

Тошкент давлат техника университети

“Informatika” kafedrası

INFORMATIKA FANIDAN

KURSISHI

Mavzu: Massivlar ustida amallar

Bajardi:98-13 Tu

Ubaydullayev A.

Tekhirdi: Тошматова Ш.С

Toshkent-2013

REJA:

I.KIRISH

1.Masalaning qo'yilishi.

II.ASOSIY QISM.

1.Nazariy qism

*2.Masalaning so'z bilan ifodalangan va blok-sxema
ko'rinishdagi algoritmi.*

3.Masalaning dastur ko'rinishi (qiqacha izohi bilan)

4.Dasturni ishlatishdan olingan natija

III.XULOSA

1.Kurs ishi masalasi bo'yicha xulosa

2.Foydalanilgan adabiyotlar ro'yxati

I.KIRISH

Hozirgi kunda biz hayotimizni axborot texnologiyalarisiz tasavvur qila olmaymiz.Axborot va telekamunikatsi texnologiyalarining jadal suratlar bilan rivojlanishi natijasida insoniyat yangi information jamiyat podevorini qurishga erishdi.XXI-asr axborot texnologiyalari asri hisoblanadi. Axborot va telekamunikatsiya texnologiyalarining tez siratlarda kirib kelishi va jadal rivojlanib borayotganligi respublikamiz rivojiga katta hissa qo`shyapdi desak mubolag'a bo`lmaydi.. Respublikamizda ham axborot va telekomunikatsiya texnologiyalari rivojlantirish va ushbu soha uchun malakali mutaxasislar tayyorlash tizimini shakllantirishga alohida e'tibor berilmoqda . Shu vaqtgacha bu borada juda katta ishlar amalga oshirildi.

Misol uchun maktab,kollej,litsey va oily o`quv yurtlari zamonaviy axborot texnologiyalari ,kompyuterlar bilan jihozlandi. Endilikda shu soha bo`yicha yetuk kadrlar yetishib chiqmoqda. Bunday malakali kadrlar respublikamiz rivoji uchun ,keyingi yosh avlodni bilimli aql zakovatli qilib tarbiyalash uchun suv va havodek zarurdir.Yoshlarni axborot texnologiyalari ,kompyuterlarga qiziqishi quvonarli hol albatta, lekin yoshlarni kompyuterda har xil bekorchi ishlar bilan shug'ullanmasligi uchun biz ularni to`g'ri yo`naltirishimiz kerak.

Bu borada hamma o`quv muassasalarida ayniqsa kasb- hunar kollejlari, litseylar va oily o`quv yurtlarida axborot texnologiyalari dastur mahsulotlarining keng imkoniyatlarini talabalarga chuqur o`rgatish, ularning dasturlash madaniyatini shakllantirish muhim ahamiyat kasb etadi. Respublikamizda ta'lim muassasalarida "Informatika", "Informatika va information texnologiyalar" fanlarining o`quv rejalarida yuqori saviyali algoritmik tillardan birining imkoniyatlarini o`rgatish rejalashtirilgan.C++dasturlash tili mana shunday algoritmik tillardan biridir. Barcha dasturlash tillarida uchraydigan tushuncha va tuzilishlarni o`z ichiga olganligi, katta imkoniyatlarga egaligi, shuningdek, obyektga yo`naltirilgan

dasturlash tillarini o`rganishga asos bolib xizmat qilganligi sababli oily o`quv yurtlarida aynan shu tilni chuqur o`rganishga e`tibor qaratilgan.C++ 1980 yillar boshida Bjarne Stroustrup tomonidan C ga asoslangan tarzda tuzildi.C++ juda ko'p qo'shimchalarni o'z ichiga olgan, lekin eng asosiysi u ob'ektlar bilan dasturlashga imkon beradi. Dasturlarni tez va sifatli yozish hozirgi kunda katta ahamiyat kasb etmoda.

Hozirgi paytda dasturlash tillarining soni juda ko`payib ketmoqda .Lekin shuni aytish karakki har qanday dasturlash tili o`zining darajasi va qo`llash sohasiga ega.Ba'zi bir tillar bir necha xil soha masalalarini yechishda ishlatiladi. Bunday tillar universal tillar deb ataladi.C++ algoritmik tili ham universal til hisoblanadi.U dasturlashning asosiy tushunchalari va konstruksiyalarini o`z ichiga olishi bilan birga boshqa universal dasturlash tillariga qaraganda ancha sodda.

1.Masalaning qo'yilishi.

1-masala.

Tasodifiy yo'l bilan topilgan ikta n ta elementi mavjud bir o'lchovli massiv berilgan. Ularning yig'indisi va ayirmasidan iborat ikta yangi massiv hosil qilinsin. Ular ikta parallel ustun shaklida chop etilsin.

2-masala.

Matritsa bir o'lchovli massivda joylashgan. k -qatorni olib tashlab, hosil bo'lgan massiv ekranga chop etilsin.

3-masala.

c massiv elementlari tasodifiy yo'l bilan hosil qilinsin. Uning o'rta arifmetik qiymati hamda eng kichik elementi topilsin.

II.ASOSIY QISM.

1.Nazariy qism.

