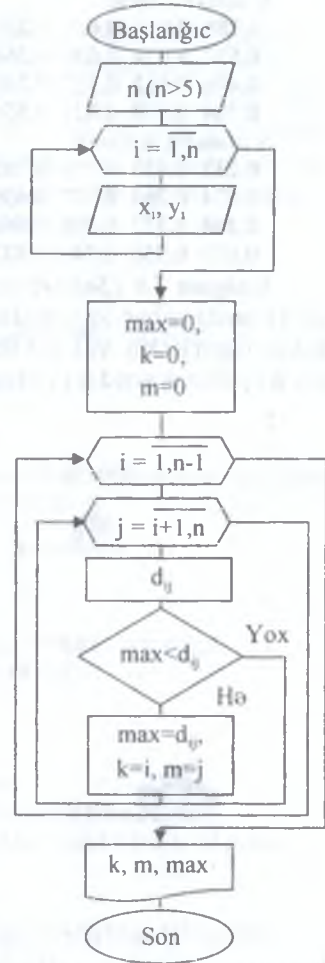
$$D = \begin{bmatrix} d_{11} & d_{12} & d_{13} & \dots & d_{1n} \\ d_{21} & d_{22} & d_{23} & \dots & d_{2n} \\ d_{31} & d_{32} & d_{33} & \dots & d_{3n} \\ \dots & \dots & \dots & \dots & \dots \\ d_{n1} & d_{n2} & d_{n3} & \dots & d_{nn} \end{bmatrix}$$

D matrisinin i -ci sətiri ilə j -ci sütununda yerləşən d_{ij} elementi i və j sayılı şəhərlər arasındakı məsafəni təyin edir. Şəhərin özü ilə məsafəsi 0 olduğundan, ixtiyari i indeksi üçün $d_{ii}=0$ olar. A_i, A_j və A_j, A_i şəhərləri arasındakı məsafələr cəmiyyətinə mənə daşdığından $d_{ij} = d_{ji}$ qəbul etmək olar.

Beləliklə, məsələnin həlli üçün D matrisinin yalnız baş diaqonalından yuxarıda yerləşən elementləri maraq kəsb edir. Bu elementlər D' matrisində təsvir edilmişdir:

$$D' = \begin{bmatrix} \dots & d_{12} & d_{13} & \dots & d_{1n} \\ \dots & \dots & d_{23} & \dots & d_{2n} \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & d_{n-1n} \\ \dots & \dots & \dots & \dots & \dots \end{bmatrix}$$

Sonuncu matrisdən görünür ki, d_{ij} məsafələrini tapmaq üçün i indeksi 1-dən $(n-1)$ -dək, j indeksi isə, hər i sayılı sətir üçün $(i+1)$ -dən n -dək qiymət alması kifayətdir. Bu məsafələr tapıldıqdan sonra, ikölçülü massivlər üçün, uyğun indekslər nəzərə alınmaqla elementlərin maksimumunun tapılması alqoritmini tətbiq etmək olar. Tapılmış maksimal elementin i və j in-



Şəkil 7.8

deksləri arasındakı məsafə ən uzun olan iki şəhəri təyin edəcəkdir. $n=6$ sayda şəhər üçün programın həll algoritminin blok-sxemi şəkil 7.8-də, listingi və icrasından alınan cavablar isə aşağıda göstərilmişdir.

```

Program Məsafə;
Uses WinCrt;
const n=6;
var i, j, k, m: integer;    max: real;
    d: array [1..n,1..n] of real;    x,y: array [1..n] of integer;
begin
  randomize;
  for i:=1 to n do begin
    x[i]:=random(100); y[i]:=random(100);
    writeln('x|',i:1,'|=' ,x[i]:3,'    y|',i:1,'|=' ,y[i]:3);
  end;
  max:=0; k:=0; m:=0;
  for i:=1 to n-1 do begin
    writeln; for j:=i+1 to n do begin
      d[i,j]:=sqrt(sqr(x[i]-x[j]) +sqr(y[i]-y[j]));
      write('d|',i:2,',',j,'|=' ,d[i,j]:5:1); write(' ');
      if max < d[i,j] then begin max:=d[i,j]; k:=i; m:=j end;
    end;
  end;
  writeln; writeln;
  writeln('k=' ,k:2,'    m=' ,m:2); writeln('d_max=' ,max:5:1);
end.

```

Cavablar:

```

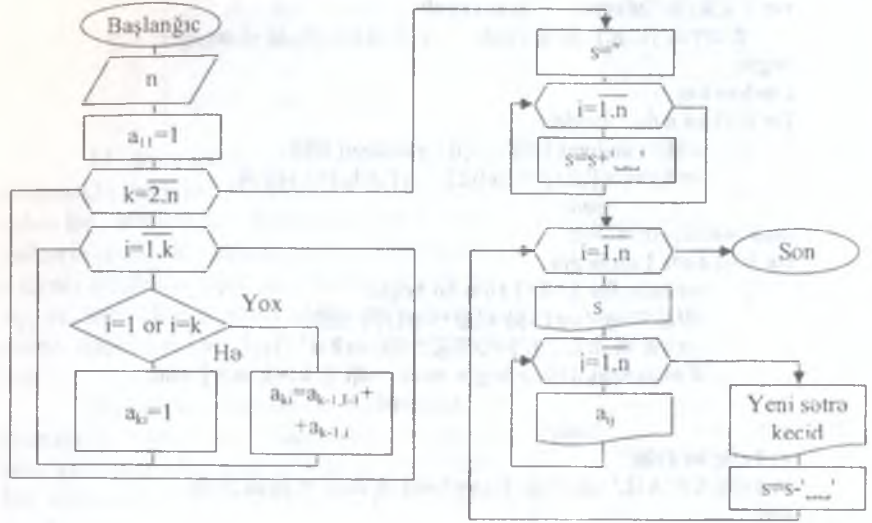
x[1]=75          y[1]=14
x[2]=17          y[2]=58
x[3]=71          y[3]=0
x[4]=53          y[4]=84
x[5]=50          y[5]=41
x[6]=36          y[6]=39
d[1,2]=72.8    d[1,3]=14.6    d[1,4]=73.4    d[1,5]=36.8    d[1,6]=46.3
d[2,3]=79.2    d[2,4]=44.4    d[2,5]=37.1    d[2,6]=26.9
d[3,4]=85.9    d[3,5]=46.1    d[3,6]=52.4
d[4,5]=43.1    d[4,6]=48.1
d[5,6]=14.1
k=3    m=4
d_max=85.9

```

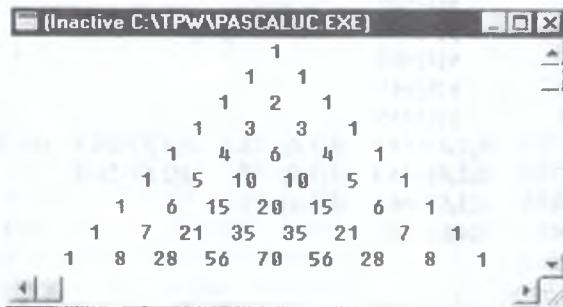
Çalışma 7.9 (Paskal üçbucağı). Natural n sayda sətirdə yerləşən ədədlərdən təşkil olunmuş üçbucağın yan tərəfindəki ədədlər 1,

oturacağındakı və daxilində yerləşən hər bir ədəd isə, özündən yuxarıdakı sətirdə ona ən yaxın olan iki ədədin cəminə bərabərdir. Verilmiş n sayda sətirdən ibarət Paskal üçbucağını qurmalı.

Həlli. Programın həll algoritminin blok-sxemi şəkil 7.9-da, icrasından alınan cavablar şəkil 7.10-da, listingi isə aşağıda göstərilmişdir:



Şəkil 7.9



Şəkil 7.10. "Pascal_Ucbucaq" programının icrasından alınan nəticələr

```

Program Pascal_Ucbucaq;
Uses WinCrt;
Const n=9;
Var k,i,j: Word; s: String; a: array [1..n,1..n] of integer;
Begin
  a[1,1]:=1;
  for k:=2 to n do for i:=1 to k do
    if (i=1) or (i=k) then a[k,i]:=1
      else a[k,i]:=a[k-1,i-1]+a[k-1,i];
    s:=''; for i:=1 to n do s:=s+' ';
    for i:=1 to n do begin write(s);
      for j:=1 to i do write(' ', a[i,j]:3);
      writeln; delete(s,1,2);
    end;
  end;

```

End.

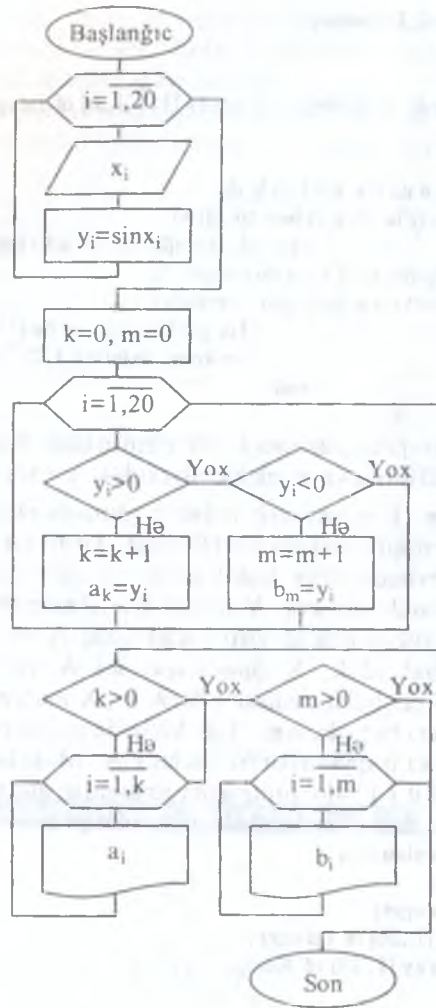
Çalışma 7.10 (Parçalanma). 20 elementdən ibarət birölçülü X massivi verilmişdir. Bu massivin elementlərindən $y = \sin x$ funksiyası ilə yeni Y massivi alınır. Y massivin müsbət qiymətli elementlərindən A , mənfi qiymətli elementlərindən isə yeni B massivlərini yaratmalı.

Həlli. İlk növbədə dövr təşkil edilir və dövr daxilində X massivin elementləri daxil olunaraq, Y massivin elementləri hesablanır. k və m dəyişənlərini uyğun olaraq yeni yaradılacaq A və B massivlərinin indeksləri üçün qəbul edək. X massivin iki A və B massivlərinə ayrılması prosesində ehtisasa halla rastlaşa bilərik ki, X massivin ya müsbət, və ya mənfi qiymətləri heç olmasın. Belə hallarda uyğun olaraq k və ya m dəyişənləri öz əvvəlki 0 qiymətlərini saxlayırlar. Məsələnin həll algoritminin blok-sxemi şəkil 7.11-də, programın icrasından alınan nəticələr şəkil 7.12-də, programın listingi isə aşağıda göstərilmişdir.

```

Program Parçalanma;
Uses WinCrt;
Var i,k,m: Integer;
    x: array [1..20] of Integer;
    y,a,b: array [1..20] of Real;
Begin
  Randomize;
  For i:=1 to 20 do Begin
    x[i]:=random(100);
    y[i]:=sin(x[i])
  End;
  Writeln('i: X matr: Y matr: i: X matr: Y matr: ');
  For i:=1 to 10 do Begin Write(i:2,x[i]:8,y[i]:8:2,' ');

```



Şəkil 7.11

Write(i+10:2,x[i+10]:8,y[i+10]:8:2);

WriteLn;

End;

For i:=1 to 20 do

```

    if y[i]>0 then Begin k:=k+1; a[k]:=y[i] End
    else if y[i]<0 then Begin m:=m+1; b[m]:=y[i] End;
  If k>0 then Begin Write('A matrisi: ');
    For i:=1 to k do Write(a[i]:5:2, ' '); Writeln;
  End;
  If m>0 then Begin Write('B matrisi: ');
    For i:=1 to m do Write(b[i]:5:2, ' '); Writeln;
  End;
End.

