

8 - laboratoriya ishi

Mavzu: Matritsalar bilan ishlash algoritmlari. Bir o'lchovli massivlar uchun massiv operatorlaridan foydalanib masalalarga dastur tuzish.

Massiv - bir xil turdagi ma'lumotlar to'plamidir. Massivg quyidagi ko'rinishda yoziladi: *o'zgaruvchi_turi massiv_nomi [massiv_uzunligi]*. O'zgaruvchining turidan keyin massiv nomi, so'ngra kvadrat qavslarda uning o'lchamlari ko'rsatiladi. Masalan, 4 ta sondan iborat massivni aniqlaymiz: *int mas [4]*. Ushbu massiv to'rtta elementga ega, ammo ularning qiymatlari aniqlanmagan. Bu elementlarga o'rta qavslar orqali boshlang'ich qiymatlarni berishimiz mumkin: *int mas [4] = {1,2,3,4};*

Agar massiv o'lchamiga nisbatan qiymatlar kamroq bo'lsa, unda qiymatlar massivning dastlabki elementlariga beriladi. Agar massiv o'lchamiga nisbatan qiymatlar ko'proq bo'lsa, unda kompilyatsiya paytida xato yuzaga keladi: *int mas [4] = {1, 2, 3, 4, 5, 6};*. Bu yerda massiv 4 o'lchamga ega, ammo unga 6 qiymat berilgan. Agar massivning o'lchami aniq ko'rsatilmagan bo'lsa, unda o'lcham berilgan qiymatlar soniga qarab aniqlanadi: *int mas [] = {1, 2, 3, 4, 5, 6};*. Bu holda, massivda 6 element mavjud bo'lib, massiv o'lchami ham 6 ga teng.

Belgilar massivlarini initsializatsiyalash (qiymat berish) o'ziga xos xususiyatlarga ega. Ushbu belgilar massivda biz boshlang'ich qiymatni belgilar orqali ham, satr ko'rinishida ham berishimiz mumkin:

```
char s1 [] = {'h', 'e', 'l', 'l', 'o'};
```

```
char s2 [] = "dunyo";
```

Bundan tashqari, ikkinchi holatda, s2 massivida 5 ta emas, balki 6 ta elementga ega bo'ladi. Chunki belgilar massivga satr orqali qiymat berganda massiv oxiriga avtomatik ravishda '\0' nol belgisi qo'shiladi. Shu bilan birga, bitta massiv elementlarini boshqa massivga o'lashtirishga ruxsat berilmaydi:

```
int nums1 [] = {1,2,3,4,5};
```

```
int nums2 [] = nums1; // xato
```

```
nums2 = nums1; // xato
```

Massivni aniqlagandan so'ng, indeks orqali uning elementlariga murojaat qilishimiz mumkin. Indekslar noldan boshlanadi, shuning uchun birinchi elementga kirish uchun 0 indeksidan foydalanish kerak. Elementga indeks orqali murojaat qilib, uning qiymatini olishimiz yoki o'zgartirishimiz mumkin:

1-misol.

```
#include <iostream>
```

```
using namespace std;
```

```
int main()
```

```

{
    int numbers[4] = {1,2,3,4};
    int first_number = numbers[0];
    cout << first_number<<endl; // 1
    numbers[0] = 34;
    cout << numbers[0] << endl; // 34
    return 0;
}

```

Sikldan foydalanib, massivning elementlariga indekslar orqali murojaat qilish mumkin:

2-misol.

```

#include <iostream>
using namespace std;
int main()
{
    int numbers[4] = {1,2,3,4};
    int size = sizeof(numbers)/sizeof(numbers[0]);
    for(int i=0; i < size; i++)
        cout << numbers[i] << endl;
    return 0;}

```

Sikl yordamida massivni bo'ylab yurish uchun avval massivning uzunligini topish kerak. Uzunlikni topish uchun sizeof operatoridan foydalaniladi. Aslida, massiv uzunligi uning elementlarining umumiy uzunligiga teng. Barcha elementlar bir xil turda va xotirada bir xil hajmni egallaydi. Shunday qilib, sizeof (numbers) ifodasidan foydalanib, butun massiv uzunligini baytlarda topamiz va sizeof (numbers [0]) yordamida bitta element uzunligini baytlarda topamiz. Birinchi qiymatni ikkinchisiga bo'lib, massivdagi elementlar sonini aniqlash mumkin. So'ng for operatoridan foydalanib, i hisoblagich qiymati massiv uzunligiga teng bo'lgunga qadar barcha elementlarga murojaat qilamiz.

3-misol. Haqiqiy $a(n)$, $n \leq 15$ massivning barcha musbat elementlarining o'rta arifmetigini hisoblang.

Belgilashlarni kiritamiz. i - massiv elementlari indekslarining qiymatlarini aniqlovchi o'zgaruvchi, $i=1,2,...,k$ (k - massiv elementlari soni, $k \leq 15$), S va kn mos ravishda musbat elementlarning o'rta arifmetik qiymati va ularning soni. $a[i] > 0$ shart bajarilsa $S=S+a[i]$, $kn=kn+1$ yig'indi hisoblanishi kerak. Ushbu yig'indini hisoblash tugagandan keyin $S=S/kn$ hisoblanadi, ya'ni musbat elementlar yig'indisi musbat elementlar soniga bo'linadi. Agar $kn=0$ bo'lsa, massivning musbat elementlari mavjud emas.

4-misol. Bir o'lchovli haqiqiy $B(N)$, $n=20$ massivning eng katta elementini va uning tartib raqamini toping.

Belgilashlar: $bmax$ va $imax$ – $B(N)$ massivning eng katta elementi va uning tartib raqami; k – massiv elementlari soni bo'lsin.

Dastlab b massivning barcha elementlari qiymatini kiritamiz. Massiv birinchi elementini eng katta element deb faraz qilamiz: $bmax=b_1$, $imax=1$. i sikl parametri 2 dan k gacha o'zgaradi. Har gal $b_i > bmax$ shart tekshiriladi, agar bu shart bajarilsa $bmax=b_i$, va $imax=i$ almashtirishlar bajariladi.

```
#include <iostream>
using namespace std;
int main()
{
//      const int n=5;
int i,k,imax;
float bmax;
//float b[n];
cout<<"k ni kiriting"<<endl;
cin>>k;
float b[k];
cout<<"b massivni kiriting \n";
for (i=1;i<=k;i++)cin>>b[i];
{bmax=b[1]; imax=1;}
for (i=1;i<=k;i++)
{
if (b[i]>bmax)
{bmax=b[i]; imax=i;}
}
cout<<"bmax="<<bmax<<"\t imax="<<imax;
return 0;
}
```