Massivlar xotirada ketma-ket joylashgan, bir tipdagi o'zgaruvchilar guruhidir. Alohida bir o'zgaruvchini ko'rsatish uchun massiv nomi va kerakli o'zgaruvchi indeksini yozamiz.C/C++ dagi massivlardagi elementlar indeksi har doim noldan boshlanadi. Bizda char tipidagi m nomli massiv bor bo'lsin va uning 4 dona elementi mavjud bo'lsin.Sxemada bunday ko'rsataylik:

m[0] =4

m[1]= -44

m[2]=109

m[3]=23

Ko'rib turganimizdek, elementga murojaat qilish uchun massiv nomi va [] qavslar ichida element indeksi yoziladi.Bu yerda birinchi element qiymati 4, ikkinchi element - 1 nomerli indeksda -44 qiymatlari bor ekan.Oxirgi element indeksi n-1 bo'ladi (n - massiv elementlari soni).[] qavslar ichidagi indeks butun son yoki butun songa olib keluvchi ifoda bo'lmog'i lozim. Masalan:

...

int k = 4, l = 2;

m[k-l] = 77; // m[2] = 77

m[3] = 2; // m[3] = 46

double d = m[0] = 6; // d = 24

cout << m[1]; // Ekranda: -44

...

Massivlarni ishlatish uchun ularni e'lon qilish va kerak bo'lsa massiv elementlarini initsalizatsiya qilish kerak.Massiv e'lon qilinganda kompilyator elementlar soniga teng hajmda hotira ajratadi.Masalan yuqorida qo'llanilgan char tipidagi m massivini e'lon qilaylik.

char m[4];

Bu yerdagi 4 soni massivdagi elementlar miqdorini bildiradi.Bir necha massivni e'londa bersak ham bo'ladi:

int m1[4], m2[99], k, l = 0;

Massiv elementlari dastur davomida initsalizatsiya qilishimiz mumkin,yoki boshlang'ich qiymatlarni e'lon vaqtida, {} qavslar ichida ham bersak bo'ladi.{ } qavslardagagi qiymatlar massiv initsalizatsiya ro'yhati deyiladi.

int n[5] = {3, 5, -33, 5, 90};

Yuqorida birinchi elementning qiymati 3, ikkinchisi 5 ... ohirgi beshinchi element qiymati esa 90 bo'ldi. Boshqa misol:

```
double array[10] = {0.0, 0.4, 3.55};
```

Bu yerdagi massiv tipi *double* bo'ldi. Ushbu massiv 10 ta elementdan iboratdir. {} qavslar ichida esa faqat boshlangich uchta element qiymatlari berildi. Bunday holda, qolgan elementlar avtomatik tarzda nolga tenglashtiriladi. Bu yerda aytib o'tishimiz kerakki, {} qavslar ichida berilgan boshlangich qiymatlar soni massivdagi elementlar sonidan katta bo'lsa, sintaksis hatosi vujudga keladi. Masalan:

```
char k[3] = {3, 4, 6, -66, 34, 90}; // Hato!
```

Uch elementdan iborat massivga 6 dona boshlangich qiymat berilyapti, bu hatodir. Boshqa misolni ko'rib chiqaylik:

```
int w[] = {3, 7, 90, 78};
```

w nomli massiv e'lon qilindi, lekin [] qavslar ichida massivdagi elementlar soni berilmadi. Bunday holda necha elementga joy ajratishni kompilyator {} qavslar ichidagi boshlangich qiymatlar miqdoriga qarab biladi. Demak, yuqoridagi misolda w massivimiz 4 dona elementdan iborat bo'ladi.

E'lon davridagi massiv initsializatsiya ro'yhati dastur ijrosi vaqtidagi initsializatsiyadan ko'ra tezroq ishlaydigan mashina kodini vujudga keltiradi.

Bir misol keltiraylik.

```
// Massivlar bilan ishlash.
```

```
#include <iostream.h>
#include <iomanip.h>
const int massiv = 8; // massiv kattaligi uchun konstanta
int k[massiv];
char c[massiv] = {5,7,8,9,3,44,-33,0}; // massiv initsializatsiya ro'yhati
int main()
{
    for (int i = 0; i < massiv; i++)
```

```

{   k[i] = i + 1;           // dastur ichida inisalizatsiya
}
for (int j = 0; j < massiv; j++)
{   cout << k[j]
    << setw(4)
    << c[j]
    << endl;
}
return (0);
}

```

Ekranda:

```

1  5
2  7
3  8
4  9
5  3
6 44
7 -33
8  0

```

Yuqorida **<iomanip.h>** faylini dasturimizga kiritdik. Bu e'lon faylida standart kirish/chiqish oqimlari bilan ishlaydigan buyruqlar berilgan. Dasturimizda qo'llanilgan `setw()` manipulyatori chiqish oqimiga berilayotgan ma'lumotlarning eng kichik kengligini belgilaydi, biz `setw()` parametrini 4 deb berdik, demak `c[]` massivi elementlari 4 harf kenglikda ekranga bosiladilar. Agar kenglik kamlik qilsa, u kattalashtiriladi, agar bo'sh joy qolsa, elementlar chapga yondashilib yoziladi.

