

# МАСИВИ

## ЕДНОМЕРНИ МАСИВИ / ARRAYS/

### 1. Деклариране на едномерен масив

**Едномерният масив е списък от променливи, които са от един и същи тип и се достигат чрез общо име.**

Всяка отделна променлива от масива се нарича **елемент на масива**.  
Масивите предоставят удобен начин за работа със свързани данни

**тип име на променлива [размер]**

**тип** – тип на данните

**име на променлива** – името на масива

**размер** – броя на елементите на масива

**Пример 1:** Програмата зарежда масива **sqrs** с квадратите на числата от 1 до 10 и след това ги изписва на екрана

```
#include <iostream>
using namespace std;
int main(void)
{
    int sqrs[10];
    int i;
    for(i=1; i<11; i++) sqrs[i-1] = i*i;
    for(i=0; i<10; i++)
        cout<< sqrs[i]<<" ";
    return 0;
}
```

**Ако искате да копирате стойностите на всички елементи на един масив в друг, трябва да сторите това поотделно за всеки елемент**

**Пример 2:** Тази програма зарежда **a1** с числата от 1 до 10 и след това ги копира в **a2**

```
#include <iostream>
using namespace std;
int main(void)
{
    int a1[10], a2[10];
```

```

    int i;
    for(i=1; i<11; i++) a1[i-1] = i;
    for(i=0; i<10; i++) a2[i] = a1[i];
    for(i=0; i<10; i++)
        cout <<a2[i];
    return 0;
}

```

**Зад 3.** Да се състави програма, която въвежда в масив средния годишен успех на всеки ученик от един клас. Да се изведе най-високият успех

```

#include<iostream>
using namespace std;
int main()
{
    float arr[30]; //дава възможност за въвеждане успеха на 30 ученика
    int i,n; //i-брояч на номер на ученик, n- бр. ученици
    cout<<"Vavedete broi uchenici  " ;
    cin>>n;
    cout<<"Vavedete uspehat na vseki edin ot tiah ";
    for(i=0;i<n;i++) //брояча брои учениците
        cin>>arr[i]; /* дава възможност за въвеждане успеха на всеки
един ученик*/
    float max=arr[0]; //приема първият успех за максимален
    for(i=1;i<n;i++) //брояча увеличава индекса
        if(arr[i]>max) max=arr[i]; /* и сравнява всеки следващ успех/ел-т
от масива/ с предишния */
    cout<<"Max uspeh e: "<<max;
    return 0;
}

```

**Пример 4:** Програмата чете дневната температура за всеки ден от месеца и след това докладва средно-месечната температура, най-горещия и най-студения ден

```

#include <iostream>
using namespace std;
int main(void)
{
    int temp[31], i, min, max, avg;
    int days;
    cout<<"How many days in the month? ";

```

```

cin>>days;
for(i=0; i<days; i++) {
    cout<<"Enter noonday temperature for day ";
    cout<<i+1;
    cin>>temp[i];
}
//намиране на средната температура
avg = 0;
for(i=0; i<days; i++)
    avg = avg + temp[i];
cout<<"Average temperature: "<< avg/days;
// намиране на min и max температура
min = 200; // инициализира min и max
max = 0;
for(i=0; i<days; i++) {
    if(min>temp[i]) min = temp[i];
    if(max<temp[i]) max = temp[i];
}
cout<<"Minimum temperature: "<<min;
cout<<"Maximum temperature: "<<max;
return 0;
}

```

**Пример 5 :** Намиране минимален елемент на масив

```

#include <iostream>
using namespace std;
int main()
{
    int array[100], minimum, broi, c, location = 1;
    cout<<"Broi elementi na masiva  ";
    cin>>broi;
    cout<<"napishi elementite: ";
    for ( c = 0 ; c < broi ; c++ )
        cin>>array[c];
    minimum = array[0];
    for ( c = 1 ; c < broi ; c++ )
    {
        if ( array[c] < minimum )
        {

