«AZƏRBAYCAN HAVA YOLLARI »

QAPALI SƏHMDAR CƏMİYYƏTİ

MİLLİ AVİASİYA AKADEMİYASI

FAKÜLTƏ: **Aerokosmik**

KAFEDRA: **Aerokosmik informasiya sistemləri**

İXTİSAS: **İnformasiya texnologiyaları və sistemləri mühəndisliyi**

K U R S İ Ş İ

MÖVZU: Massivlərin emalında fayllardan istifadə

QRUP: **1444 a**

TƏLƏBƏ: **Sübhanə Quluzadə**

MÜƏLLİM: **Səlahəddin Həbibullayev**

BAKI - 2015

Mündəricat

Giriş

Fayllarla iş axında göstəricinin elanı ilə başlayır. Elan aşağıdakı formatdadır:

```
FILE *göstərici_adı;
```

Məsələn, `FILE *fp;`

FILE sözü struktur tipin standart adı olmaqla, stdio.h adlı faylla elan olunur. FILE - nin strukturundakı informasiya axınlarla işin aparılmasına kömək edir.

Növbəti addım fopen() standart funksiyanın köməyi ilə axının açılmasıdır. Bu funksiya axının göstəricisi üçün konkret qiymət qaytarır. Operatorun formatı aşağıdakı kimidir:

```
göstəricinin_adı=fopen(faylın_adı, açmaq_rejimi);
```

fopen() funksiyasının parametrləri simvol massivinin sabitləri və göstəricisi ola bilər. Məsələn,

```
fp=fopen("test.dat","r");
```

Burada test.dat –¿ diskin carikataloqunun fiziki faylının adıdır ki, bununla da fp göstəricisi axınla əlaqələndirilir. r parametri isə faylın oxunması üçün açılması rejimini göstərir. Qeyd edək ki, "faylın açılması" ifadəsini, "axının açılması" ifadəsi kimi işlətmək olar.

Turbo Pascal dilində olduğu kimi burada da, yazmaq üçün mövcud faylın açılması əvvəlki informasiyanın itməsinə səbəb olur. Əgər belə fayl mövcud deyilsə, onda o yaradılır. Oxumaq üçün yalnız mövcud fayllar açılır.

Axın ya mətn üçün, ya da ikilik mübadilə rejimləri üçün açılır.

Yuxarıdakı rejimlərin parametrləri yalnız mətn fayllarını açır. Əgər ikilik fayl göstərmək tələb olunursa, onda parametrə b hərfi əlavə olunur. Məsələn, rb və wb və ya r+b. Bəzi kompilyatorlarda mətn mübadilə rejimi t hərfi ilə işarə olunur, yəni a+t və ya rt yazılır.

Axının açılışında hər hansı səbəbdən baş verdikdə, fopen() funksiyası NULL konstant qiymətini qaytarır. Bu konstant həmçinin stdio.h faylında təyin edilib. Səhv diskdə açılan fayl olmadıqda, dinamiki yaddaş üçün yerin çatmamasında və s. baş verir. Faylın açılması üçün aşağıdakı üsul məsləhət görülür:

```
FILE *fp;
if (fp=fopen("test.dat","r"))
```

```
puts ("Fayli aca bilmirem\n")ı
return;
```

Səhv nəticəsində proqram əvvəl açılmış bütün faylların bağlanmasının yerinə yetirilməsini sona çatdırır.

Axının (faylın) proqram əvvəl açılmış bütün faylların bağlanması fclose() funksiyası ilə həyata keçirilir ki, prototipi aşağıdakı kimidir:

```
int fclose(FILE *fptr);
```

Burada fptr bağlanmalı olan axının göstəricisinin formal adını işarə edir. Əgər bağlanma əməliyyatı müvəffəqiyyətlə yerinə yetirilərsə funksiya sıfırı qaytarır. Başqa qiymət isə səhvi göstərir.

Simvolların yazılışı və oxunuşu. Simvolların axına yazılışı putc() funksiyası ilə həyata keçirilir ki, prototipi aşağıdakı kimidir:

```
int putc(int ch, FILE *fptr);
```

Əməliyyat müvəffəqiyyətlə getsə, onda yazılan simvol qaytarılır. Səhv olduqda isə EOF konstantı qaytarılır.