Misolimizda massiv nomli konstantani qo'lladik. Uning yordamida massiv chegaralarini va **for** strukturasidagi chegaraviy qiymatlarni berdik. Bunday o'zgarmasni qo'llash dasturda xatoga yo'l qo'yishni kamaytiradi. Massiv chegarasi o'zgarganda, dasturning faqat bir joyiga o'zgarish kiritiladi. Massiv hajmi e'lonida faqat **const** sifatli o'zgaruvchilar qo'llanilishi mumkin.

Massivlar bilan ishlaganda eng ko'p yo'l qoyiladigan hato bu massivga 0 dan kichkina va (n-1) dan (n: massivdagi elementlar soni) katta indeks bilan murojaat qilishdir. Bunday xato dastur mantig'i xatosiga olib keladi. Kompilyator bu turdagi hatolarni tekshirmaydi. Keyinroq o'zimizga yozgan massiv klaslarida ushbu xatoni tekshiriladigan qilishimiz mumkin.

10 ta sonning tushish ehtimolini ko'rsaturvchi dastur yozaylik.

// Ehtimollar va massivlar

```
#include <iomanip.h>
#include <iostream.h>
#include <time.h>
#include <stdlib.h>

int main ()
{
    const int massivHajmi = 10;
    int m[massivHajmi] = {0}; // hamma 10 ta element
    srand( time(NULL) ); // 0 ga tenglashtirildi

    for (int i = 0; i < 1000; i++)
    { ++m[ rand() % 10 ];
      }
    for (int j = 0; j < massivHajmi; j++) {
        cout << j << setw(4) << m[j] << endl;
    }
    return (0);
}
```

Ekranda:

```
0 96
1 89
2 111
3 97
4 107
5 91
6 100
7 118
8 99
9 92
```

Ko'rib turganimizdek, sonlarning tushish ehtimoli nisbatan tengdir. Albatta, bu qiymatlar dasturning har yangi ishlashida o'zgaradi.

`++m[ rand() % 10 ]`; Yozuvi bilan biz massivning `rand() % 10` indeksli elementini birga oshirmoqdamiz. Bunda `rand () % 10` ifodasidan chiqadigan qiymatlar `[0;9]` ichida yotadi.

Satrlar, yani harflar ketma-ketligi ("Toshkent", "Yangi yilingiz bilan!"...) C/C++ da char tipidagi massivlar yordamida beriladi. Bunday satrlar bilan islovlar juda tez bajariladi. Chunki ortiqcha tekshirishlar bajarilmaydi.

Harflar, yani literalar, aytib o'tganimizdek, C++ da char tipi orqali beriladi. Literalar apostroflarga ('S', '*' ...) olinadi. Satrlar esa qo'shtirnoqlarga olinadi. Satrlar e'loniga misol beraylik:

```
char *p = "Ikkinchi!?!";
```

Satrlarni yuqoridagi yo'l bilan initsalizatsiya qilsa bo'ladi. Satrlar bu uslubda e'lon qilinganda, yani pointer mexanizmi qo'llanilganda, ba'zi bir kompilyatorlar satrlarni hotiraning konstantalar saqlanadigan qismiga qo'yadi. Ya'ni ushbu satrlarga o'zgartirish kiritilishi ta'qiqlanadi. Shu sababli satrlarni har doim o'zgartirish imkoni bo'lishi uchun ularni char tipidagi massivlar ko'rinishida e'lon qilish afzaldir.

Satrlar massiv yordamida berilganda, ularning uzunligi noma'lumdir. Shu sababli satrning tugaganligini bildirish uchun satr oxiriga maxsus belgi nol literasi qo'yiladi. Uning dasturda belgilanishi '\0' ko'rinishga ega. Satrlarni massiv initsalizatsiya ro'yhati ko'rinishida ham bersak bo'ladi:

```
char c[6] = {'A', 'B', 'C', 'D', 'E', '\0'};
```

```
...
```

```
cout << c;
```

```
...
```

Ekkranda:

ABCDE

Biz **cout** bilan c ning qiymati ekranga bosib chiqardik.

Eng muhimi satr bilan '\0' belgisi uchun yetarli joy bo'lishi kerak. Satrlar massiv yordamida berilganligi uchun, alohida elementlarga indeks orqali yetishish mumkin. masalan:

```
char k[] = "Bahor keldi, gullar ochildi.";
```

```
for (int i = 0; k[i] != '\0'; i++)  
  
    if ( (i mod 2) == 0 )    // juft sonlar uchun haqiqat bo'ladi  
  
        cout << k[i] << " ";
```

Ekkranda:

B h r k l i u l r o h l i

Yuqoridagi misolda, sikl tugashi uchun k[i] element '\0' belgiga teng bo'lishi kerak.