```

```

[[Inactive C:\TPWAP7_5 EXE]]
i: X matr: Y matr:      i: X matr: Y matr:
1      25      -0.13      11      90      0.89
2       3       0.14      12      46      0.98
3      94      -0.25      13      37      -0.64
4       4      -0.76      14       8      0.99
5      50      0.99      15       8      0.88
6      13      0.42      16      36      -0.99
7      58      0.99      17      57      0.44
8      93      -0.95      18      38      0.38
9       25      -0.13      19      16      -0.29
10     99      -1.88      20       9      0.41

A matrisi:  0.14  0.99  0.42  0.99  0.89  0.98  0.99  0.44  0.38  0.41
B matrisi: -0.13 -0.25 -0.76 -0.95 -0.13 -1.88 -0.64 -0.99 -0.29

```

Şəkil 7.12. "Parçalanma" programının icrasından alınan nəticələr

7.2. Laboratoriya işi № 9. Massivlərin emalı

1. 20 elementdən ibarət birölçülü X massivi verilmişdir. X massivin elementlərindən $y = 2 \sin x$ funksiyası ilə yeni Y massivi yaradılır. Y massivin müsbət elementlərinin sayını tapmalı.

2. 15 elementdən ibarət birölçülü X massivi verilmişdir. X massivin elementlərindən $y = \frac{2 \sin x}{x^2 + 1}$ funksiyası ilə yeni Y massivi yaradılır.

Y massivin ayrılıqda müsbət və mənfi elementlərinin sayını tapmalı.

3. 20 elementdən ibarət birölçülü X massivi verilmişdir. X massivin elementlərindən $y = \cos x - \sin x^2$ funksiyası ilə yeni Y massivi yaradılır. Y massivin mənfi elementlərinin cəminin modulunu tapmalı.

4. 12 elementdən ibarət birölçülü X massivi verilmişdir. X massivin elementlərindən $y = \cos^3 x \cdot \sqrt{|\sin x|}$ funksiyası ilə yeni Y massivi yaradılır. Y massivin müsbət elementlərinin ədədi ortasını tapmalı.

5. 12 elementdən ibarət birölçülü X massivi verilmişdir. X massivin elementlərindən $y = \cos^3 \sqrt{|x|}$ funksiyası ilə yeni Y massivi yaradılır. Y massivin $[-0,5; 0]$ intervalına düşən elementlərindən yeni A massivi, $[0,5; 1]$ intervalına düşən elementlərindən yeni B massivi, qalan elementlərindən isə yeni C massivi yaradın.

6. 20 sayda müsbət elementdən ibarət birölçülü X massivi verilmişdir. X massivin qiyməti 40-dan kiçik olan elementlərini 2,3-ə vurmaq, qiymətləri 40-dan böyük və ya bərabər olan elementlərdən isə kvadrat kök almaqla yeni Y massivi yaradın.

7. 25 elementdən ibarət birölçülü X massivi verilmişdir. X massivin elementlərindən $y = \frac{\sin x^2}{1 + \sqrt{|\cos x|}}$ funksiyası ilə yeni Y massivi yaradılır. Y massivin ən böyük elementini tapın.

8. X[10] və Y[8] massivləri verilmişdir. İlk 10 elementi X, sonrakı 8 elementi Y massivindən götürülmüş yeni Z[18] massivi yaradın.

9. 10 elementdən ibarət birölçülü X massivi verilmişdir. X massivin elementlərindən $y = \frac{x+1}{|\sin x| + 0,25}$ funksiyası ilə yeni Y massivi yaradılır. Y massivin ən kiçik elementini tapın.

10. X[10] və Y[10] massivləri verilmişdir. Elementləri X massivin birinci, Y massivin birinci, X massivin ikinci, Y massivin ikinci və s. qaydada götürülmüş elementlərindən təşkil edilmiş yeni Z massivi yaradın.

11. X[N] ($N > 0$) massivi verilmişdir. Massivin indeksi 3-ə tam bölünən elementlərini kvadrata yüksəltməli, qalan elementləri isə 2-yə vurmali.

12. X[N] ($N > 0$) massivi verilmişdir. Massivin tək saylı indeksli elementlərinin cəmindən cüt saylı elementlərinin cəmini çıxmalı.

13. 8×6 elementdən ibarət ikiölçülü X massivi verilmişdir. X massivin elementlərindən $y = \sin x$ funksiyası ilə yeni ikiölçülü Y massivi yaradılır. Y massivin hər bir sətrindəki müsbət elementlərin sayını və cəmini tapmalı.

14. 7×9 elementdən ibarət ikiölçülü X massivi verilmişdir. X massivin hər bir sütunundakı maksimal elementi və onun indekslərini tapmalı. Maksimal element bir neçə sayda isə, nəticə kimi kiçik indeksli elementi götürməli.

15. 6×6 elementdən ibarət X kvadrat matrisi verilmişdir. X massivin baş diaqonalında yerləşən elementlərdən ən kiçiyini və bu elementin indekslərini tapmalı. Minimal element bir neçə sayda isə, nəticə kimi böyük indeksli elementi götürməli.

16. 3×10 sayda elementdən təşkil olunmuş X matrisi verilmişdir. X matrisinin sətirlərindən təşkil edilmiş A , B və C birölçülü matrislərini yaratmalı.

17. 8×10 sayda elementdən təşkil olunmuş X matrisi verilmişdir. Matrisin ardıcıl olaraq götürülmüş k -cəq saylı sətirlərini, özündən sonrakı cüt saylı sətirlə yerlərini dəyişməli.

18. 12 sayda elementdən ibarət birölçülü X massivin elementləri oğlan və qız adlarından təşkil olunub. Massivin elementlərini əlifba sırasına görə nizamlayın.

19. 2×8 sayda elementdən ibarət ikiölçülü X massivi verilmişdir. Massivin birinci sətirə oğlan, ikinci sətirə isə qız adları yazın. Sətir elementlərinə görə matrisi nizamlayın.

20. 20 sayda elementdən ibarət birölçülü X massivi şəhər, ölkə və dəniz adlarından təşkil olunub. Elementlərin birinci simvolu göstərici rolunu oynayır və S – «Şəhər», O – «Ölkə», D – «Dəniz» simvollarından biri ilə ifadə olunur. Məsələn, «DXezer», «SBaki», «Oİtaliya» və s. X massivin elementlərindən istifadə edib, şəhər adlarından ibarət A , ölkə adlarından ibarət B və dəniz adlarından ibarət C birölçülü massivlərini yaradın.

21. 12 sayda elementdən ibarət birölçülü X massivi ölkə adlarından təşkil olunub. Elementlərin birinci simvolu kimi hansı simvolun daha çox istifadə olunmasını və onun sayını təyin edin.

22. 2×3 elementdən ibarət sətir tipli ikiölçülü X massivin hansı elementindəki cümlə daha çox sözdən təşkil olunmuşdur. Bu sözlərin sayını təyin edin.

23. 2×3 elementdən ibarət sətir tipli ikiölçülü X massivin hansı elementindəki cümlə bu elementlərə daxil olan ən uzun sözün özündə saxlayır. Cümlələrdəki ən uzun söz bir neçə yerdə rast gəlinirsə, birinci istifadə olunan cümləni götürməli.

24. 3×2 elementdən ibarət sətir tipli ikiölçülü X massivinin hansı elementindəki cümlədə daha çox durğu işarəsindən istifadə olunmuşdur. Belə elementlərin sayı bir neçədirsə, onların hamısını müəyyən etməli.

25. 15 elementdən ibarət simvol tipli birölçülü massiv elementləri təsadüfi götürülmüş böyük latın hərfləri simvollarından təşkil olunmuşdur. Massivdə hansı simvoldan daha çox və neçə dəfə istifadə olunduğunu təyin edin.

26. 20 elementdən ibarət simvol tipli birölçülü massiv elementləri təsadüfi götürülmüş böyük latın hərfləri simvollarından təşkil olunmuşdur. Massivdə latın əlifbasına daxil olan çatışmayan simvolları siyahı şəklində göstərin.

27. 20 elementdən ibarət simvol tipli birölçülü massiv elementləri təsadüfi götürülmüş kiçik latın hərfləri simvollarından təşkil olunmuşdur. Massivdə təkrarlanan simvolları və bu simvollar üçün təkrarlanmaların sayını müəyyən edin.

28. $A[5]$ birböcülü massivi və $x=1,5$ verilib. Massivin birinci elementi $2\sin x$, ikinci elementi $2\cos x$, üçüncü elementi $\sqrt[4]{x}$, dördüncü elementi $\ln x$, beşinci elementi isə $\arctg x$ qiymətini alır. Massivin ən böyük və ən kiçik qiymət alan elementlərinin cəmini tapmalı.

29. $X[6,6]$ massivinin «*» simvolu ilə qeyd olunmuş elementlərinin cəmini tapmalı:

29.1.

$$X = \begin{pmatrix} * & & * & . & * & . \\ . & * & . & * & . & * \\ * & . & * & . & * & . \\ . & * & . & * & . & * \\ * & . & * & . & * & . \\ . & * & . & * & . & * \end{pmatrix}$$

29.4.

$$X = \begin{pmatrix} * & . & . & . & . & . \\ * & * & . & . & . & . \\ * & * & * & . & . & . \\ * & * & * & * & . & . \\ * & * & * & * & * & . \\ * & * & * & * & * & * \end{pmatrix}$$

29.2.

$$X = \begin{pmatrix} * & . & . & . & . & * \\ . & * & . & . & * & . \\ . & . & * & * & . & . \\ . & . & * & * & . & . \\ . & * & . & . & * & . \\ * & . & . & . & . & * \end{pmatrix}$$

29.5.

$$X = \begin{pmatrix} * & * & * & * & * & * \\ . & * & * & * & * & * \\ . & . & * & * & * & * \\ . & . & . & * & * & * \\ . & . & . & . & * & * \\ . & . & . & . & . & * \end{pmatrix}$$

29.3.

$$X = \begin{pmatrix} . & . & * & * & . & . \\ . & . & * & * & . & . \\ * & * & * & * & * & * \\ * & * & * & * & * & * \\ . & . & * & * & . & . \\ . & . & * & * & . & . \end{pmatrix}$$

29.6.

$$X = \begin{pmatrix} . & . & . & * & * & * \\ * & * & * & . & . & . \\ . & . & . & * & * & * \\ * & * & * & . & . & . \\ . & . & . & * & * & * \\ * & * & * & . & . & . \end{pmatrix}$$

25. İndeksler arasındaki mövcud asılılıqlardan istifadə etməklə aşağıdakı matrisləri formalaşdırın:

30.1.

$$X = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

30.3.

$$X = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{pmatrix}$$

30.2.

$$X = \begin{pmatrix} 1 & 4 & 9 & 16 & 25 & 36 \\ 0 & 1 & 4 & 9 & 16 & 25 \\ 0 & 0 & 1 & 4 & 9 & 16 \\ 0 & 0 & 0 & 1 & 4 & 9 \\ 0 & 0 & 0 & 0 & 1 & 4 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

30.4.