Axından simvolun oxunması, yəni oxumaq üçün açılması gets() funksiyası ilə həyata keçirilir ki, prototipi aşağıdakı kimidir:

```
int gets(FILE *fptr);
```

1. Mətn fayllarla işləmə

Əgər klaviaturadan çoxlu sayda verilənlər daxil olunursa, onda onları hər dəfə təkrar daxil etməsi yorucu bir iş olar. Bunun üçün diskdə yeni bir fayl yaradılıb, lazımi verilənlər ora yazılır və proqram lazımi məlumatları məhz bu fayldan oxuyur.

Fayllar **mətn** (bu tip fayllara hərfləri, rəqəmləri, mötərizələri və digər simvollar daxildir) və **binar**(bu tip fayllarda simvol cədvəlinin istənilən simvolları saxlanıla bilər) tipli olur. (Bundan sonra yalnız mətn tipli fayllara baxılır)

Proqramdan fayllara necə müraciət etmək olur?

Fayllarla iş sendviç prinsipinə əsaslanır:

"Faylı açmaq" onunla işə başlamaq, onu aktiv etmək və digər proqramların bu fayla müracətini blokadaya salmaq deməkdir. Faylı bağlayan zaman o blokadadan çıxarılır, digər proqramlar artıq onunla iş görə bilir və bütün dəyişikliklər diskdə yadda saxlanılır.

Faylla işləmək üçün **faylın göstəricisi** adlanan xüsusi dəyişəndən istifadə edilir. Bu açıq fayl haqqında bütün məlumatları özündə saxlayan yaddaş blokunun ünvanıdır. Fayl göstəricisi aşağıdakı kimi elan olunur:

```
FILE *fp;
```

Faylı açmaq üçün **fopen** funksiyasına müraciət edilir. Bu funksiya faylı açıb onun ünvanını **fp** dəyişəninə yazır. Bundan sonra fayla müraciət etmək üçün adına yox, **fp** göstəricisinə müraciət etmək lazımdır.

```
fp=fopen("c:\\data\\qq.dat","r");
```

Marşrutu göstərərkən "\" (əks sləş) simvolu həmişə cüt yazılmalıdır. "r" (fayldan oxuma) rejimindən başqa digər rejimlərdən də istifadə oluna bilər:

"r"	Faylı oxumaq üçün açmaq
"w"	Yeni fayla yazma. Əgər diskdə həmin adda fayl mövcuddursa, onda əvvəlcə o silinəcəkdir
"a"	Faylın sonuna əlavə. Əgər diskdə bu adda fayl varsa, onda yeni məlumatlar faylın sonuna əlavə olunacaqdır. Əgər fayl mövcud deyilsə, onda yeni fayl yaranacaqdır

"r+"	Yazıb/oxumaq üçün mövcud faylı açmaq
"w+"	Yazıb/oxumaq üçün yeni fayl yaratmaq (əgər həmin adda fayl varsa, onda o, yenisi ilə əvəz olunur)

Bəzən proqram faylı açа bilmir. Əgər fayl oxumaq üçün açılırsa, onda bu vəziyyət aşağıdakı hallarda ola bilər:

1. Fayl düzgün verilməyib və ya bu fayl diskdə yoxdur.
2. Fayl digər proqram tərəfində istifadə olunur və blokadaya salınıb.

Əgər fayl açılıbsa, onda bu əməliyyat aşağıdakı hallarda uğursuzluqla nəticələnə bilər:

1. Diskdə yer yoxdur.
2. Fayl yazmaq üçün müdafiə olunur (**protected).**
3. Faylın adı düzgün yazılmayıb (məsələn, adında iki nöqtə, sual işarəsi və s. simvollar varsa).

Əgər faylı açmaq mümkün deyilsə, onda **fopen** funksiyası **NULL**kimi ifadə olunan xüsusi sıfır qiyməti (boş göstərici) qaytarır. Ona görə də, faylın düzgün açılmasına nəzarət etmək lazımdır. Əgər fayl açılmırsa, onda səhv haqqında məlumat verib, proqramdan çıxmaq lazımdır.

```
if(fp==NULL)
{
printf("Faylmovcuddeyil");
return 1;
}
```