BIR NECHA INDEKSLI MASSIVLAR

Massivlar bir necha indeksga ega bo'lishlari mumkin. C++ kompilyatorlari eng kamida 12 ta indeks bilan ishlashlari mumkin. Masalan, matematikadagi m x n kattalikdagi matritsani ikkita indeksli massiv yordamida bersak bo'ladi.

```
int matritsa [4][10];
```

Yuqorida to'rt satrlik, 10 ustunlik matritsani e'lon qildik. Bir indeksli massivlar kabi ko'p indeksli massivlarni initsalizatsiya ro'yhati bilan birga e'lon qilish mumkin. Masalan:

```
char c[3][4] = {  
  
    { 2, 3,9, 5}, // birinchi satr qiymatlari  
  
    {-10, 77,5, 1}, // ikkinchi "    "  
  
    { 90,233,3,-3} // uchinchi "    "  
  
};
```

```
int m[2][2] = {56,77,8,-3}; // oldin birinchi satrga qiymatlar beriladi,  
  
                        // keyin esa ikkinchi satrga
```

```
double d[4][3][6] = {2.55, -46,0988}; // birinchi satrning dastlabki ikkita  
// elementi qiymat oladi,  
// massivning qolgan elementlari esa  
// nolga tenglashtiriladi
```

Massivning har bir indeksi alohida [] qavslar ichiga olinishi kerak. Yuqoridagi c[][] massivining ikkinchi satr, birinchi ustunidagi elementi qiymatini birga oshirish uchun

```
++c[1][0]; // yoki c[1][0]++;  
  
//    c[1][0] += 1;  
  
//    c[1][0] = c[1][0] + 1;
```

deb yozishimiz mumkin. Massiv indeksleri 0 dan boshlanishini unutmaslik zarur. Agar

```
++c[1,0];
```

deb yozganimizda hato bo'lar edi. C++ bu yozuvni

```
++c[0];
```

deb tushunar edi, chunki kompilyator vergul bilan ajratilgan ro'yhatning eng ohirgi elementini qabul qilardi. Xullas, C++ dagi ko'p indeksli massivlar dasturchiga behisob imkoniyatlar beradi. Undan tashqari, ular xotirada static joylashganligi uchun ularning ishlash tezligi kattadir.

C++ dagi ko'p indeksli massivlar xotirada ketma-ket joylashgandir. Shu sababli agar massiv funksiyaga kirish parametri sifatida berilsa, faqat birinchi indeks tushurilib qoldiriladi, qolgan indekslar esa yozilishi shartdir. Aks taqdirda funksiya massiv kattaligini to'g'ri keltirib chiqarolmaydi. Massiv parametrli bir funksiya e'lonini beraylik.

```
//Ko'p indeksli massivlar  
#include <iostream.h>  
int indeks = 3;
```

```
int intArray[indeks][4] = {}; // hamma elementlar 0 ga tenglashtirildi
void printArray(int mass[][4], int idx){ // funksiya e'loni
```

```
    for (int i = 0; i < idx; i++) { // massivning birinchi indeksini
        // o'zgartirsa bo'ladi
```

```
        for (int k = 0; k < 4; k++){ // massivning ikkinchi indeksi o'zgarmaydi
            cout << mass[i][k];
        }
        cout << endl;
    }
    return;
}
...
```

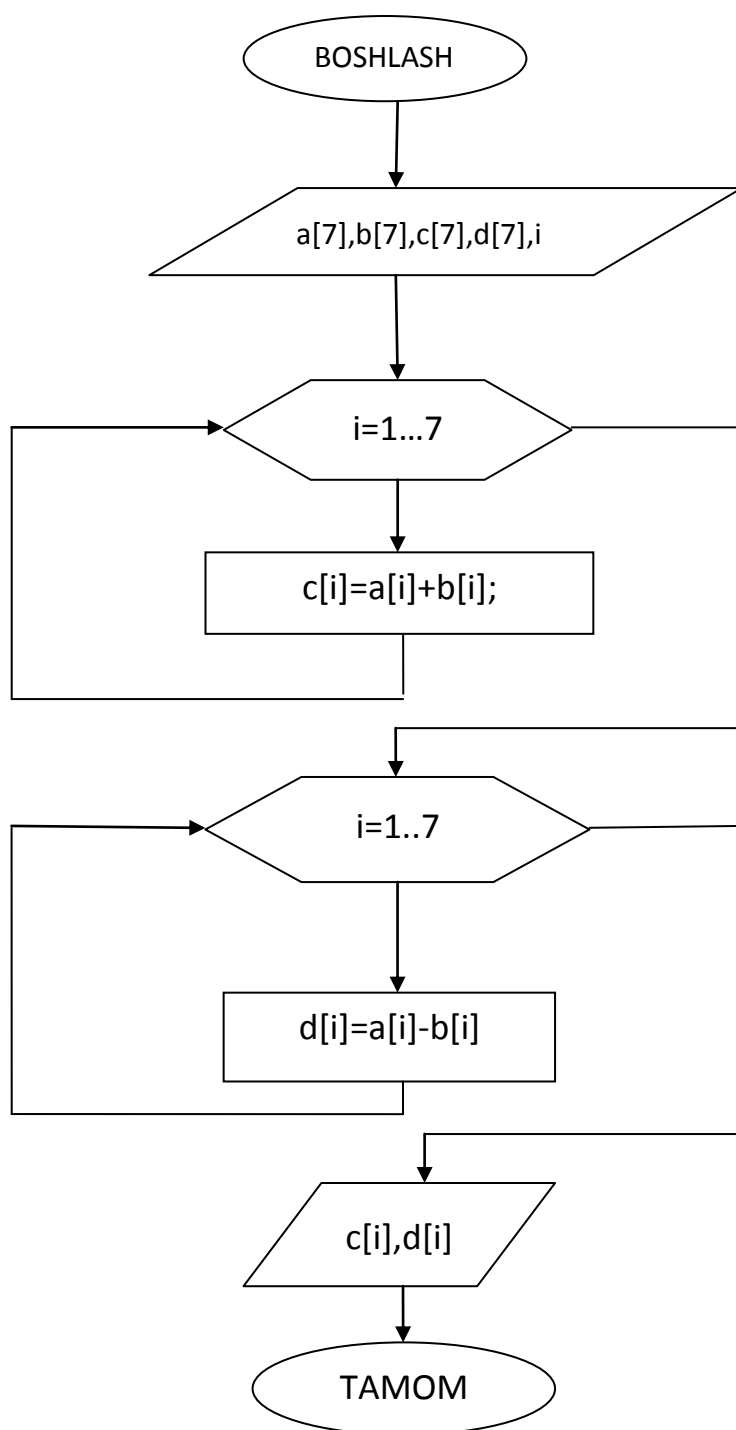
```
int main()
{
    ...
    printArray(intArray); // funksiya chaqirig'i
    ...
    return (0);
}
```

