

# Масиви

Последователност от еднотипни  
променливи

Пример

```
int a[10];
```

е обява за последователност от  
10 целочислени променливи

Разполагаме с променливите:

`a[0], a[1], a[2], ....., a[9]`

Употреба:

Присвояване на стойности:

```
int i=0; int v; cin >> v; a[i]=v;
```

```
int w=a[i];
```

# Общ вид на дефиниция на масив

<тип на променливите><име>[<размер>];

Примери:

```
int a[10];
```

```
char c[20];
```

```
const int N=100; double d[N];
```

# Предупреждение

Когато извикваме елемент от масив  $a[i]$ , винаги трябва да сме сигурни, че стойността на индекса  $i$  има стойност **в границите от дефиницията на масива**.

Пример: ако дефиницията е `int a[10]`, може да използваме  $i$  със стойност 0, 1, 2, ..., 9

Грешка е и когато извикваме  $a[i]$ , със стойност на  $i$ , която не е дефинирана:

Пример за грешка: `int a[10]; int i; a[i]=0;`

# Цикъл и масив

```
for(int i=0; i<10; i++) cin >> a[i];
```

```
for(int i=0; i<10; i++) a[i] = i*i;
```

```
for(int i=0; i<10; i++) cout << a[i] << endl;
```

# Максимален и актуален размер

```
const int N=1000;
```

```
int n;
```

```
int a[N];
```

```
cin >> n;
```

```
for(int i=0; i<n; i++) cin >> a[i];
```

# Зареждане със случайни стойности

```
#include<iostream>
#include<cstdlib>
#include<ctime>
using namespace std;
const int N=100;
int a[N];
int n;

int main()
{
    cin >> n;
    srand(time(0));
    for(int i=0;i<n;i++) a[i]=1+rand()%6;
    for(int i=0;i<n;i++) cout << a[i] << endl;
}
```

# Основни елементарни задачи с масиви

Сума от елементите на масив:

```
int s=0; for(int i=0;i<n;i++) s += a[i];  
cout << s << endl;
```

Произведение от елементите на масив:

```
int p=1; for(int i=0;i<n;i++) p *= a[i];  
cout << p << endl;
```



# Максимум и минимум

```
const int MaxVal=(int)1e9;
int max_v=-MaxVal; int max_i;
int min_v=MaxVal; int min_i;
for(int i=0;i<n;i++)
{
    if(max_v<a[i]) {max_v=a[i]; max_i=i;}
    if(min_v>a[i]) {min_v=a[i]; min_i=i;}
}
cout << maxv << endl; cout << maxi << endl;
cout << minv << endl; cout << mini << endl;
```

# Търсене в масив

```
int x; cin >> x;
```

```
int j=-1;
```

```
for(int i=0;i<n;i++) {if(a[i]==x){j=i; break;}}
```

```
if(j== -1) cout << "не е намерен" << endl;
```

```
else cout << j << endl;
```

# Сортиране

Разместване на елементите, така че да се подредят в растящ ред

пример 1: 7 5 1 8 3 -> 1 3 5 7 8

пример 2: 7 5 7 1 2 9 -> 1 2 5 7 7 9

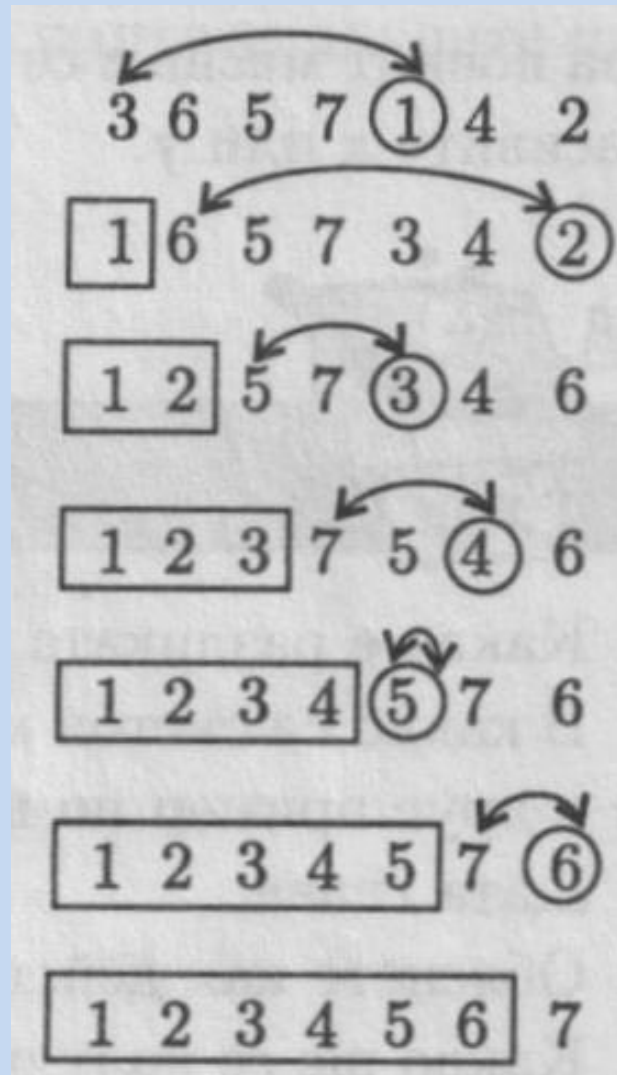
# Използване на стандартна функция

```
include<iostream>
#include<algorithm> // библиотека STL
using namespace std;

const int N=100;
int a[N];
int n;

int main()
{
    n=5;
    for(int i=0;i<n;i++) cin >> a[i];
    sort(a,a+n);
    for(int i=0;i<n;i++) cout << a[i] << endl;
}
```

# „Изобретяване“ на (макар и по-неефективен) алгоритъм за сортиране



# Реализация на идеята – сортиране с избор

```
for(int j=0;j<n;j++)  
{  
    // намираме най-малката стойност  
    // измежду a[j], a[j+1], ..., a[n-1]  
    // нека p е индексът на тази стойност  
    //  
    // взаимно разменяме стойностите a[j] и a[p]  
}
```

// сортиране на масива  $a[i]$ ,  $i=0, \dots, n-1$

```
for(int j=0;j<n;j++)  
{  
    int m=a[j]; int p=j;  
    for(int i=j+1;i<n;i++) if(m>a[i]){m=a[i];p=i;}  
    swap(a[j],a[p]);  
}
```

# Сливане на два сортирани масива

```
// a[m], b[n], c[m+n]
```

```
int i=0, j=0, k=0;
```

```
while((i<m) || (j<n))
```

```
    if(j>=n) {c[k]=a[i]; i++; k++;}
```

```
    else if(i>=m){c[k]=b[j]; j++; k++;}
```

```
    else if(a[i]<b[j]) {c[k]=a[i]; i++; k++;}
```

```
    else             {c[k]=b[j]; j++; k++;}
```