Əgər fayl açılıbsa, onda bu fayldan verilənləri oxumaq olar. Bundan ötrü **fscanf** funskiyasından istifadə olunur. Bu funksiya **scanf** funksiyasına oxşayır. Fərq ondan ibarətdir ki, bu funskiya verilənləri klaviaturadan yox, fayldan oxuyur və onun birinci parametri fayl göstəricisidir.

```
n=fscanf(fp,"&d",&A[i]);
```

Nəticə olaraq, **fscanf** funksiyası oxuduğu ədədlərin sayını qaytarır. Əgər bir ədəd oxunursa, onda **n** dəyişəninin qiyməti 1 ola bilər. Əgər oxuma prosesi səhvlə nəticələnirsə (verilənlər qurtarır və ya səhvdir, yəni fayla ədədlərin əvəzinə sözlər yazılıb), onda n-in qiyməti **0** olur. Fayla verilənləri düzgün yazmaq üçün onlar

boşluqlarla və ya yeni sətrə keçmə simvolu ilə (**Enter** simvolu) bir-birindən ayırmalıdırlar.

Əgər fayl yazmaq üçün açılıbsa, onda **fprintf** funksiyası vasitəsi ilə bu fayla verilənləri yazmaq olar. Bu funskiya **printf** funskiyasına oxşardır.

Faylla işi bitirdikdən sonra onu **fclose** funksiyası vasitəsi ilə bağlamaq lazımdır.

```
fclose(fp);
```

Bundan sonra **fp** göstəricini fayl üçün istifadə etmək olar.

Mətn fayllarla iş zamanı massiv elementləri fayldan ardıcıl oxunmalıdır. Faylın oxunmasında yaranan səhvləri (elementlərin azlığı və ya faylın boş olması) proqramda emal etmək lazımdır.

```
#include <stdio.h>
constint M=5;
constint N=4;
main()
{
	inti,j,A[M][N];
	FILE *fp;

	fp=fopen("input.dat","r");
	for(i=0;i<M;i++)
	for(j=0;j<N;j++)
	if (0==fscanf(fp,"%d",&A[i][j]))
	{
		puts ("Verilenlercatismir!");
		fclose (fp);
		return 1;
	}
	fclose (fp);

}
```

Massivi fayla yazmaq üçün analoji əməliyyatlardan istifadə olunur. Əvvəlcə **fopen**funksiyası vasitəsilə faylı açmaq lazımdır (burada yazma rejimindən istifadə olunmalıdır "**w**"). Sonra isə, ikiqat dövrdə **fprintf** funksiyasından istifadə edərək massivi fayla ötürmək və axırda **fclose**-la bağlamaq lazımdır

2. Müəyyən ölçülü massivlərin emalında fayllardan istifadə

Misal. input.dat faylından 10 tam tipli ədədlərdən ibarət massivi daxil edib, hər elementi 2-yə vurub, alınmış masssivi **output.dat** faylına yazmaq.
Bu məsələni həll etmək üçün **fopen, fscanf, fprintf** və **fclose** funskiyalarından istifadə etmək lazımdır. Proqramda iki vəziyyəti (səhvi) nəzərə almaq lazımdır:

- fayl yoxdur (açılmır);
- faylda verilənlərin sayı azdır və ya verilənlər səhv yazılıb.

```
#include<stdio.h>
constint N=10;
main()
{
inti,A[N];
FILE *fp;
fp=fopen("input.dat","r");  //oxumaqucunfaylinacilmasi
if(fp==NULL)
{
printf("Faylmovcuddeyil!");
return 1;
}
for(i=0;i<N;i++)
if (fscanf(fp,"&d",&A[i])==0)
{ printf("Fayldaverilenlerinsayiazdir.");
break;
}
fclose(fp);
for(i=0;i<N;i++)
A[i]=A[i]*2;
```

```
fp=fopen("output.dat","w");
for(i=0;i<N;i++)
fprintf(fp,"&d\n",A[i]);
fclose(fp);
}
```

Bu proqramın nəticələri ekrana çıxmır, **output.dat** faylına yazılır.

3. Qeyri-müəyyən ölçülü massivlərin emalında fayllardan istifadə

Misal. input.dat faylında iki sütun şəklində (**x, y**) cütlükləri yazılıb. Hər cütlük üçün cəmləri hesablayıb nəticəni **output.dat** faylına yazmaq.

Məsələnin mürəkkəbliyi ondan ibarətdir ki, **input.dat** faylına yazılan cütlüklərin sayı əvvəlcədənməlum deyil. Hər iki **x** və **y** massivinə yaddaşda yer ayırmaq mümkün olmayacaqdır, çünki fayla yazılan ədədlərin sayı məlum deyil.