Massivning indekslarini funksiyaga bildirish yana muammoligicha qoladi. Albatta, birinchi indeksdan tashqari qolgan boshqa indekslar kattaligini funksiya ichida berish ma'noga egadir. Lekin birinchi indeks kattaligini tashqaridan, qo'shimcha parametr sifatida bersak, funksiyamiz chiroyliroq chiqadi, turli kattalikdagi massivlarni o'lish imkoniga ega bo'ladi.

2. Masalaning soʻz bilan ifodalangan va blok-sxema koʻrinishdagi algoritmi

1-masala.

Tasodifiy yoʻl bilan topilgan ikta n ta elementi mavjud bir oʻlchovli massiv berilgan. Ularning yigʻindisi va ayirmasidan iborat ikta yangi massiv hosil qilinsin. Ular ikta parallel ustun shaklida chop etilsin.

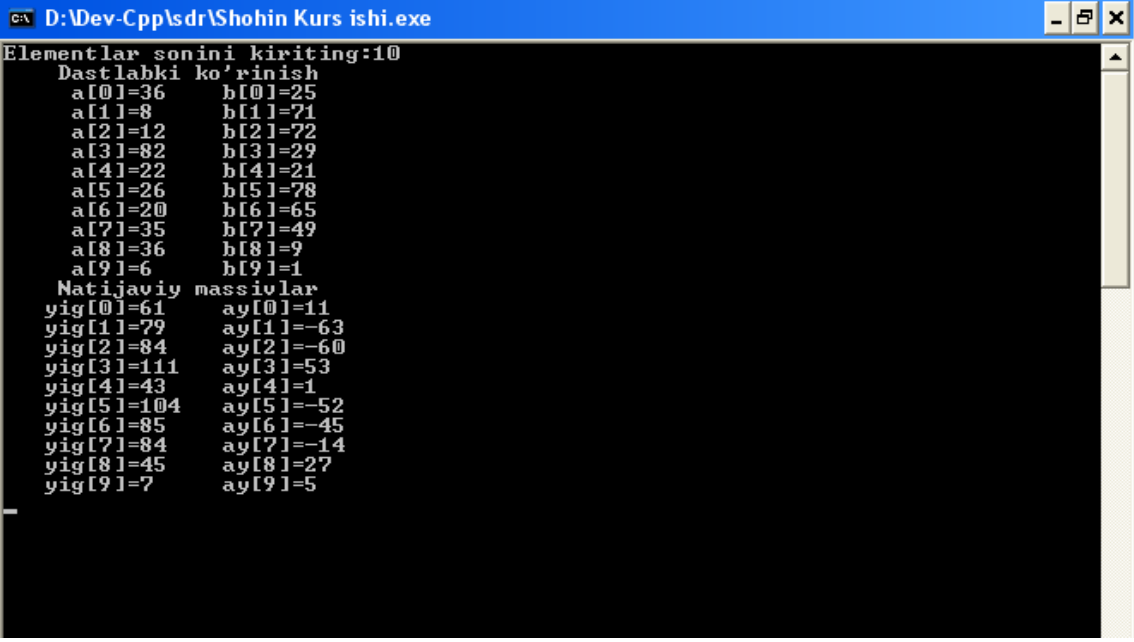


3. Masalaning dastur ko'rinishi (qiqacha izohi bilan)

1-masala.

```
#include<iostream.h>
#include<conio.h>
#include<iomanip.h>
#include<time.h>
#include<stdlib.h>
void main()
{
    int a[100],b[100],n,i,yig[100],ay[100];
    cout<<"Elementlar sonini kiriting:";
    cin>>n;
    srand(time(0));
    for(i=0;i<n;i++)
    {
        a[i]=rand()%100;//son qiymatlarni soatdan oloadi
        b[i]=rand()%100;
    }
    cout<<"  Dastlabki ko'rinish\n";
    for(i=0;i<n;i++)
    {
        cout<<setw(7)<<"a["<<i<<"]="<<a[i]<<"\t"<<"b["<<i<<"]="<<b[i]<<endl;
    }
    for(i=0;i<n;i++)
    {
        yig[i]=a[i]+b[i];//yig'indini hisoblaydi
        ay[i]=a[i]-b[i];//ayirmani hisoblaydi
    }
    cout<<"  Natijaviy massivlar\n";
    for(i=0;i<n;i++)
    cout<<setw(7)<<"yig["<<i<<"]="<<yig[i]<<"\t"<<"ay["<<i<<"]="<<ay[i]<<endl;
    getch();
}
```