```

```

        minimum = array[c];
        location = c+1;
    }
}
cout<<"Minimalniat element se namira pod nomer : ";
cout<<location<<endl;
cout<<"stoinostta mu e : ";
cout<<minimum;
return 0;
}

```

**Пример 6 :** Намиране максимален елемент на масив

```

#include <iostream>
using namespace std;
int main()
{
    int array[100], maximum, size, c, location = 1;
    cout<<"broi elementi na masiva: ";
    cin>>size;
    cout<<"Napishi elementite: ";
    for (c = 0; c < size; c++)
        cin>>array[c];
    maximum = array[0];
    for (c = 1; c < size; c++)
    {
        if (array[c] > maximum)
        {
            maximum = array[c];
            location = c+1;
        }
    }
    cout<<"Max element se namira pod nomer: ";
    cout<<location;
    cout<<" i stoinostta mu e : ";
    cout<<maximum;
    return 0;
}

```

**Търсене на елемент в масив**

**Зад 7.** Да се състави програма, която въвежда в масив средният годишен успех на всеки ученик в един клас. Да се провери

съществува ли ученик с успех x, предварително зададен от клавиатурата.

```
#include<iostream>
using namespace std;
int main()
{
float arr[30],x; /*дава възможност за въвеждане успеха на 30
ученика и x -успех на един от учениците*/
int i,n; //i-бройча на номер на ученик, n- бр. ученици
cout<<"Vavedete broi uchenici  " ;
cin>>n;
cout<<"Vavedete uspehat na vseki edin ot tiah ";
for(i=0;i<n;i++) //бройча брой учениците
    cin>>arr[i]; /* дава възможност за въвеждане успеха на всеки
един ученик*/
cout<<"uspehat, koito shte tarsim e x=  " ;
cin>>x;
i=0;
while((arr[i]!=x)&&(i<n)) i++;
if(arr[i]==x)
    cout<<"Nameren e uchenik s tozi uspeh"<<x;
else
    cout<<"Niama uchenik s tozi uspeh ";
return 0;
}
```

**Размяна стойностите на два елемента от масива**

**Зад 8. Даден е масив от цели числа:**

**arr[n]={10,8,32,6,1}**

**Да се сортира масивът във възходящ ред и да се изведе на екрана.**

**За решението на задачата може да се приложат 2 метода за сортиране:**

**I) Метод на пряка селекция**

**Този метод реализира следната идея:**

- 1. Намира се минималният ел-т в редицата от числа и се разменя с първият**
- 2. В редицата без първият ел-т , действията се повтарят – намира се мин. ел-т и се разменя с вторият**

```
#include<iostream>
```

```

using namespace std;
int main()
{
    const int n=5;
    int arr[n]={20,8,32,6,1},min,k,swap;
    for(int i=0;i<n-1;i++)
    {
        k=i;
        min=arr[i];
        for(int j=i+1;j<n;j++)
            if(arr[j]<min)
            {
                min=arr[j]; //намира мин ел-т в неподредената редица
                k=j;
            }
        swap=arr[k]; /*разменя миним. елемент с първия неподреден
елемент от поредицата*/
        arr[k]=arr[i];
        arr[i]=swap;
    }
    for(int k=0;k<n;k++)
        cout<<"arr"<<k<<"]="<<arr[k]<<endl;
return 0;
}

```

## II) Метод на мехурчето

В редицата от числа се сравняват всеки два съседни елемента ( първият с втория, втория с третия и т.н) Ако елементът с по-малък индекс е по-голям, те се разменят. Така след първото преглеждане на масива елементът с най-голяма стойност заема последното място в редицата.

```

#include<iostream>
using namespace std;
int main()
{
    const int n=5;
    int arr[n]={20,8,32,6,1},swap;
    for(int i=i+1;i<n-1;i++)
        for(int j=0;j<n-1-1;j++)