$$X = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

8. YAZILAR

8.1. Yazıların təsviri və emalı

Yazılar bir identifikatora malik qeyd olunmuş sayda müxtəlif tip komponentdən ibarət olurlar. Yazının hər bir komponenti **sahə** adlanır və özlüyündə unikal ada malik olur. Yazılar **Record** xidməti sözlə aşağıdakı formatda təsvir olunurlar:

<İdentifikator> = Record <Sahələrin siyahısı> End;

Burada **<İdentifikator>** yazının identifikatorudur. **<Sahələrin siyahısı>** bölməsində tipləri verilməklə sahələr göstərilir. Aşağıdakı misalda hərbi qulluğa çağırılmış çağırışçılar haqqında əsas məlumatlar yazı şəklində təsvir olunmuşdur:

Const Say=500;

Type THerbi = Record

Il:

Word;

Kvartal:

Byte;

Fam, Ad:

String[15];

Rayon:

String[10];

Tehsil:

(ali, orta, natamam_ali);

Tevellud:

Record

TGun, TAY:

Byte;

TII:

Word;

End;

End;

Var

Herbi: Array [1..Say] of THerbi;

Yazının identifikatoru "Herbi" qəbul edilmişdir və ona **Il** – çağırış ili, **Kvartal** – çağırış ilindəki kvartal, **Fam** – çağırışçının familiyası, **Ad** – çağırışçının adı, **Rayon** – çağırışçının yaşadığı ərazi bölgüsü, **Tehsil** – çağırışçının təhsili və **Tevellud** – çağırışçının təvəllüdü sahələri daxil edilmişdir. **Il** sahəsi **Word**, **Kvartal** sahəsi **Byte**, **Fam**, **Ad** və **Rayon** sahələri **String**, **Tehsil** sahəsi sadalanan tiplidir. **Tevellud** sahəsi özü yazı şəklində təsvir olunmuşdur və **TGun**, **TAY**, **TII** sahələrindən ibarətdir. Bu misaldan görünür ki, yazılarda iç-içəlik xassəsi vardır, yəni yazını təşkil edən sahə özlüyündə yazı ola bilər.

Bir yazı daxilində sahə adları unikal olmalıdır, yəni bu sahələrin adlarında təkrarlanmaya yol verilmir. Lakin, iç-içə yerləşən yazıların müxtəlif iç-içəlik səviyyəsində eyni adlardan istifadə etmək olar. Məsələn,

Type TA = Record

A: Byte; B: Real;

C: Record

A: String[10];

D: Word;

End;

End;

Massivlərdə olduğu kimi, eyni tipli yazılar üçün də mənsub etmə əməliyyatı aparmaq olar, yəni a və b eyni tipli yazılardırsa, onda a:=b operatoru icra olunandır və bu operatorun icrasından sonra b dəyişəninin sahələri a dəyişəninin uyğun sahələrinə mənsub edilir. Yazıların müqayisəsi isə yalnız ayrı-ayrılıqda uyğun sahələr üzrə aparıla bilər.

Yazının hər hansı sahəsinə müraciət etmək üçün tərkibli addan istifadə olunur. Məsələn,

Herbi[15].Fam:='Aliyev';

Herbi[15].Rayon:='Yasamal';

Herbi[15].Kvartal:=4;

Herbi[15].Tehsil:=orta;

Herbi[15].Tevellud.TAy:=11;

Yazıların sahələrinə müraciəti sadələşdirmək məqsədilə **With .. Do** qoşma operatorundan istifadə olunur. **With .. Do** qoşma operatorunun yazılış formatı aşağıdakı kimidir:

With <Dəyişən> Do <Operator>;

Yazılışda <Dəyişən> - yazı tipli dəyişənin adıdır. Ada iç-içə yerləşən sahələrin siyahısı da əlavə edilə bilər. <Operator> - Turbo Pascal-in ixtiyari operatorudur. Məsələn,

With Herbi[1] Do Ad:='Kenan';

With Herbi[1] Do Rayon:='Yasamal';

With Herbi[1].Tevellud Do TGun:=16;

Sonuncu yazılış aşağıdakı yazılışlarla ekvivalentdir:

With Herbi[1] Do With Tevellud Do TGun:=16;

With Herbi[1],Tevellud Do TGun:=16;

Herbi[1].Tevellud.TGun:=16;

With .. Do operatoru eyni yazının bir neçə sahəsinə, və ya eyni yazıya dəfələrlə müraciət etdikdə əlverişli hesab edilir. Məsələn,

With Herbi[4] Do Begin Fam:='Cafarov';

Rayon:='Nizami';

Kvartal:=3;

Tehsil:=ali;

Tevellud.Til:=1972;

End;

Turbo Pascal-da yazıların təsvirində variant hissədən də istifadə etmək olar. Variant hissə **Case .. Of** cümləsi ilə bir neçə variantdan təşkil

olunur. Qeyd etmək lazımdır ki, yazıda yalnız bir sayda variant cümləsindən istifadə oluna bilər və o, yalnız yazının sonunda göstərilməlidir. Bu səbəbdən də variantların sonunda uyğun operatorun sonundakı End yazılır. Lakin, yazıda istifadə olunmuş Case .. Of cümləsinin tərkibində digər Case .. Of cümləsindən istifadə etməyə yol verilir, yəni Case .. Of cümlələri iç-içə yerləşdirilə bilər. Variantdan istifadə üsulunun əhəmiyyətli xüsusiyyəti ondadır ki, hər bir variant üçün ayrıca yaddaş sahəsi ayrılır, başqa sözlə bütün variantlar üçün ayrılmış ümumi yaddaş sahəsini yalnız tələbə görə seçilmiş variant zəbt edir. Ümumi qəbul edilmiş yaddaş sahəsinin uzunluğu variantlardan hər birinin tələb etdiyi yaddaş sahələrinin ən böyüyünün uzunluğuna bərabər götürülür. Aşağıda baxılan çalışmada tərkibində Case .. Of cümləsi olan yazının təsviri və emalı üsulu nəzərdən keçirilmişdir.

Çalışma 8.1 (Tələbənin məşguliyyəti). Verilənlərin məşindəxili təsvir formatı üçün yazıdan istifadə edərək, tələbənin məşguliyyəti haqqında məlumatları daxil etməli.

Həlli. Məsələnin həlində istifadə olunan verilənlərin təsviri üçün yazı formatı seçilmişdir. Tərtib edilmiş proqramda tələbənin фамилиясы (Fam), adı (Ad), qrup (Qrup) və kurs (Kurs) nömrələri daxil edilir. Nəzərə alınmışdır ki, tələbənin dərslərdən sonrakı məşguliyyəti (Mesh) müxtəlif ola bilər və bu müxtəliflik üçün 3 növ qəbul edilmişdir: idman, musiqi və qiraət. Proqramın icrası zamanı tələbənin məşguliyyəti idmanla bağlıdırsa, bu idman növü (IdmanNov) və tələbənin idman dərəcəsi (Derece), musiqi ilə bağlıdırsa, tələbənin hansı musiqi alətində çala bilməsi (MusiqiAlet), qiraətlə (Qiraet) bağlıdırsa, tələbənin ən sevimli kitabı (Kitab) və kitabın müəllifi (Muellif) sorğu edilir. Proqram mətninin listinqi aşağıda, icrasından alınan nəticələr isə, şəkil 8.1-də göstərilmişdir.



Program Meshqulluq;

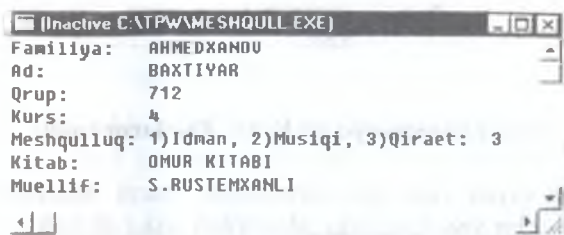
Uses WinCrt;

Type TTelebe = Record

Fam,Ad: String[15]

Qrup, Kurs: Byte;

Case Mesh: Byte Of



Şəkil 8.1. "Meshqulluq" programının icrasından alınan nəticələr

```

0:      (IdmanNov:      String[10];
        Derece:        Byte);
1:      (MusiqiAlet:    String[10]);
2:      (Qiraet:        Record
        Kitab:         String[25];
        Muellif:       String[15]
        End);
End;

Var     Telebe: TTelebe;
Begin
    With Telebe Do Begin
        Write ('Familiya: '); Readln(Fam);
        Write ('Ad: '); Readln(Ad);
        Write ('Qrup: '); Readln(Qrup);
        Write ('Kurs: '); Readln(Kurs);
        Repeat
            Write ('Meshqulluq: 1)Idman, 2)Musiqi, 3)Qiraet: ');
            Readln (Mesh);
        Until (Mesh=1) Or (Mesh=2) Or (Mesh=3);
        Case Mesh Of
            1: Begin
                Write ('Idman novu: '); Readln (IdmanNov);
                Write ('Derece: '); Readln (Derece);
                End;
            2: Begin
                Write ('MusiqiAlet: '); Readln (MusiqiAlet);
                End;
            3: With Qiraet Do Begin
                Write ('Kitab: '); Readln (Kitab);
                Write('Muellif: '); Readln (Muellif);
                End;
        End;
    End;
End;

```

End;
End;
End.

8.2. Laboratoriya işi № 10. Yazıların emalı

1. 10 sayda yazı tipli elementdən ibarət birölçülü X massivi verilmişdir. Hər bir yazı aşağıdakı sahələrdən təşkil olunub:

- Şagirdin soyadı;
- Şagirdin adı;
- Sınıfın nömrəsi;
- Biologiya fənni üzrə illik qiymət;
- Riyaziyyat fənni üzrə illik qiymət;
- Fizika fənni üzrə illik qiymət;
- İngilis dili fənni üzrə illik qiymət.

Şagirdlərin soyadı, adı və sinfin nömrəsi klaviaturadan daxil edilir.

Sınıfın nömrəsini ilk altı yazı üçün 10^a , sonrakı yazılar üçün isə 10^b qəbul edin. Fənlər üzrə şagirdlərin qiymətlərini Random funksiyasından istifadə etməklə seçin.

1.1. Bütün fənlərdən «5» qiymət alan şagirdlərin siyahısını tərtib edin.

1.2. Riyaziyyat fənnindən «4» və ya «5» qiymət alan şagirdlərin sayını tapın.

1.3. İngilis dili fənnindən «3» qiymət alan şagirdlərin sayını tapın və siyahısını tərtib edin.

1.4. 10^b sinfində təhsil alan şagirdlərin siyahısını tərtib edin.

1.5. Yazıdakı verilənlərin hansı sinifləri əhatə etdiyini təyin edin.

1.6. 10^a sinfində oxuyan şagirdlərin bütün 10-cu siniflərdə oxuyan şagirdlərin neçə faizini təşkil etdiyini təyin edin.

1.7. 10^a və 10^b siniflərində ayrılıqda neçə şagird «2»-siz qiymətlərlə oxuyur.

1.8. «2» qiymət alan şagirdlər bütün onuncuların neçə faizini təşkil edir.

1.9. Riyaziyyat fənnindən «5» qiymət alıb, ingilis dilindən «2» qiymət almayan şagirdlərin siyahısını tərtib edin.

1.10. Aşağıdakı cədvəli formalaşdırın:

Nö	Sınıf	«5» alanlar	«5» və «4» alanlar	Yalnız «3» alanlar	«2» alanlar
...	...	...	...	...	...