4. Dasturni ishlatishdan olingan natija



```
Elementlar sonini kiriting:10
Dastlabki ko'rinish
a[0]=36    b[0]=25
a[1]=8     b[1]=71
a[2]=12    b[2]=72
a[3]=82    b[3]=29
a[4]=22    b[4]=21
a[5]=26    b[5]=78
a[6]=20    b[6]=65
a[7]=35    b[7]=49
a[8]=36    b[8]=9
a[9]=6     b[9]=1
Natijaviy massivlar
yig[0]=61   ay[0]=11
yig[1]=79   ay[1]=-63
yig[2]=84   ay[2]=-60
yig[3]=111  ay[3]=53
yig[4]=43   ay[4]=1
yig[5]=104  ay[5]=-52
yig[6]=85   ay[6]=-45
yig[7]=84   ay[7]=-14
yig[8]=45   ay[8]=27
yig[9]=7    ay[9]=5
```

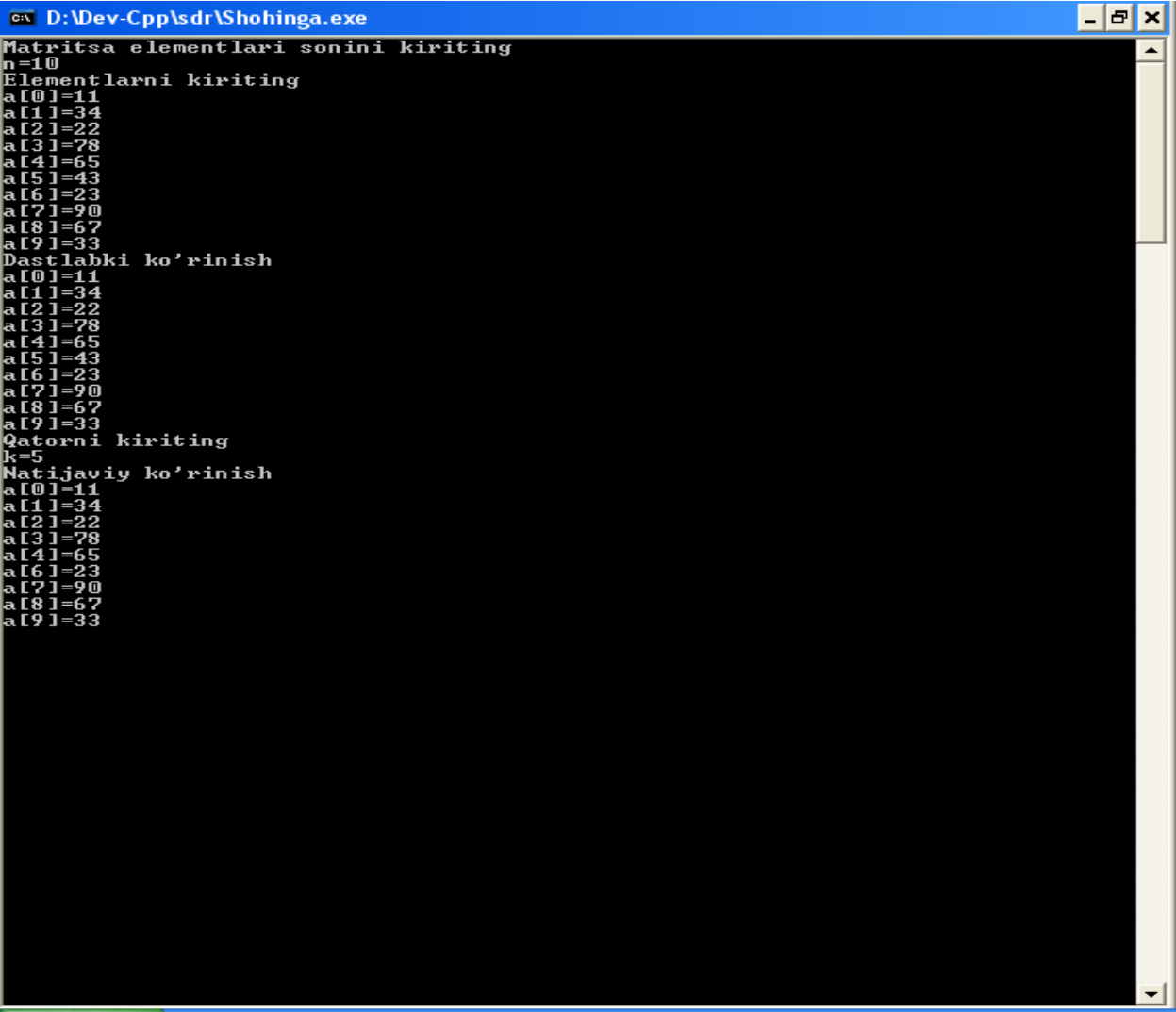
2-masala.