Lakin, bu məsələni başqa cür həll etmək olar. Hər cütlüyün cəmini hesablamaq üçün yalnız iki ədəd lazımdır, qalan ədədləri yaddaşda saxlamasaq da olar. Cəmi hesabladıqdan sonra, onu yaddaşda saxlamaq lazım deyil, onu birbaşa çıxış faylına yazmaq lazımdır. Məsələnin həlli üçün aşağıdakı alqoritmdən istifadə olunacaqdır:

1. iki faylı açmaq(biri –¿ verilənləri oxumaq, digəri isə verilənləri yazmaq üçün üçün istifadə olunur);
2. iki ədədi **x** və **y** dəyişənlərə oxuyuruq. Əgər oxumaq olmursa (verilənlər yoxdur və ya düz deyil), onda işi bitirmək;
3. **x** və **y** toplayıb, nəticəni çıxış faylına yazmaq;
4. 2-ci addıma keçmək.

Oxuma əməliyyatının müvəffəqiyyətlə başa çatmasını yoxlamaq üçün **fscanf** funskiyasının qaytardığı qiyməti qeyd edəcəyik (bu funskiya oxunan ədədlərin sayını nəticə kimi qaytarır). Hər dəfə iki ədəd, yəni **x** və **y** oxuyacağıq. Əgər hər şey normaldırsa, onda **fscanf** funskiyasının qiyməti 2-yə bərabər olacaqdır. Əgər funskiyanın qiyməti 2-dən azdırsa, onda verilənlər bitib və ya səhvdir.

Qeyd etmək lazımdır ki, eyni zamanda iki açıq faylla işləmək lazımdır, ona görə də iki fayl göstəricisindən istifadə olunmalıdır. Onları **fin** və **fout** kimi işarə edək. Proqramı qısaltmaq məqsədilə faylın açılması zamanı rast gələ biləcəyimiz səhvlər emal olunmur.

```
#include<stdio.h>
```

```
main()
{
int n,x,y,cem;
FILE *fin,*fout;
fin=fopen("input.dat","r");
fout=fopen("output.dat","w");
while(1){
      n=fscanf(fin,"%d%d",&x,&y);
      if(n<2)  break;
      cem=x+y;
      fprintf(fout,"%d\n",cem);
      }
fclose(fout);
fclose(fin);
}
```

Proqramda sonsuz **while** dövründən istifadə olunur. Əgər faylda verilənlər qurtarırsa, onda dövr öz işini bitirir.

4. Massivlərin emalında binar fayllardan istifadə

Mətn fayllardan fərqli olaraq binar fayllarda məlumatlar maşın dilində yazılır. Binar fayla baxarkən heç nə başa düşülmür. Bir əmr vasitəsilə bütöv massivi və ya onun bir hissəsini yazmaq olar.

Binar faylların açılması zamanı **"r", "w"** və **"a"** rejimlərin əvəzinə, uyğun olaraq, **"rb"**, **"wb"** və **"ab"**rejimlərdən istifadə olunur. Əlavə **"b"** hərfi onu göstərir ki, fayl binardır (ing. binary). Baxılan məsələni binar fayl üçün həll etsək:

Misal. input.dat binar faylından 10 tam ədədi oxuyub, hər elementi 2-yə vurub, **output.dat** binar faylına yazmaq.

```
#include<stdio.h>
constint N=10;
main()
{
inti,n,A[N];
FILE *fp;
fp=fopen("input.dat","rb");
n=fread(A,sizeof(int),N,fp);
if(n<N) {
printf("Fayldaelementlerinsayiazdir!");
break;
      }
fclose(fp);
for(i=0;i<N;i++)
   A[i]=A[i]*2;
fp=fopen("output.dat","wb");
fwrite(A,sizeof(in),N,fp);
fclose(fp);
}
```

Binarfayldanverilənlərioxumaqüçün**fread**funskiyasındanistifadəolunur. Bu funskiyanın 4 parametrivar:

1. **yaddaşünvanı,** yənioxunmuşverilənlərin (yuxarıdakımisalda**A**massivininbirincielementininünvanı, **&A[0]** və ya sadəcə, A kimi işarə edilir) harayayazılması;
2. **birelementinölçüsü.**Ölçünükompüterinözününtəyinetməsidahaməqsədə uyğundur. Bu misalda**sizeof(int)** –¿ tam ədədinölçüsüdür. Baxmayaraqki, DevC++ - da tam ədəd 4 baytyertutur, digərproqramlaşdırmapaketlərindəburəqəmfərqliolabilər.