2. Ali məktəbin «İnformatika» fakültəsi tələbələrinin bakalavr dərəcəsi almaq üçün 4 il ərzində əldə etdiyi tədris göstəriciləri aşağıdakı cədvəldə verilmişdir:

Tələbə	Qrup	Kurs	Semestr	Fənn	Qiymət
...	...	1	1	İnformatika	...
				Riyaziyyat	...
				Tarix	...
			2	Əməliyyat sistemləri	...
				Proqramlaşdırma	...
				İngilis dili	...
		2	3	Informasiya sistemləri	...
				Kompüter texnologiyası	...
				Fəlsəfə	...
			4	...	...
		3	5	...	...
			6	...	...
		4	7	...	...
			8	...	...
...	...	...	...	...	...

Bu verilənlər yazı tipli birölçülü massivin elementləridir. Massivin 40 sayda elementi vardır. Bütün elementlərin qiymətləri Random funksiyası vasitəsilə daxil edilir. «Tələbə» sahəsi 2 böyük latın hərfi ilə ifadə olunur, məsələn, M.E. Aşağıdakı sorğuların emalı üçün proqram tərtib edin.

2.1. Klaviaturadan daxil edilmiş tələbənin soyadına və adına görə onun fənlər üzrə yekun orta qiymətini tapın.

2.2. «Fəlsəfə» fənni üzrə bütün tələbələrin orta qiymətini tapın.

2.3. Hansı fəndən tələbələrin qiymətləri «4» və ya «5»-dir.

2.4. Klaviaturadan daxil edilmiş qrup nömrəsinə görə fənlər üzrə yekun orta qiymətini tapın.

2.5. Massivə daxil edilmiş qruplardan hansının orta qiyməti ən yüksəkdir?

2.6. Hansı qrup tələbələri «İnformatika» fənnindən daha yüksək orta qiymətə malikdir?

2.7. Hansı fəndən tələbələrin orta qiyməti ən aşağıdır?

2.8. Hansı fəndən tələbələr yalnız «3» qiymət almışlar?

2.9. «İngilis dili» fənnindən tələbələrin qiymətlərini faizlə ifadə edin.

2.10. Aşağıdakı cədvəli formalaşdırın:

Fənlər	Kurslar üzrə orta qiymət				Yekun orta qiymət
	1	2	3	4	

2.11. Aşağıdakı cədvəli formalaşdırın:

Qruplar	Kurslar üzrə orta qiymət				Yekun orta qiymət
	1	2	3	4	

2.12. Aşağıdakı cədvəli formalaşdırın:

Fənlər	Kurslar üzrə qiymətlərin sayı		
	«3»	«4»	«5»

2.13. Aşağıdakı cədvəli formalaşdırın:

Qruplar	Kurslar üzrə qiymətlərin sayı		
	«3»	«4»	«5»

9. ÇOXLUQLAR

9.1. Çoxluqların təsviri və emalı

Bir-biri ilə məntiqi əlaqələndirilmiş bircins obyektlər **çoxluq** əmələ gətirir. Obyektlər arasındakı əlaqələr proqramçı tərəfindən yaradılır və Turbo Pascal bu əlaqələrə nəzarət etmir. Çoxluğun elementlərinin sayı 256-dan çox ola bilməz. Ola bilər ki, çoxluğun heç bir elementi olmasın. Belə halda çoxluq boş çoxluq adlanır və belə işarə olunur: [].

Çoxluqları Set xidməti sözündən istifadə etməklə təsvir edirlər:
<İdentifikator> = Set Of <Baza tipi>;

Bu yazılışda <İdentifikasi-
 tor> çoxluğun adını ifa-
 də edən düzgün identifi-
 katorudur. <Baza tipi>
 kimi Word, Integer və
 LongInt tiplərindən sa-
 vayı istənilən nizamlı tiptən istifadə etmək olar. Aşağıda çoxluqlara
 misallar göstərilmişdir:

Type

**TAd = Set Of (Cemil,Fereh,Farid,Faxri,Gunel,Kamal,Konul,
 Necef,Nermin,Rumiye,Shirin,Surayya,Vuqar);**
TByte = Set Of Byte;

Var

a, b, c, d: TAd; x, y, z, u: TByte;

a:=[Fereh, Kamal, Farid, Necef, Cemil, Vuqar, Nermin, Shirin];

b:=[Faxri, Konul, Shirin, Gunel, Rumiye, Necef];

c:=[Necef, Surayya, Fereh];

x:=[4..6, 45..47, 88];

y:=[5, 6, 46, 87..89];

z:=[87, 6, 88, 5, 89, 46];

Çoxluqlar üzərində aşağıda göstərilən əməliyyatlar təyin olunmuşdur:

- **Kəsişmə.** İki çoxluğun kəsişməsindən alınan çoxluq, bu çoxluqların eyni elementlərindən təşkil olunur. Kəsişmə əməliyyatı "*" simvolu ilə göstərilir. Məsələn:

d:=a*b; {Nəticə: d=[Necef, Shirin]}

d:=a*c; {Nəticə: d=[Necef, Fereh]}

d:=b*c; {Nəticə: d=[Necef]}



$u = x * y$; {Nəticə: $u = [5, 6, 46, 88]$ }

- **Birləşmə**. İki çoxluğun birləşməsindən alınan çoxluq, bu çoxluqlardan birinin bütün və ikinci birincidə olmayan elementlərindən təşkil olunur. Birləşmə əməliyyatı "+" simvolu ilə göstərilir. Məsələn,

$d = b + c$; {Nəticə: $d = [\text{Fereh, Faxri, Gunel, Konul, Necef, Shirin, Rumiye, Surayya}]$ }

$u = x + y$; {Nəticə: $u = [4, 5, 6, 45, 46, 47, 87, 88, 89]$ }

- **Çıxma**. İki çoxluğun fərqiindən alınan çoxluq, birinci çoxluğun ikinci çoxluqda olmayan elementlərindən təşkil olunur. Çıxma əməliyyatı "-" simvolu ilə göstərilir. Məsələn,

$d = b - a$; {Nəticə: $d = [\text{Faxri, Konul, Gunel, Rumiye}]$ }

$d = c - a$; {Nəticə: $d = [\text{Surayya}]$ }

$u = x - y$; {Nəticə: $u = [4, 45, 47]$ }

$u = y - x$; {Nəticə: $u = [87, 89]$ }

Çoxluqlar üzərində aparılan əməliyyatlar zamanı bu çoxluqlarda elementlərin sayı dəyişə bilər. Məhz bu xüsusiyyət onları massivlərdən və yazılardan fərqləndirir. Yerləşmə ardıcılığı nəzərə alınmadan, eyni elementlərdən ibarət çoxluqlar ekvivalent sayılır. Məsələn, $m = [\text{Alma, Limon}]$, $n = [\text{Limon, Alma}]$ olarsa, onda $m = n$.

Çoxluqlar üzərində aşağıdakı nisbət əlaqələri mümkündür:

"=": iki çoxluq ekvivalentdirsə, True qaytarılır;

"<>": iki çoxluq ekvivalent deyilsə, True qaytarılır;

"<=": birinci çoxluğun bütün elementləri ikincidə varsa, birinci çoxluq ikinci çoxluğa daxildir, yəni altçoxluğudur. Bu zaman nəticə True qiymətini alır;

">=": ikinci çoxluğun bütün elementləri birincidə varsa, ikinci çoxluq birinci çoxluğa daxildir, yəni altçoxluğudur. Bu zaman nəticə True qiymətini alır.

Elementin ədədi çoxluğa mənsubiyyəti "in" xidməti sözü ilə yoxlanıla bilər. Məsələn,

46 in y - ifadə True qiymətini alır;

90 in y - ifadə False qiymətini alır;

Nernin in a - ifadə True qiymətini alır;

Rumiye in a - ifadə False qiymətini alır.

Çoxluqlarla işləyərkən, əməliyyatların aşağıda göstərilən üstünlük dərəcələrini nəzərə almaq lazımdır:



- 1) * ;
- 2) +, -
- 3) in, =, <>, <=, <, >=, >.

Məsələn,

...

```
x:=|4..6|; y:=|5..7|; z:=|3..5|;
u:= x+y*z;           {Nəticə: u=|4, 5, 6|}
u:=(x+y)*z;          {Nəticə: u=|4, 5|}
```

Çoxluqların yazılışında Byte tipli dəyişəndən də istifadə etmək olar. Məsələn,

```
Var x, y: Set Of Byte; k: Byte; t: Boolean;
```

...

```
k:=3; x:=|4,5,6|; y:=|k+1,k+2,k+3|;
If x=y then t:=True else t:=False; {Nəticə: t=True}
```

N sayda elementdən ibarət çoxluğun 2^N sayda altçoxluğu vardır, yəni 2 sayda elementi olan çoxluq $2^2=4$ sayda altçoxluğa, 3 sayda elementi olan çoxluq $2^3=8$ sayda altçoxluğa, 4 sayda elementi olan çoxluq $2^4=16$ sayda altçoxluğa və s. malikdir. Məsələn, aşağıda göstərilən çoxluqlar $a=[5, 9, 14]$ çoxluğunun altçoxluqlarıdır:

```
a1=| |, a2=|5|, a3=|9|, a4=|14|, a5=|5, 9|,
a6=|5, 14|, a7=|9, 14|, a8=|5, 9, 14|.
```

Çalışma 9.1. Aşağıda verilmiş proqramı kompüterdə icra etməyə tələsməyin! Onu təhlil edib, nəticəni tapmağa çalışın. Cavab 5.3 sayılı çalışmanın adının ilk sözü ilə eynidir.

```
Program CavabTap;
```

```
Uses WinCrt;
```

```
Type TChar = Set Of Char;
```

```
Var x, y, z, t, d: TChar; i: Byte;
```

```
Begin t:=|'A'..'D', 'X', 'Y'|; x:=|'A'..'C', 'E', 'O', 'S', 'T', 'X', 'Y'|;
```

```
y:=|'D', 'E', 'J', 'T'|; z:=|'A', 'B', 'E', 'Y'|;
```

```
d:=t*(y-x*z)+(x-t)-z;
```

```
For i:=65 to 90 Do If (Chr(i) in d) then Write(Chr(i));
```

```
End.
```

9.2. Laboratoriya işi № 11. Çoxluqların emalı

1. X çoxluğu 3-ə, Y çoxluğu isə 5-ə bölünən ikirəqəmli tam ədədlərdən təşkil olunub. Bu çoxluqların kəsişməsindən, birləşməsindən və fərqindən alınan çoxluğu təyin edin.

2. X çoxluğu 100-ə, Y çoxluğu isə 25-ə bölünən üçrəqəmli tam ədədlərdən təşkil olunub. Bu çoxluqların kəsişməsindən, birləşməsindən və fərqiindən alınan çoxluğu təyin edin.

3. $X = \{ 'A', 'L', 'C', 'R', 'U', 'W', 'Q', 'D' \}$ çoxluğu verilmişdir. Klaviaturadan dövrü olaraq 15 sayda ixtiyari simvol daxil edin və bu simvolların X çoxluğuna mənsubiyyətini təyin edin.

4. $X = \{ 'T', 'N', 'V', 'K', 'M', 'G', 'E', 'P' \}$ çoxluğu verilmişdir. $s = \text{'NIZAMI GENCEVI'}$ sətirinin neçə simvolu X çoxluğunun elementi-dir.

5. $P = \{ 'A', 'M', 'Z', 'L' \}$ və $Q = \{ 'M', 'N', 'G', 'A', 'R', 'L' \}$ çoxluqları verilmişdir. $s = \text{'NIZAMI'}$ sətirinin hansı simvollarının P və Q çoxluqlarının kəsişməsinə, birləşməsinə və fərqiinə daxil olduğunu təyin edin.