Matritsa bir o'lchovli massivda joylashgan. k-qatorni olib tashlab, hosil bo'lgan massiv ekranga chop etilsin.

```
#include<iostream.h>
#include<conio.h>
void main()
{
    int a[100],i,n,k;
    cout<<"Matritsa elementlari sonini kiriting\nn=";
    cin>>n;
    cout<<"Elementlarni kiriting\n";
    for(i=0;i<n;i++)
    {
        cout<<"a["<<i<<"]="";
        cin>>a[i];
    }
    cout<<"Dastlabki ko'rinish\n";
    for(i=0;i<n;i++)
    {
        cout<<"a["<<i<<"]="<<a[i]<<endl;
    }
    cout<<"Qatorni kiriting\nk=";
    cin>>k;
    cout<<"Natijaviy ko'rinish"<<endl;
    for(i=0;i<n;i++)
    if(i!=k)          //k-qatorni olib tashlaydi
    {
        cout<<"a["<<i<<"]="<<a[i]<<endl;
    }

    getch();
}
```

Dasturni ishlatishdan olingan natija



```

D:\Dev-Cpp\src\Shohinga.exe
Matritsa elementlari sonini kiriting
n=10
Elementlarni kiriting
a[0]=11
a[1]=34
a[2]=22
a[3]=78
a[4]=65
a[5]=43
a[6]=23
a[7]=90
a[8]=67
a[9]=33
Dastlabki ko'rinish
a[0]=11
a[1]=34
a[2]=22
a[3]=78
a[4]=65
a[5]=43
a[6]=23
a[7]=90
a[8]=67
a[9]=33
Qatorni kiriting
k=5
Natijaviy ko'rinish
a[0]=11
a[1]=34
a[2]=22
a[3]=78
a[4]=65
a[5]=43
a[6]=23
a[7]=90
a[8]=67
a[9]=33

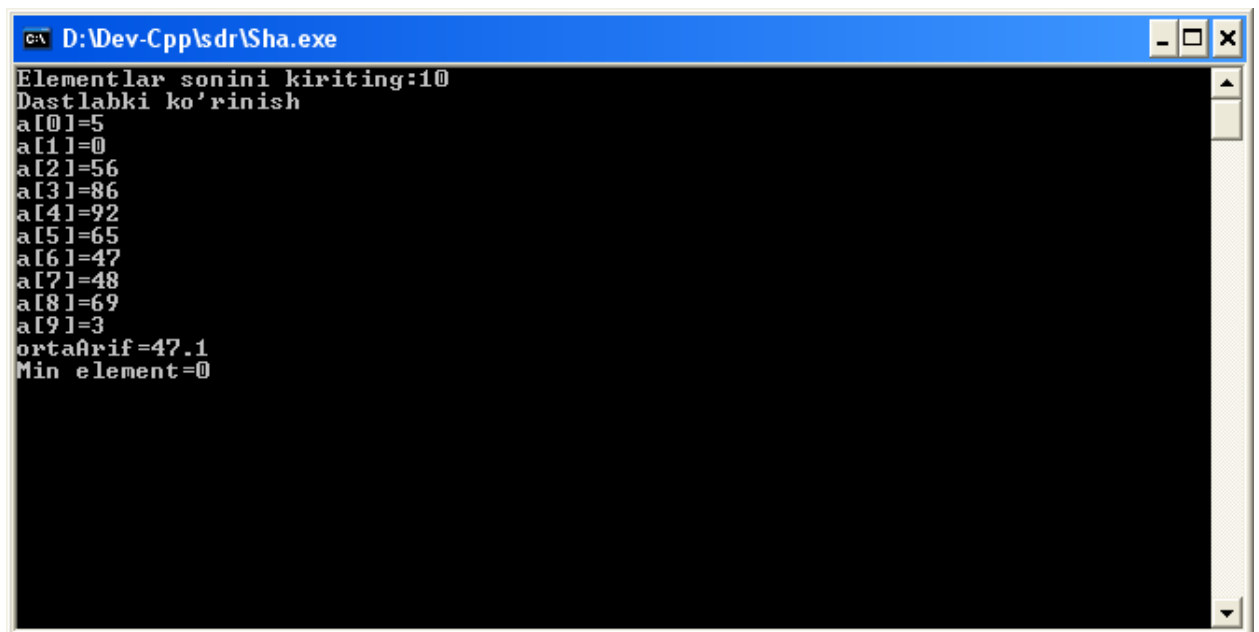
```

3-masala.