Onagörədəstandart**sizeof**funksiyasındanistifadəetməkdahaməqsədəuyğun dur, çünkiondaproqram**daşınabilən (**ing. **compatible)**olur;

3. **elementlərinsayı (N);**
4. **faylgöstəricisi(fp).**

Nəticəolaraq, **fread**funksiyasıoxunmuşmassivelementlərininümumisayınıqaytarır. Bu qiymətisəhvlərinemalıüçünistifadəetməkolar. Əgərfunksiyanınqaytardığıqiymətelementlərinsayındankiçikdirsə, ondafayldaverilənlərçatışmır.

Massivibinarfaylayazmaqüçün**fwrite** funksiyasından istifadə olunur. Bu funskiyanın parametrləri **fread** funksiyasının parametrləri ilə üst-üstə düşür. Nəticə olaraq, bu funksiya fayla yazılmış elementlərin sayını qaytarır.

Bu üsulun əsas üstünlüyü ondan ibarətdir ki, massiv bir vahid blok şəklində həm oxunur, həm də yazılır. Bu isə proqramın sürətini artırır.

Əgər verilənlər başqa proqram vasitəsilə fayla yazılırsa, onda bu verilənləri binar faylı ilə oxumaq rahat olacaqdır. Binar faylların əsas üstünlüyü ondadır ki, onlardan oxunma və onlara yazma sürətlə baş verir, çünki bütöv massiv birdəfəlik oxunur və ya yazılır. Bu zaman **fread** və **fwrite** funksiyalarında bir elementin uzunluğunu və ümumi sayını, yəni M × N göstərmək lazımdır.

Aşağıdakı proqramda massiv binar fayldan oxunur, onun üzərində müəyyən əməliyyatlar aparılır və sonra o, binar fayla yazılır.

```
#include <stdio.h>
const int M=5;
const int N=4;
main()
{
int total, A[M][N];
FILE *fp;
fp=fopen("input.dat","rb");
total=fread(A,sizeof(int),M*N,fp);
fclose (fp);
```

```
if(total!=M*N)
{
printf("Verilenlerin sayi azdir!");
return 1;
}
fp=fopen("output.dat","wb");
if(M*N!=fwrite(A,sizeof(int),M*N,fp));
printf("Faylin yazilmasinda sehv var!");
fclose (fp);}
```

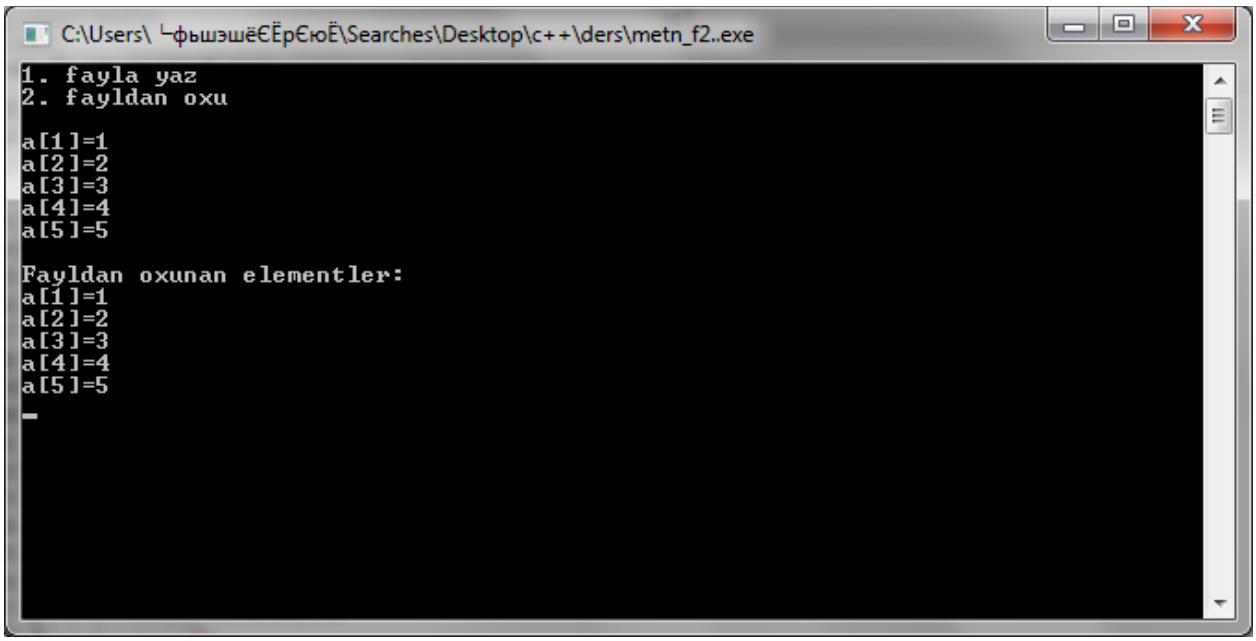
Səhvlərin emalı üçün o faktdan istifadə olunur ki, **fread** və **fwrite** funksiyaları, nəticə olaraq oxunmuş və ya yazılmış elementlərin sayını göndərirlər və əgər o say elementlərin ümumi sayına bərabər deyilsə, onda ekranda səhv haqqında məlumat çıxarılır.