6. M çoxluğu (10; 20) intervalına, N çoxluğu isə [15; 25] parçasına daxil olan tam ədədlərdən təşkil olunub. M və N çoxluqlarının kəsişməsindən, birləşməsindən və fərqiindən alınan çoxluqları tapın. Bu çoxluqlara hansı intervallar uyğun gəlir.

7. n natural ədədinin 1-dən 15-dək qiymətlərində hesablanmış $n^2 + 2n + 2$ ifadəsinin qiymətləri A çoxluğunun elementləridir. $B = \{ 4, 26, 33, 22, 82, 73, 145, 101, 10, 151 \}$ çoxluğu verilmişdir. A və B çoxluqlarının kəsişməsindən, birləşməsindən və fərqiindən alınan çoxluqları tapın.

8. $X = \{ \text{Baki, Sumqayıt, Qazax, Sabirabad} \}$, $Y = \{ \text{Gəncə, Yevlax, Ordubad, Sumqayıt, Qazax} \}$, $Z = \{ \text{Baki, Ordubad, Sabirabad} \}$ çoxluqları verilmişdir. $F = (X + Y * Z) - (X * Y)$ çoxluğunun elementlərini təyin edin.

9. $A = \{ \text{Almaz, Gulare, Emin} \}$, $B = \{ \text{Adil, Esmira, Mehriban, Gunel} \}$, $C = \{ \text{Almaz, Esmira, Gunel} \}$ çoxluqları verilmişdir. $X = (A - B * C) + (A * C + B) * C$ çoxluğunun elementlərini təyin edin.

10. $A = \{ 5, 6, 7, 8 \}$, $B = \{ 3, 4, 5, 6 \}$, $C = \{ 8, 9 \}$ olarsa, aşağıdakı eyniliklərin doğruluğunu təyin edin. Eynilik doğrudursa, ekrana TRUE, əks halda isə FALSE məlumatını çıxarın.

10.1.	$A * (B + C) = (A * B) + (A * C)$	10.6.	$A * B - C = A * (B - C)$
10.2.	$A + (B * C) = (A + B) * (A + C)$	10.7.	$A + B * C = A + (B * C)$
10.3.	$A * (B * C) = (A * B) + (A * C)$	10.8.	$A * (B * C) = A * B * C$
10.4.	$A + (B + C) = (A + B) + C$	10.9.	$A - (B * C) = (A - B) * C$
10.5.	$A + A * C + B * C + B = (A + B) * C$	10.10.	$A * (B - C) = A * B - A * C$

10. FAYLLAR

10.1. Faylların növləri

Əvvəlki bölmələrdə təsvir olunan dəyişənlər istifadə olunduqları proqramın icrası zamanı əməli yaddaşda yerləşdirilir və müəyyən qiymətlərə malik olurlar. Proqramın icrası başa çatdıqdan sonra bu dəyişənlər əməli yaddaşdan ləğv edirlər, onların qiymətləri isə yadda saxlanılmır, itirilir. Lakin, bir çox hallarda, əvvəlki proqramların icrasından alınan nəticələr sonradan digər proqramlarla verilənlərin emalı üçün lazım gəlir. Belə hallarda, nəticələri fərdi kompüterlərin xarici yaddaşlarında - sərt və ya çevik maqnit disklərində saxlamaq zəruriyyəti qarşıya çıxır. Verilənlər xarici yaddaşda **fayllarda** yadda saxlanırlar. Fayllarda müxtəlif formaya malik verilənləri, o cümlədən mətnləri, təsvirləri, cədvəlləri, siyahıları və s. yadda saxlamaq olar. Fayllar təşkil olunma üsuluna görə 3 müxtəlif növə malikdirlər: **mətn** faylları, **tipləşdirilmiş** və **tipləşdirilməmiş** fayllar.

10.2. Mətn faylları

Mətn faylları müxtəlif uzunluqlu simvol sətirlərindən təşkil olunur və hər bir sətir "karetkanın qaytarılması" funksiyasını daşıyan 13 sayılı ASCII kodu ilə başa çatır. Faylın sonunda **"End Of File"** ingilis sözündən götürülmüş **EOF** əlaməti qoyulur. İstifadəçilər öz tələblərinə görə yeni mətn faylları yaratmaq, mövcud fayllardakı verilənlər üzərində dəyişikliklər aparmaq, fayllardakı verilənləri oxumaq, fayla yazılar əlavə etmək və s. imkanlara malikdirlər. Lakin, aparılan əməliyyatların müxtəlifliyinə baxmayaraq, fayllarla işləyən istifadəçi iş ərafəsində aşağıdakı qəbul edilmiş ardıcılığı pozmamalıdır:

- proqram ilə faylı əlaqələndirmək;
- oxunma və ya yazma rejimlərinin birində faylı aktivləşdirmək;
- fayldan verilənlərin oxunmasını və ya verilənlərin fayla yazılmasını təşkil etmək;
- faylı bağlamaq.

Turbo Pascal-da mətn faylları ilə aparılan işi **System** modulu təşkil edir. Bu modul verilənlərin daxiletmə və xaricetmə prosesləri, sətirlər üzərində müxtəlif əməliyyatların aparılması və s. ilə əlaqədar sistem funksiyaları və prosedurlarını özündə cəmləşdirir. System modulu

proqramla avtomatik olaraq əlaqələndirilir və onun Uses operatorunda təsvir olunması tələb olunmur.

Mətn fayllarında verilənlərin axtarışı üçün yalnız ardıcıl müraciət üsulundan istifadə etmək olar, yəni istifadəçi mətnə daxil olan hər hansı sətiri emal etmək istəyirsə, o, bu sətirdən əvvəlki bütün sətrləri bir-bir addamalıdır. Ardıcıl müraciət zamanı istifadəçinin proqram vasitəsilə verilənlərin fayldan bir qismini oxumaq, və ya faylın hər hansı bir hissəsinə yazı əlavə etmək imkanı olmur.

Mətn fayllarına verilənlərin əlavə edilməsi zamanı nəzərə almaq lazımdır ki, tipi simvol tipli olmayan bütün verilənlər simvol tipinə çevrilməlidirlər. Proqramı diskdəki faylla əlaqələndirmək üçün fayl dəyişənindən istifadə olunur və bu dəyişən **Text** tipi ilə təsvir edilir. Məsələn,

```
Program Metn;  
Var Fd: Text;
```

...

Burada, **Fd** – fayl dəyişənidir. Fayl dəyişəninin adı identifikatora qoyulan tələblərə cavab verilməsi şərti ilə istifadəçi tərəfindən ixtiyari seçilə bilər.

Mətn faylları ilə işləyən bəzi funksiya və prosedurları, eləcə də faylları özlərində saxlayan qovluqlarla işləmək üçün istifadə olunan prosedurları nəzərdən keçirək:

- **Assign(<Fayl dəyişəni>: Text, <Faylın adı>: String);**

Prosedurun icrası zamanı, verilənlərin sonrakı emalı prosesini həyata keçirmək üçün fayl dəyişəni və mətn faylının adı bir-biri ilə əlaqələndirilir. Məsələn,

```
Var Fd: Text;  
Ad: String[12];
```

...

```
Ad:='Metn.txt';  
Assign(Fd, Ad);
```

və ya,

```
Var Fd: Text;  
Ad: String;
```

...

```
Readln(Ad);  
Assign(Fd, Ad);
```

- **Rewrite(<Fayl dəyişəni>: Text);**

Prosedur icra olunarkən Assign prosedurundakı fayl dəyişəni ilə əlaqələndirilmiş, uyğun identifikatora malik olan yeni fayl yaradılır və bu

və ya,

```
Var Fd: Text;  
...  
Assign(Fd, 'Metn.txt');
```

fayl oxunma və yazma rejimində aktivləşdirilir. Əgər göstərilən adda fayl diskdə əvvəlcədən mövcud isə, onda onun tərkibi silinir və fayl yenidən aktivləşdirilir. Məsələn,

```
Var Fd: Text;
```

```
***
```

```
Assign(Fd,'Metn.txt');
```

```
Rewrite(Fd);
```

```
- Reset(<Fayl dəyişəni>: Text);
```

Prosedur icra olunarkən **Assign** prosedurundakı fayl dəyişəni ilə əlaqələndirilmiş, uyğun identifikatora malik olan mövcud fayl yalnız oxunma rejimində aktivləşdirilir. Əgər fayl mövcud deyilsə və onu aktivləşdirmək cəhdi göstərilirsə, onda proqramın icrası səhv nəticə ilə başa çatır. Məsələn,

```
Var Fd: Text;
```

```
***
```

```
Assign(Fd,'Mtn.txt');
```

```
Reset(Fd); {'Mtn.txt' faylı diskdə mövcud deyil}
```

```
{ERROR – Səhv haqqında məlumat}
```

Belə hallarda, baş verən qəzalı vəziyyəti ləğv etmək və proqramın icrasını davam etdirməklə vəziyyətdən çıxmaq üçün kompilyatorun **{SI-}** direktivasından istifadə etmək olar. Məsələn,

```
Var Fd: Text;
```

```
{SI-}
```

```
***
```

```
Assign(Fd,'Mtn.txt');
```

```
Reset(Fd); {'Mtn.txt' diskdə mövcud deyil}
```

```
{Səhv haqqında məlumat verilmir və proqramın icrası davam etdirilir}.
```

```
- Append(<Fayl dəyişəni>: Text);
```

Prosedur mövcud mətn faylına yeni yazılar əlavə etmək üçün onu aktivləşdirir və icra olunarkən fayl göstəricisi faylın sonunda qeyd olunur. Sonradan fayla daxil edilən bütün yazılar fayla yalnız onun sonundan başlayaraq əlavə ediləcək. Append proseduru ancaq mətn fayllarına aid edilə bilər.

```
Var Fd: Text;
```

```
***
```

```
Assign(Fd,'Metn.txt');
```

```
Append(Fd);
```

```
- Close(<Fayl dəyişəni>: Text);
```

Prosedur icra olunarkən **Assign** prosedurundakı fayl dəyişəni ilə əlaqələndirilmiş aktiv fayl bağlanır. Məsələn,

```
Var Fd: Text;
```

```
***  
Assign(Fd,'Metn.txt');  
Rewrite(Fd);
```

```
***  
Close(Fd);
```

Əgər fayl diskdə mövcud deyilsə, və ya passivsə, onda proqramın icrası səhv nəticə ilə başa çatır. Qəzalı vəziyyəti ləğv etmək və proqramın icrasını davam etdirmək üçün kompilyatorun {\$I-} direktivasından istifadə etmək olar.

Qeyd etmək lazımdır ki, **Close** proseduru aktiv faylı bağlasa da, faylın əvvəllər **Assign** prosedurundakı fayl dəyişəni ilə əlaqəsini saxlayır və fayl dəyişəni ilə əlaqələndirməni yenidən yaratmadan faylı təkrar aktivləşdirmək olar.

- **Eof(<Fayl dəyişəni>: Text): Boolean;**

Funksiya məntiqi tipli olub, faylın sonunu aşkar etmək üçün istifadə olunur. İcra olunarkən faylın sonudursa, funksiya **True**, sonu deyilsə **False** qiymətini alır. **Assign** prosedurundakı fayl dəyişəni ilə əlaqələndirilmiş aktiv mətn faylının yazılarının oxunması iterativ olaraq faylın sonunadək yerinə yetirilir. Bu məqsədlə **Eof** – "Faylın sonudur", və ya **Not Eof** – "Faylın sonu deyil" funksiyalarından istifadə etmək olar. Çalışma 10.2 və 10.6-da funksiyanın tətbiqinə misallar göstərilmişdir.