```

```

        if(arr[j]>arr[j+1]) //сравнява се със съседните елементи
        {
            swap=arr[j]; //разменят се съседните елементи
            arr[j]=arr[j+1];
            arr[j+1]=swap;
        }
        for(int k=0;k<n;k++)
            cout<<"arr"<<k<<"]="<<arr[k]<<endl;
return 0;
}

```

**Зад 9.** Да се напише програма, която декларира и въвежда масив от символи, след което извежда колко пъти символът 'а' се среща в масива

```

#include<iostream>
using namespace std;
int main()
{
    int k,br=0;
    cin>>k; //въвеждаме бр. на символите
    char s[100]; //въвеждаме символите
    for(int i=0;i<=k;i++)
        cin>>s[i];
    for(int j=0;j<k;j++) //намираме бр. на символите 'а'
        if(s[j]=='a') br++;
    cout<<br<<endl;
    return 0;
}

```

**Зад 10.** Да се напише програма, която въвежда от клавиатурата цяло число  $1 < n < 11$ , редица от числа  $a_0, a_1, a_2, \dots, a_{n-1}$  и цяло число  $x$ . Програмата да проверява дали  $x$  се съдържа в редицата и да извежда подходящо съобщение.

```

#include<iostream>
using namespace std;
int main()
{
    int arr[10],n,x,flag=0;
    cin>>n; //въвеждане бр. на елементите на масива
    for(int i=0;i<n;i++)

```

```

    cin>>arr[i]; //въвеждане на елементите на масива
    cin>>x; //въвеждане на число x
for(int j=0;j<n;j++)
if(arr[j]==x) //при срещане на x, флагът приема 1 и цикълът спира
{
    flag=1;
    break;
}
if(flag)cout<<"yes"<<endl;
else cout<<"no"<<endl;
return 0;
}

```

Зад 11. Да се напише програма, която въвежда от клавиатурата цяло число  $2 \leq n \leq 10$  и редица от реални числа  $a_0, a_1, a_2 \dots a_{n-1}$ , след което намира и извежда сумата на отрицателните елементи от редицата.

```

#include<iostream>
using namespace std;
int main()
{
    int n;
    double arr[10],sum=0;
    cin>>n; //въвежда броя на елементите
    for(int i=0;i<n;i++);
    cin>>arr[i]; //въвежда елементите на масива
    for(int j=0;j<n;j++)
        if(arr[j]<0) sum+=arr[j];
    cout<<sum<<endl; //извежда сумата на отрицателните ел-ти
    return 0;
}

```

Зад 12. Да се напише програма, която въвежда от клавиатурата цяло число  $1 < n < 36$ , редица от числа  $a_0, a_1, a_2 \dots a_{n-1}$  и цяло число  $x$ . Програмата да извежда позицията на първото срещане на  $x$  в редицата. Ако  $x$  не се съдържа в редицата, да се извежда подходящо съобщение.

```

#include<iostream>
using namespace std;
int main()
{
    int arr[35],n,x,pos=-1;

```



```

cin>>n; //въвежда броя на елементите
for(int i=0;i<n;i++)
cin>>arr[i]; //въвежда масива от елементи
cin>>x; //въвежда цяло число
for(int j=0;j<n;j++)
    if(arr[j]==x) //ако цялото число x се намира в редицата от масива
    {
        pos=j; //присвоява се на позицията първото срещнато число x
        break;
    }
if(pos!=-1)
cout<<(pos+1)<<endl;
else
    cout<<"no"<<endl;
return 0;
}

```

**Зад 13.** Да се напише програма, която въвежда от клавиатурата цяло число  $2 < n < 30$  и редица от цели числа  $a_0, a_1, a_2, \dots, a_{n-1}$ , след което намира и извежда броя на четните числа от редицата.

```

#include<iostream>
using namespace std;
int main()
{
    int n,arr[29],br=0;
    cin>>n; //въвеждане бр. елементи на масива
    for(int i=0;i<n;i++) //цикъл, в който се броят четните числа
        cin>>arr[i];
    for(int j=0;j<n;j++)
        if(arr[j]%2==0)br++;
    cout<<br<<endl;
    return 0;
}