5. Proqram tərtibatı

Tərtib olunmuş proqram seçimdən asılı olaraq yaddaşa yazılmış massivi fayla yazır və fayldan oxuyur:

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
#include<stdlib.h>
#define n 5
int main(void)
{ char z;
      int i,a[n+1];
FILE *f;
printf("1. fayla yaz\n");
printf("2. fayldan oxu\n");
f=fopen("info.txt","w");fclose(f);
```

```
do{
z=getch();
if(kbhit())z=getch();
switch(z)
 {
  case '1':{
              for(i=1;i<=n;i++)
        {
          printf("a[%d]=",i); scanf("%d",&a[i]);
          f=fopen("info.txt","a");
                    fprintf(f,"%s\n","");
                    fprintf(f,"%d",a[i]);
                    fclose(f); }
                 }break;
  case '2':{
           f=fopen("info.txt","r");
           i=1;
           while(!feof(f))
           {
             fscanf(f,"%d",&a[i]);
                          printf("a[%d]=%d\n",i,a[i]);
                          i++;
           }
             fclose(f);
                    }break;
  }
}while(z!='0');
}
```


C:\Users\ └фышэшёЄЁрЄюЁ\Searches\Desktop\c++\ders\metn_f2..exe
1. fayla yaz
2. fayldan oxu
a[1]=1
a[2]=2
a[3]=3
a[4]=4
a[5]=5
Fayldan oxunan elementler:
a[1]=1
a[2]=2
a[3]=3
a[4]=4
a[5]=5

NƏTİCƏ

Bu kurs işində "Massivlərin emalında fayllardan istifadə" mövzusundan bəhs edilir. Kurs işi giriş, nəzəri hissə, proqram tərtibatı, nəticə, istifadə olunmuş ədəbiyyat siyahısından ibarətdir.

Girişdə fayllar haqqında ümumi məlumat verilmişdir.

Nəzəri hissə bir neçə hissədən ibarətdir: Mətn fayllarla işləmə, Müəyyən ölçülü massivlərin emalında fayllardan istifadə, Qeyri-müəyyən ölçülü massivlərin emalında fayllardan istifadə, Massivlərin emalında binar fayllardan istifadə. Nəzəri hissədə hər bir mövzuya uyğun olaraq məlumat və qısa proqramlar verilmişdir.

Proqram tərtibatında tərtib edilmiş massivi fayla yazılması proqramı və nəticəsi göstərilmişdir.

Ədəbiyyat isə istifadə olunmuş kitabın müəlliflərindən ibarətdir. Ədəbiyyat siyahısı 3 adda siyahıdan ibarətdir. Bura Azərbaycan dilində ədəbiyyat daxildir.

ƏDƏBİYYAT

1. R.Ə.Sadıxov, S.B.Həbibullayev - Proqramlaşdırma II hissə. Bakı, 2003
2. Allahverdiyeva N.R., Namazov M.B. - C dilində proqramlaşdırma. Bakı
3. Ə.Sadıxov - C++ proqramlaşdırma dili. Bakı, 2013