- **Eoln(<Fayl dəyişəni>: Text): Boolean;**

Funksiya məntiqi tipli olub, sətirin sonunu aşkar etmək üçün istifadə olunur. İcra olunarkən sətirin sonudursa, funksiya **True**, sonu deyilsə **False** qiymətini alır. Çalışma 10.6-da funksiyanın tətbiqinə misal göstərilmişdir.

- **SeekEoln(<Fayl dəyişəni>: Text): Boolean;**

Funksiyanın icrası zamanı **Eoln** sətirin sonu markerinə və ya ilk qiymətli simvola kimi bütün «aralıq» simvolları və «tabulyasiya» işarələri buraxılır, sətirin sonu markeri tapılıbsa **True**, əks halda **False** qiyməti qaytarılır.

- **SeekEof(<Fayl dəyişəni>: Text): Boolean;**

Funksiyanın icrası zamanı **Eof** faylın sonu markerinə və ya ilk qiymətli simvola kimi bütün «aralıq» simvolları, «tabulyasiya» işarələri və **Eoln** sətirin sonu markerləri buraxılır, faylın sonu markeri tapılıbsa **True**, əks halda **False** qiyməti qaytarılır.

Verilənlərin fayldan oxunması **Read** və ya **Readln**, fayla yazılışı isə **Write** və ya **Writeln** prosedurları vasitəsilə yerinə yetirilir. Məsələn,

```
Readln(Fd,s); və ya Read(Fd,s);
```

Writeln(Fd, s); və ya **Write(Fd, s);**
Çalışma 10.1 (Yuramazın nəsihəti). "Metn.txt" adlı mətn faylı yaratmalı və Yuramazın oğluna nəsihətini bu fayla yazmalı;

"Bəsdir ey oğul boş yerə bu elmə çalışma,

Qanın tələf oldu.

Gündüz-gecə səy eyləyibən dərsə alışma,

Canın tələf oldu.

Bu şəhərdə çoxdur görürəm elm oxuyanlar,

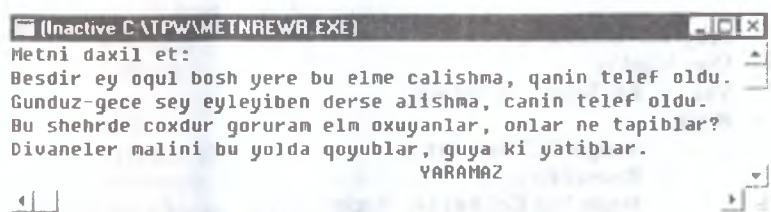
Onlar nə tapıblar?

Divanələr malını bu yolda qoyublar,

Guya ki yatıblar.

"Molla Nəsrəddin", 25 avqust 1906.

Həlli. Programın listingi aşağıda, icrası zamanı yaradılan dialog və verilənlərin daxil edilməsi prosesi şəkil 10.1-də göstərilmişdir.



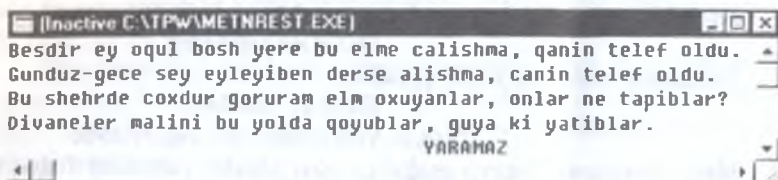
Şəkil 10.1. "MetnRewrite" programının icrasından alınan nəticələr

```

Program MetnRewrite;
Uses WinCrt;
Var   Fd: Text;
      s: String;
Begin
  Assign(Fd, 'Metn.txt');
  Rewrite(Fd);
  Writeln('Metni daxil et:');
  Readln(s);
  While s<>'' Do Begin
    Writeln(Fd, s);
    Readln(s);
  End;
  Close(Fd);
End.
```

Çalışma 10.2 (Yaramazın nəsihəti - ardı). "Metn.txt" adlı mətn faylından Yaramazın oğluna nəsihəti haqqında verilənləri oxuyaraq, ekrana çıxarmalı.

Həlli. Proqramın listingi aşağıda, icrasından alınan nəticələr şəkil 10.2-də göstərilmişdir.



Şəkil 10.2. "MetnRest" proqramının icrasından alınan nəticələr

```

Program MetnRest;
Uses WinCrt;
Var Fd: Text; s: String;
Begin
    Assign(Fd, 'Metn.txt');
    Reset(Fd);
    While Not Eof(Fd) Do Begin
        Readln(Fd, s); Writeln(s);
    End;
    Close(Fd);

```

End.

- **Rename(<Fayl dəyişəni>: Text, <Faylın yeni adı>: String);**

Prosedur icra olunarkən Assign prosedurundakı fayl dəyişəni ilə əlaqələndirilmiş faylın adı <Faylın yeni adı> ilə əvəz olunur. Qeyd etmək lazımdır ki, adı dəyişdirilən fayl bağlı olmalıdır. Məsələn,

```
Var Fd: Text;
```

...

```
Assign(Fd, 'Metn.txt');
```

```
Rename(Fd, 'MetnYeni.txt');
```

Misalda **Rename** proseduru 'Metn.txt' faylının adını 'MetnYeni.txt' adı ilə əvəz edir.

Əgər yeni adda fayl əvvəlcədən diskdə varsa, onda prosedur qəzalı başa çatır. Qəzalı vəziyyəti ləğv edib, proqramın icrasını davam etmək üçün kompilyatorun {SI-} direktivasından istifadə etmək olar. Qəzanın

qarşısını almaqdan ötrü tövsiyə olunur ki, yeni addakı faylın diskdə əvvəlcədən olması yoxlanılsın.

- Erase(<Fayl dəyişəni>: Text);

Prosedur icra olunarkən Assign prosedurundakı fayl dəyişəni ilə əlaqələndirilmiş fayl silinir. Qeyd etmək lazımdır ki, silinən fayl bağlı olmalıdır. Məsələn,

Var Fd: Text;

...

Assign(Fd, 'Metn.txt');

Erase(Fd);

Misalda Erase proseduru 'Metn.txt' faylını diskdən silir.

- GetDir(<Qurğu>: Word, <Qovluq>:String);

Prosedur cari qovluğun adını təyin edir. <Qurğu> operandında qurğunun nömrəsi göstərilir. Bu nömrə kimi 0 – susmaya görə qurğu, 1 – A diski, 2 – B diski, 3 – C diski və s. götürülür. <Qovluq> operandmda qovluğun marşrutu qaytarılır. Məsələn,

Var s: String;

t: Word;

...

t:=3;

GetDir(t, s);

Writeln(s);

<Nəticə: C:\TPW>

- ChDir(<Marşrut>: String);

Prosedurun icrasından sonra operandda göstərilən marşrutdan cari qovluq seçilir. Məsələn,

Var s, s1: String;

t: Word;

...

{ Cari qovluq: C:\TPW }

s1:='C:\ Pascal';

ChDir(s1);

Getdir(t, s);

Writeln(s);

<Nəticə: C:\ Pascal >

Fraqmentin icrasından əvvəl C:\TPW qovluğu cari idi. ChDir proseduru C:\Pascal qovluğunu cari edir.

- Flush(<Fayl dəyişəni>: Text);

Turbo Pascal-da fayllara ixtiyari müraciət daxili bufer vasitəsilə həyata keçirilir. Prosedurun icrası ilə daxili bufer təmizlənir və beləliklə də, faylda aparılan son dəyişikliklərin yadda saxlanılmasına zəmanət

verilir. Fayl **Reset** proseduru ilə «yalnız oxunma» rejimində aktivləşdirilibsə, **Flush** proseduru icra zamanı nəzərə alınmır.

- **MkDir(<Qovluq>: String);**

Prosedur icra olunarkən diskdə verilmiş marşruta uyğun qovluq yaradılır. Diskdə əvvəlcədən göstərilən adda qovluq varsa, prosedurun icrası səhv məlumatla başa çatır. Məsələn,

Var s: String;

s:='c:\Pascal';

MkDir(s);

Prosedurun icrasından sonra **C:** diskində **Pascal** qovluğu yaradılır.

- **Rmdir(<Qovluq>: String);**

Prosedur icra olunarkən diskdə verilmiş marşruta uyğun qovluq silinir. Diskdə göstərilən addakı qovluq boş deyilsə, prosedurun icrası qəzalı başa çatır. Məsələn,

Var s: String;

s:='c:\Pascal';

Rmdir(s);

Prosedurun icrasından sonra **C:** diskindəki **Pascal** qovluğu silinir.

- **IOResult**

Funksiya fayllarla işləyərkən səhvlərin aşkar edilməsi üçün tətbiq olunur. Proqramda kompilyatorun **{SI-}** direktivasından istifadə olunduğu halda fayllarla iş zamanı qəzalı nəticələr haqqında məlumat verir. Fayllarla aparılan əməliyyat normal başa çatıbsa, funksiyanın qiyməti 0 olur. Əks halda funksiya qəza halına uyğun gələn kodu qaytarır. Funksiyadan adətən fayllarla aparılan əməliyyatdan sonra məntiqi operatorlarda, onun qiymətini 0-la müqayisə etməklə istifadə olunur. Məsələn,

{SI-}

Var Fd1: Text;

Assign(Fd1,'Mtn.txt');

Reset(Fd1);

if IOResult<>0 then writeln('Mtn.txt-Faylin aktivleshmesi zamani qeza');

Çalışma 10.3 (Yaramazın nəsihəti - ardı). "Metn.txt" adlı mətn faylındakı verilənlərin şəkil 10.2-də təsvir edilmiş formatını çalışma 10.1-də tələb olunan formata gətirməli.

Həlli. Məsələnin həlli üçün yeni yaradılmış 'MetnYeni.txt' köməkçi faylından istifadə edilir. Proqramın icrası zamanı 'Metn.txt' faylı

yalnız oxunma rejimində aktivləşdirilir. 'Metn.txt' faylının ilk dörd sətirinin hər biri ayrılıqda yaddaşa oxunur və "," simvoluna nəzərən iki yeni sətərə ayrılır. Ayrılmış iki sətirdən ikincisində "boşluq" simvolundan sonrakı kiçik hərf simvolu **UpCase** funksiyası vasitəsilə böyük hərf simvoluna çevrilir. Nəticələr 'MetnYeni.txt' faylına iki sətirlə əlavə edilir. 'Metn.txt' faylının 5-ci sətirində dəyişiklik aparılmadan bu sətir olduğu kimi 'MetnYeni.txt' faylına yazılır. 'MetnYeni.txt' faylının sonuna nəsihətin götürüldüyü jurnalə istinadı göstərən verilənlər bir yazı şəklində əlavə olunur.

'MetnYeni.txt' faylı tələbə uyğun tamamilə formalaşdırıldıqdan sonra, 'Metn.txt' faylı diskdən silinir və 'MetnYeni.txt' faylının adı 'Metn.txt' adı ilə əvəz olunur. Məsələnin həlli üçün tərtib edilmiş proqramın listingi aşağıda göstərilmişdir.

