

## Основни алгоритми за обработка на масив

**1. Въвеждане стойности за елементите на масива и извеждане стойностите на елементите на масива** – въвеждането и извеждането става поелементно с помощта на цикъл (най-често for)

Пример:

```
/*въвеждане стойности за елементи на масив*/
for (i=0;i<=10;i++)
{
    cout<<"a["<<i<<"]=";
    cin>>a[i];
}
/*извеждане стойности на елементи на масив*/
for (i=0;i<=10;i++)
    cout<<"a["<<i<<"]= "<<a[i]<<"\n";
```

### 2. Намиране max, min елемент на масива

Задачата за намиране най – големия (най- малкия) по стойност елемент от елементите на масива се свежда до следната идея: Последователно, от първия към последния елемент на масива, се обхождат всички елементи на масива; като най-голямата стойност се помни в междинна променлива (например в MAX); всеки прегледан елемент се сравнява с MAX и ако е по – голям, стойността на MAX се променя и става равна на стойността на този елемент. По този начин на всяка стъпка MAX ще съдържа най – голямата стойност, открита до тази стъпка. След като се обхождат всички елементи на масива, MAX ще има стойността на най – големия елемент.

За начална стойност на MAX се избира първия елемент на масива. Този алгоритъм може да се приложи и за намиране на най – малкия елемент на масива.

Пример:

```
/*намиране max/min елемент*/
int max=a[0],min=a[0];
for (i=1;i<=10;i++)
{
    if (max<a[i])
        max=a[i]; /* намиране на max елемент */
    if (min>a[i])
```

```

min=a[i]; /* намиране на min елемент */
}
cout<< "макс.елемент е " <<max<<"\n";
cout<< "миним.елемент е " <<min<<"\n";

```

### 3. Сумиране елементите на масив без условие/при условие

Въвежда се променлива (с име SUMA, SUMA1) в която ще се натрупва сумата от стойностите на елементите на масива:

```

/*сума на всички елементи на масива */
int suma=0,suma1=0;
for (i=0;i<=10;i++) cin>>a[i];
suma=suma+a[i]; /*сума без условие*/
cout<<"сума на всички елементи на масив =" <<suma<<"\n";
for (i=0;i<=10;i++)
if (a[i]<0)
suma1=suma1+a[i]; /*сума с условие – сума на всички отрицателни елементи */
cout<<"сума на отрицателните елементи на масив=" <<suma1<<"\n";

```

### 4. Броене елементи на масив с условие

Алгоритъмът, по който се решават задачите за броене има следните стъпки:

- Предварително на променлива от цял тип (например br-брояч) се присвоява стойност 0 (в края br ще съдържа броя на елементите, отговарящи на поставеното условие)
- Последователно се обхождат всички елементи, като за всеки от тях се проверява дали отговаря на условието; ако елементът отговаря на условието, стойността на променливата се увеличава с 1.

```

/*броене на елементите на масив */
int br=0;
for (i=0;i<=10;i++) cin>>a[i];
if (a[i]>=0)
br=br+1; /*брояча се увеличава с 1*/
cout<<"брой полож.елементи в масива =" <<br<<"\n";

```

### 5. Търсене на елемент по определено условие

Последователно, от първия към последния елемент на масива, се обхождат всички елементи на масива; всеки елемент се проверява дали отговаря на поставеното условие за търсене; като резултат елементът може да бъде намерен или не.

Тази задача прилича на задачата за броене елементи на масив при определено условие.

```
/*търсене на елементите на масив по условие*/  
int br=0;  
for(i=0;i<=10;i++) cin >>a[i];  
if (a[i]==5 ) /* условие за търсене */  
br=br+1; /*брояча се увеличава с 1*/  
cout<<"броя на елементите = "<<br<<"\n";
```

## 6. Размяна на стойностите на елементите на масив

Тази операция се извършва с помощта на допълнителна променлива от типа на масива.

Като пример ще използвам размяната на стойностите на първия и петия елемент на масива А (Р е променлива от от типа на масива):

```
p=a[1];  
a[1]=a[5];  
a[5]=p;
```