c massiv elementlari tasodifiy yo'l bilan hosil qilinsin. Uning o'rta arifmetik qiymati hamda eng kichik elementi topilsin.

```
#include<iostream.h>
#include<conio.h>
#include<iomanip.h>
#include<time.h>
#include<stdlib.h>
int main()
{
    int a[100],min=100,n,i,j=0,k;float ortaArif=0;
    cout<<"Elementlar sonini kiriting:";
    cin>>n;
    srand(time(0));
    for(i=0;i<n;i++)
    {
        a[i]=rand()%100; //son qiymatlarni soatdan oladi
    }
    cout<<"Dastlabki ko'rinish\n";
    for(i=0;i<n;i++)
    {
        cout<<"a["<<i<<"]="<<a[i]<<endl;
    }
    for(i=0;i<n;i++)
    if(min>a[i])
    {
        min=a[i]; //eng kichik elementni hisoblaydi
    }
    for(i=0;i<n;i++)
    {
        j++;
        ortaArif=(ortaArif+a[i]);
    }
    ortaArif=ortaArif/j; //o'rta arifmetik qiymatini hisoblaydi
    cout<<"ortaArif="<<ortaArif<<endl;
    cout<<"Min element="<<min<<endl;
    getch();
}
```

Dasturni ishlatishdan olingan natija



The screenshot shows a Windows command prompt window with the title bar "C:\ Dev-Cpp\sdr\Sha.exe". The window contains the following text:

```
Elementlar sonini kiriting:10
Dastlabki ko'rinish
a[0]=5
a[1]=0
a[2]=56
a[3]=86
a[4]=92
a[5]=65
a[6]=47
a[7]=48
a[8]=69
a[9]=3
ortaArif=47.1
Min element=0
```

III.XULOSA

1.Kurs ishi masalasi bo'yicha xulosa

“Kadrlar tayyorlash milliy dasturi”da shaxsni rivojlantirishga berilgan juda katta e'tiborning boisi shundaki, faqat shaxsning kamoloti uning intellektual zakovatigina jamiyat taraqqiyotining omili bo'la oladi. Shaxsdagi intellektual zakovat esa yaxshi tashkil etilgan ta'lim-tarbiya jarayonidagina uyg'onishi, taraqqiy etishi va shaxs kamolotini ta'minlashi mumkin. O'qituvchi o'quvchilarning intellektual imkoniyatlarini maksimal ishga solish yo'lini izlab topishi lozim.

Ko'p yillar davomida an'anaviy dars o'tish ta'limning asosiy shakllaridan biri bo'lib keldi.

An'anaviy dars- muayyan muddatga mo'ljallangan, ta'lim jarayoni ko'proq o'qituvchi shaxsiga qaratilgan, mavzuga kirish, yoritish, mustahkamlash va yakunlash bosqichlaridan iborat ta'lim metodidir.

An'anaviy dars o'tish modelida ko'proq ma'ruza, savol-javob, amaliy mashq kabi metodlardan foydalaniladi.

Buning natijasi o'laroq yoshlarimiz Informatika va Axborot texnologiyalari fanidan kuchli bilim va tajribaga ega bo'lmoqdalar.

Ayniqsa dasturlash sohasida ko'plab yoshlarimiz ulkan yutuqlarga erishmoqdalar. Chunki prezidentimiz I.A.Karimov Ta'lim sohasini rivojlantirish maqsadida Davlat byudjetini 50% ni yoshlarni ta'lim olishi uchun barcha sharoitlarni yaratib berishga sarfladilar.

Shunday ekan, an'anaviy dars shaklini saqlab qolgan holda, unga turli-tuman o'quvchilar faolligini oshiradigan metodlar bilan boyitilib borilsa, o'quvchilar o'zlashtirish darajasini ko'tarilishiga olib keladi.

Buning uchun dars jarayoni oqilona tashkil qilinishi, o'qituvchi tomonidan o'quvchilarning qiziqishini ortirib, ularning ta'lim jarayonida faolligi muttasil rag'batlantirilib turilishi, o'quv materialini kichik-kichik bo'laklarga bo'lib, ularning mazmunini ochishda baxs, munozara, aqliy hujum, kichik guruhlarda ishlash, tadqiqot rolli o'yinlar metodlarini qo'llash, rang-barang baholash usullaridan foydalanish ta'lim vositalaridan o'rinli foydalanish talab etiladi.

Zamonaviy o'qitish texnologiyalarida talaba shaxs pedagogik jarayon markaziga qo'yiladi uning rivojlanishiga va tabiiy imkoniyatlarini ro'yobga chiqarishga qulay shart- sharoit yaratiladi. O'zbekiston Respublikasining ta'lim tizimi milliy modelining tarkibiy qismi sifatida “shaxs” kiritilganligi va unga birinchi o'rin ajratilganligi shaxsga yo'naltirilgan ta'lim texnologiyalariga alohida e'tibor berilayotganligi fikrimizning yaqqol dalilidir. Shunday ekan biz yoshlar vatanga munosib farzand bo'lishga va yurtimiz bayrog'ini yuksaklarga ko'tarishga bor kuch va g'ayratimizni ishga solgan holda harakat qilishimiz shart.

2. Foydalanilgan adabiyotlar ro'yxati

1)"Informatika" Tayloqov.

2)"Dasturlash" P. Karimov, S.Irisqulov.

3)"Informatika va axborot texnologiyalari" fanidan ma'ruzalar matni

T.U .O'tapov, G'.R.Yodgorov.

4)"Dasturlash" Sh.I.Razzoqov. M.J.Yunusova.

5)"Zamonaviy Dasturlash tillari" Xudoyorov Sh .J.

6)musad.uz sayti.

7) "Kurs ishini bajarish uchun qo'llanma" Idiatulina E.M, Rasulova M.R,
Gapirova U.I