Program MetnDuzelish;

Uses WinCrt;

Var Fd1, Fd2, Fd3: Text;

s1, s2: String[100];

t: Byte;

i, j: Integer;

Begin

Assign(Fd1, 'Metn.txt');

Reset(Fd1);

Assign(Fd2, 'MetnYeni.txt');

Rewrite(Fd2);

For i:=1 to 4 Do Begin

Readln(Fd1, s1);

t:=Pos(' ', s1);

s2:=Copy(s1, 1, t);

Writeln(Fd2, s2);

s2:=''; for j:=1 to 33 do s2:=Concat(s2, ' ');

s2:=s2+Copy(s1, t+1, length(s1)-t);

s2[35]:=UpCase(s2[35]);

Writeln(Fd2, s2);

End;

Readln(Fd1, s1); Writeln(Fd2, s1);

s1:=''; for i:=1 to 33 do s1:=Concat(s1, ' ');

s1:=s1+' Molla Nesreddin, 25 avqust, 1906.';

Writeln(Fd2, s1);

Close(Fd1);

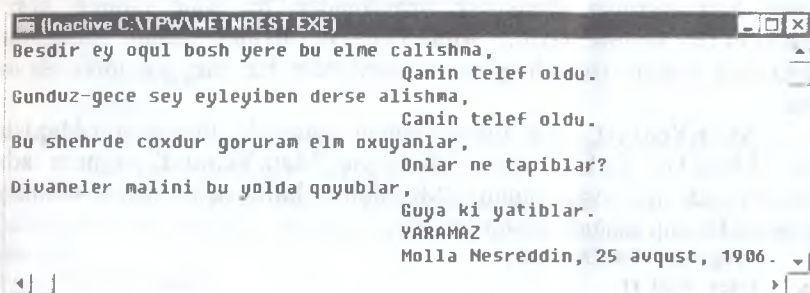
Close(Fd2);

Erase(Fd1);

Rename(Fd2, 'Metn.txt');

End.

"MetnDuzelish" programının icrasından alınan nəticələri yoxlamaq üçün, çalışma 10.2-də təsvir olunan "MetnRest" programını yenidən icra etmək olar (şəkil 10.3).



Şəkil 10.3. "MetnDuzelish" programının icrasından alınan nəticələrin "MetnRest" programı ilə ekrana çıxarılması

Çalışma 10.4 (Fayl adının yeniləşdirilməsi). Diskdə yerləşən "Metn.bak" faylının adını "Metn.txt" adı ilə əvəz etməli.

Həlli. Faylın adını dəyişdirərkən aşağıdakı qəzalı halları programda nəzərə almaq lazımdır:

- Köhnə adda faylın diskdə mövcudluğunun yoxlanılması. Bu adda fayl diskdə yoxdursa, onun adını dəyişmək prosesi heç bir məna daşımır;
- Yeni adda faylın diskdə yoxlanılması. Bu adda fayl diskdə yoxdursa, köhnə adda faylın adını yenisi ilə əvəz etmək olar. Əks halda, yəni yeni adda fayl diskdə əvvəlcədən mövcuddursa, addəyişmə prosesini həyata keçirmək üçün onu diskdən silmək lazımdır. Lakin, diskdəki bu adda fayl lazımlı ola bilər və onun diskdən silinməsindən qabaq bu haqda istifadəçiyə xəbərdarlıq vermək tövsiyə olunur.

Programın həll alqoritminin blok-sxemi şəkil 10.4-də, listingi isə aşağıda göstərilmişdir.

```
Program YeniAd;  
Uses WinCrt;  
Var FdKohne, FdYeni: Text; d: Byte;  
{SI-}  
Begin  
Assign(FdKohne, 'Metn.bak');
```

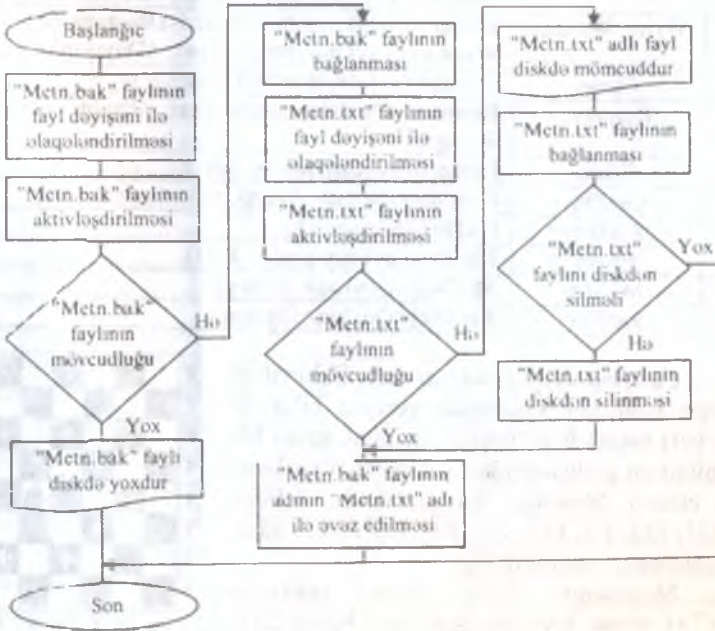
```

Reset(FdKohne);
if IOResult<>0 then      begin
    Writeln('Kohne adda fayl diskde yoxdur'); Halt;
end;

Close(FdKohne);
Assign(FdYeni,'Metn.txt');
Reset(FdYeni);
if IOResult=0 then      begin
    Writeln('Yeni adda fayl diskde evvelceden var. ');
    Repeat
        Writeln('Bu fayli silmele? 1-He, 2-Yox ');
        Read(d);
    Until (d=1) or (d=2);
    Close(FdYeni);
    if d=1 then Erase(FdYeni) else Halt;
end;
Rename(FdKohne,'Metn.txt');

```

End.



Şəkil 10.4

Cədvəl 10.1-də fayllara tətbiq olunan funksiyalar və prosedurlar göstərilmişdir.

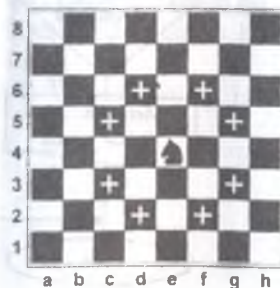
Cədvəl 10.1

Fayllara tətbiq olunan funksiyalar və prosedurlar

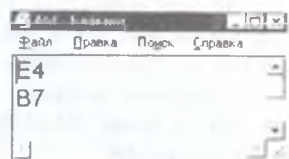
No	Adı	Təyinatı
Funksiyalar		
1	Eof	Eof(<Fayl dəyişəni>: Text): Boolean;
2	Eoln	Eoln(<Fayl dəyişəni>: Text): Boolean;
3	IOResult	IOResult
4	SeekEoln	SeekEoln(<Fayl dəyişəni>: Text): Boolean;
5	SeekEof	SeekEof(<Fayl dəyişəni>: Text): Boolean;
Prosedurlar		
6	Assign	Assign(<Fayl dəyişəni>: Text, <Faylın adı>: String);
7	Rewrite	Rewrite(<Fayl dəyişəni>: Text);
8	Reset	Reset(<Fayl dəyişəni>: Text);
9	Append	Append(<Fayl dəyişəni>: Text);
10	Read, Readln	Read(<Fayl dəyişəni>: Text, <Dəyişənlər>); Readln(<Fayl dəyişəni>: Text, <Dəyişənlər>);
11	Write, Writeln	Write(<Fayl dəyişəni>: Text, <Dəyişənlər>); Writeln(<Fayl dəyişəni>: Text, <Dəyişənlər>);
12	Close	Close(<Fayl dəyişəni>: Text);
13	Rename	Rename(<Fayl dəyişəni>: Text, <Faylın yeni adı>: String);
14	Erase	Erase(<Fayl dəyişəni>: Text);
15	GetDir	GetDir(<Qurğu>: Word, <Qovluq>:String);
16	ChDir	ChDir(<Marşrut>);
17	Flush	Flush(<Fayl dəyişəni>: Text);
18	MkDir	MkDir(<Qovluq>: String);
19	Rmdir	Rmdir(<Qovluq>: String);

Çalışma 10.5 (Şahmat atı). Şahmat atı taxtanın istənilən xanasında yerləşə bilər və onun yeri təsadüfi götürülür. Şahmat atının bütün mümkün gedişlərindən sonra onun yerlərini təyin etməli. Məsələn, E4 xanasında yerləşən atın C3, D2, F2, G3, G5, F6, D6 və C5 xanalarına gedişləri mümkündür.

Məsələnin həlli üçün verilənlər AtD.Txt mətn faylında saxlanır. Fayla daxil



edilmiş verilənlər şəkil 10.5-də göstərilmişdir. Şahmat atının taxta üzərindəki $N=2$ sayda vəziyyəti, hər biri yeni sətirdə olmaqla göstərilir. Atın vəziyyəti şahmatda qəbul edilən yazılışla qeyd olunur.



Şəkil 10.5

Şahmat atının verilmiş sayda vəziyyətləri üçün məsələnin həllindən alınan nəticələr AtX.Txt mətn faylında aşağıdakı qaydada göstərilməlidir (şəkil 10.6):

- 1-ci sətir. Şahmat atının birinci vəziyyəti;

- 2-9-cu sətirlər. Şahmat atının birinci vəziyyətinə uyğun yerləşdiyi və mümkün gedışlərini əks etdirən matris. Şahmat taxtasının xanaları 0-la, şahmat atının vəziyyəti 9-la, mümkün gedışləri isə, 1-lə təsvir edilir;

10-cu sətir. Ayrıncı boş sətir;

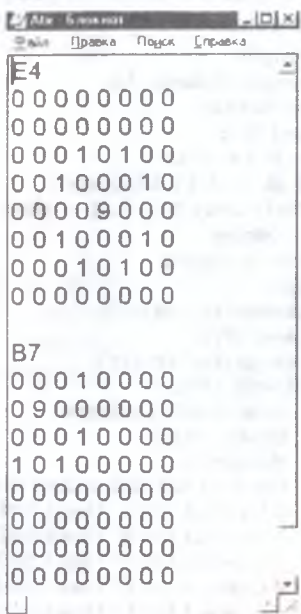
11-ci sətir. Şahmat atının ikinci vəziyyəti;

12-19-cu sətirlər. Şahmat atının ikinci vəziyyətinə uyğun yerləşdiyi və mümkün gedışlərini əks etdirən matris.

Həlli: Məsələnin həlli üçün şahmat taxtasının xanalarını uyğun sətir və sütun nömrələri ilə təyin edək. Məsələn, şahmat taxtasında E4 xanasına taxtanın 4-cü sətiri və 5-ci sütunu uyğun gəlir. Proqramda xanalar müraciət tam tipli dəyişən kimi elan edilmiş **i1** və **i2** indeksləri ilə həyata keçirilir. Taxta

üzərində yerləşən şahmat atının vəziyyətini taxtanın **k1** və **k2** indeksli xanası təyin edir. Bu indekslərin qiymətləri AtD.Txt mətn faylından götürülür. Bu məqsədlə hər iterasiyada faylın uyğun sətiri oxunur və verilən **s** dəyişəninə mənsub edilir. Sonradan **s** sətir dəyişəninənin qiymətindən **k1** və **k2** indekslərinin qiymətləri hesablanır.

Şahmat taxtası 8 sətir və 8 sütundan ibarət simvol tipli ShT massivi ilə ifadə olunur. Əvvəlcədən bu massivənin bütün elementlərinə '0' simvolu, şahmat atının yerləşdiyi xanaya massivdə uyğun gələn **k1** və **k2** indeksli elementə isə '9' simvolu mənsub edilir.



Şəkil 10.6

Şahmat atının gedişinin "I" simvoluna uyğunluğunu nəzərə alaraq, proqramda bu gedişlərin mümkünlüyü yoxlanılır. Gediş mümkün isə, ShT massivinin uyğun elementinə «1» simvolu mənsub edilir.

Proqramın icrasının sonunda alınan nəticələr məsələnin şərtinə uyğun olaraq (şəkil 10.6), AtX.Txt mətn faylına yazılır.

Proqramın icrası zamanı ekrana çıxarılan nəticələr şəkil 10.7-də, proqramın listingi isə aşağıda göstərilmişdir.

```
Program Shahmat_At;
Uses WinCrt;
Const N=2;
Var Fr,Fw: Text;
    i, id, i1, i2, k1, k2: Integer;
    ShT: array [1..8, 1..8] of char;
    s: String;
    Sh: String[40];
```

Begin

Assign (Fr, 'Atd.txt');

Reset (Fr);

Assign (Fw, 'Atx.txt');

Rewrite (Fw);

For id:=1 to N Do Begin

Readln (Fr,s);

Writeln(s);

For i1:=1 to 8 do For i2:=1 to 8 do ShT[i1,i2]:='0';

if Copy(s,1,1)='A' Then k2:=1;

if Copy(s,1,1)='B' Then k2:=2;

if Copy(s,1,1)='C' Then k2:=3;

if Copy(s,1,1)='D' Then k2:=4;

if Copy(s,1,1)='E' Then k2:=5;

if Copy(s,1,1)='F' Then k2:=6;

if Copy(s,1,1)='G' Then k2:=7;

if Copy(s,1,1)='H' Then k2:=8;

if Copy(s,2,1)='1' Then k1:=1;

if Copy(s,2,1)='2' Then k1:=2;

if Copy(s,2,1)='3' Then k1:=3;

if Copy(s,2,1)='4' Then k1:=4;

if Copy(s,2,1)='5' Then k1:=5;

if Copy(s,2,1)='6' Then k1:=6;

if Copy(s,2,1)='7' Then k1:=7;

if Copy(s,2,1)='8' Then k1:=8;

ShT[k1,k2]:='9';

If (k1+1<9) and (k1+1>0) and (k2-2>0) and (k2-2<9) then ShT[k1+1,k2-2]:='1';



Şəkil 10.7

```

If (k1+2<9) and (k1+2>0) and (k2-1>0) and (k2-1<9) then ShT[k1+2,k2-1]:='1';
If (k1+2<9) and (k1+2>0) and (k2+1>0) and (k2+1<9) then ShT[k1+2,k2+1]:='1';
If (k1+1<9) and (k1+1>0) and (k2+2>0) and (k2+2<9) then ShT[k1+1,k2+2]:='1';
If (k1-1<9) and (k1-1>0) and (k2+2>0) and (k2+2<9) then ShT[k1-1,k2+2]:='1';
If (k1-2<9) and (k1-2>0) and (k2+1>0) and (k2+1<9) then ShT[k1-2,k2+1]:='1';
If (k1-2<9) and (k1-2>0) and (k2-1>0) and (k2-1<9) then ShT[k1-2,k2-1]:='1';
If (k1-1<9) and (k1-1>0) and (k2-2>0) and (k2-2<9) then ShT[k1-1,k2-2]:='1';
Writeln (Fw,Copy(s,1,2));
For i1:=8 downto 1 do Begin
    sh:='';
    For i2:=1 to 8 do
        sh:=sh+ShT[i1,i2]+' ';
    Writeln (Fw,sh);
    Writeln (sh);
End;
Writeln (Fw,'');
End;
Close(Fr);
Close(Fw)
End.

```

Çalışma 10.6 (Sətrin sonu). Şahmat atı proqramının icrasından alınan AtX mətn faylının sətirləri içərisində boş sətrin olmasını təyin edin. Sətir boş deyilsə ekrana «F», boş isə «T» simvolunu çıxarın.

Həlli. Məsələnin həlli üçün tərtib edilmiş proqram aşağıda göstərilmişdir.

```

Program SetirSon;
Uses WinCrt;
Var Fd: Text; s: String;
Begin
    Assign (Fd, 'AtX.txt');
    Reset (Fd);
    Repeat
        if Eoln(Fd) then Write ('T') else Write ('F');
        Readln (Fd,s);
    Until Eof(Fd);
    Close (Fd);
End.
Cavab:
FFFFFFFFFTFFFFFFFFFT

```

10.3. Laboratoriya işi № 12. Mətn faylları

1. A1.TXT mətn faylı yaradın və aşağıdakı mətni fayla yazın. Faylda istifadə olunmuş durğu işarələrinin sayını təyin edin.

Simnar. Sən nankor adamsan, həkim! Mən sənə bu qədər hörmət edirəm, amma sən inadından əl çəkmirsən.

Həkim. (Zənciri göstərir). Görürəm hörmətini, xan!

Simnar. Mən o zəncirləri də qızıldan qayırdırmışam. Bəs bu hörmət deyil, nədir?

Həkim. Ancaq nə qızıl, nə də zəncir məni öz fikrimdən döndərə bilməyəcək, xan!

2. A2.TXT mətn faylı yaradın və aşağıdakı mətni fayla yazın. Faylda istifadə olunmuş sözlərin sayını təyin edin.

Xətayi, can arxına,

Əhli-ürfan arxına,

Mərifətdən su gəlib,

Tökülür can arxına.

Xətəyiyəm xəttaram,

Haq sirrinə səttaram.

Həkimlərin dərmanı,

Təbiblərə əttaram.

3. A2.TXT faylındakı şəri hər birində 4 sətir olmaqla iki hissəyə ayırın. Hissələr arasında '*' * '*' simvollarını qoymaqla yeni A4.TXT faylını yaradın.

4. A3.TXT köməkçi faylından istifadə edərək A1.TXT və A2.TXT fayllarındakı mətnlərin yerlərini dəyişin.

5. A2.TXT faylında ardıcıl olaraq tək və cüt saylı sətirləri birləşdirərək, nəticəni yeni B.TXT faylına yazın.

6. A1.TXT faylında ən çox istifadə olunan sözü tapın.

7. A1.TXT faylında durğu işarələrindən hər birinin neçə dəfə istifadə olunmasını təyin edin. Ən çox hansı durğu işarəsindən istifadə olunmuşdur.

8. A1.TXT faylını oxuyun. Mətnə daxil olan hər bir latın hərfi simvolunu əlifba sırasındakı özündən sonrakı hərflə, əlifbanın sonuncu hərfini birinci ilə əvəz edərək mətni şifrəleyin. Durğu işarələrini saxlayın. Yeni mətni A3.TXT faylına yazın. A3.TXT şifrələnmiş mətn faylında şifri açmaqla, mətni yenidən oxuyun.

9. A2.TXT faylının hər bir sətirində ən uzun sözü tapın. Bu sözlərdən təşkil olunmuş yeni B2.TXT faylı yaradın.

10. L1.TXT mətn faylını yaradın və fayla SOYAD_AD formasında 20 sayda yazı əlavə edin. L1.TXT faylını oxuyaraq, hər sətirdəki verilənləri AD_SOYAD formatına gətirməklə yeni L2.TXT faylını formalaşdırın.

11. OLKE.TXT mətn faylını yaradın və fayla hər bir sətirdə bir ad olmaqla 20 sayda ölkə adı yazın. Faylın sətirlərindəki verilənləri əlifba sırasına görə nizamlayın, nizamlanmış ölkə adlarını yeni NİZAMOL.TXT faylına yazın.

12. C: diskində AtaSoz qovluğu yaradın və bu qovluğa hər bir sətirdə bir atalar sözü olmaqla 6 sayda sətirdən ibarət AtaSoz.Txt faylını yazın. Faylın 1-ci və 4-cü, 2-ci və 5-ci, 3-cü və 6-cı sətirlərinin yerlərini dəyişməklə yeni AtaSozl.Txt faylını yaradın və həmin qovluğa yazın.

13. S1.TXT faylı 1 sətirdən ibarətdir. Bu sətirdə şahmat topunun taxta üzərindəki C4 vəziyyəti qeyd olunmuşdur. Şahmat topunun mümkün gedişlərini təyin edin. Nəticələri C:\SHAHMAT qovluğunda S2.TXT adlı mətn faylına yazın.

14. S1.TXT faylı 1 sətirdən ibarətdir. Bu sətirdə şahmat vəzirinin taxta üzərindəki F7 vəziyyəti qeyd olunmuşdur. Şahmat vəzirinin mümkün gedişlərini təyin edin. Nəticələri C:\SHAHMAT qovluğunda S2.TXT adlı mətn faylına yazın.

15. S1.TXT faylı 1 sətirdən ibarətdir. Bu sətirdə şahmat filinin taxta üzərindəki D5 vəziyyəti qeyd olunmuşdur. Şahmat filinin mümkün gedişlərini təyin edin. Nəticələri C:\SHAHMAT qovluğunda S2.TXT adlı mətn faylına yazın.

16. S1.TXT faylı 1 sətirdən ibarətdir və həmin sətirdə «B2F4» simvolları yazılmışdır. İlk iki simvol ağ şahmat atının, sonrakı iki simvol isə qara şahın taxta üzərindəki vəziyyətlərini əks etdirir. Şahmat atı bir gedişlə şah elan edə bilirmi? Əgər şah elan etmək mümkün isə bu gedişi göstərin.

17. S1.TXT faylı 2 sətirdən ibarətdir. Birinci sətirdə «C4H6», ikinci sətirdə isə «E7B4» simvolları yazılmışdır. Bu simvollar ardıcılığında ilk iki simvol ağ şahmat atının, sonrakı iki simvol isə qara vəzinin taxta üzərindəki vəziyyətlərini təyin edir. Gediş qara fiqurların isə, hansı halda qara vəzir ağ atı vura bilər?

18. S1.TXT faylı 2 sətirdən ibarətdir. Birinci sətirdə «F1B8», ikinci sətirdə isə «C3G2» simvolları yazılmışdır. Bu simvollar ardıcılığında ilk iki simvol ağ şahmat atının, sonrakı iki simvol isə qara filin taxta üzərindəki vəziyyətlərini təyin edir. Hər bir halda ağ atın hansı gedişləri təhlükəlidir və bu gedişlərdən sonra qara fil atı vura bilər? Nəticələri S2.TXT faylına uyğun sətirlərdə yazın.

ƏDƏBİYYAT

1. Kərimov S.Q., Həbibullayev S.B., İbrahimzadə T.İ. İnformatika. Bakı, 2002.
2. Məmmədov Ə.M., Səfəraliyeva F.Q. Paskal proqramlaşdırma dili. Bakı, 2001.
3. Ə.M.Məmmədov, S.A.Məmmədov. Proqramlaşdırma dilləri. Bakı, 1992.
4. Həbibullayev S.B., Rəhimova Y.Q., Rəhimova N.Ə. Paskal dilində mürəkkəb tip verilənlərin emalı. Metodiki göstəriş. Bakı, 1999.
5. Фаронов В.В. Турбо Паскаль. Начальный курс. М., 2000.
6. Турбо Паскаль 7.0. Учебное пособие. Киев, 2000.
7. Немнюгин С., Перколаб Л. Изучаем Turbo Pascal. Санкт-Петербург, 2002.
8. Петров А.В., Алексеев В.Е. и др. Вычислительная техника и программирование. М., 1990.
9. Абрамов В.Г., Трифонов Н.П., Трифонова Г.Н. Введение в язык Паскаль. М., 1988.
10. Абрамов С.А., Зима Е.В. Начала программирования на языке Паскаль. М., 1987.
11. Белецкий Я. Турбо Паскаль с графикой для персональных компьютеров. М., 1991.
12. Эрбс Х.-Э., Штольц О. Введение в программирование на языке Паскаль. М., 1989.
13. Боон К. Паскаль для всех. М., 1988.
14. Васюкова Н.Д., Тюляева В.В. Практикум по основам программирования. Язык Паскаль. М., 1991.