```

**Зад 14.** Да се напише програма, която въвежда от клавиатурата цяло число  $2 < n < 25$  и редица от цели числа  $a_0, a_1, a_2, \dots, a_{n-1}$ , след което намира и извежда средноаритметичното на положителните елементи от редицата.

```

#include<iostream>
using namespace std;

```

```

int main()
{
int n,br=0;
double arr[24],sum=0;
cin>>n; //въвеждане бр. елементи на масива
for(int i=0;i<n;i++)
    cin>>arr[i]; //въвеждане на елементите на масива
for(int j=0;j<n;j++) //цикъл, намираш сумата на положителните
    числа
{
    if(arr[j]>0)
    {
        br++;
        sum+=arr[j];
    }
}
if (br!=0)
    cout<<(sum/br)<<endl; //извежда средно аритметичното
else
    cout<<0<<endl;
return 0;
}

```

**Зад 15.** Да се напише програма, която въвежда от клавиатурата цяло число  $2 < n < 40$  и редица от цели числа  $a_0, a_1, a_2, \dots, a_{n-1}$ , след което намира и извежда произведението на тези числа от редицата, които са кратни на 7. Ако няма такива, да изведе N.

```

#include<iostream>
using namespace std;
int main()
{
int n,pr=1,arr[39],flag=0;
cin>>n; //въвежда броя на ел-те на масива
for(int i=0;i<n;i++) //цикъл за намиране произведението на
    числата кратни на 7
    cin>>arr[i]; //въвежда елементите на масива
for(int j=0;j<n;j++)
    if(arr[j]%7==0) //проверка дали числото е кратно на 7
    {
        pr*=arr[j];
    }
}

```

```

        flag=1;
    }
    if(flag)
        cout<<pr<<endl;
    else
        cout<<"no"<<endl;
    return 0;
}

```

## Многомерни масиви

В основата си двумерният масив е масив от едномерни масиви и е най-лесно да се представи във вид на редове и колони

```

r   5  2  3
    1  6  8
    9 10  4

```

```

int m [ ] [ ]
      ред  колона

```

Пример 1: Програмата изписва масива m[3][3]

```

#include <iostream>
using namespace std;
int main() {
    int m[3][3] = {
        {1,5,2},
        {4,6,9},
        {8,3,7}
    };
    int row = 0;
    int col = 0;
    for( row = 0 ; row < 3; row++)
    {
        for( col=0; col < 3; col++ )
        {
            cout<<row<<col <<m[row][col];
        }
        cout<<" ";
    }
    return 0;
}

```

```
}
```

**Резултат:**

**001015022 104116129 208213227 /ред, колона, елемент/**

**Пример 2:** Изваждане на елементите на матрици

```
#include <iostream>
using namespace std;
int main()
{
    int m, n, c, d, first[10][10], second[10][10], difference[10][10];

    cout<<"Enter the number of rows and columns of matrix";
    cin>>m>>n;
    cout<<"Enter the elements of first matrix";
    for (c = 0; c < m; c++)
        for (d = 0 ; d < n; d++)
            cin>>first[c][d];
    cout<<"Enter the elements of second matrix";
    for (c = 0; c < m; c++)
        for (d = 0; d < n; d++)
            cin>>second[c][d];
    for (c = 0; c < m; c++)
        for (d = 0; d < n; d++)
            difference[c][d] = first[c][d] - second[c][d];
    cout<<"difference of entered matrices:";
    for (c = 0; c < m; c++)
    {
        for (d = 0; d < n; d++)
            cout<<difference[c][d];
        cout<<" ";
    }
    return 0;
}
```

**Пример 3:** Тази програма създава таблица за справки. Двумерен масив 5x2 се инициализира така, че първият елемент от всеки ред е номерът на файловият сървър от мрежата, а вторият елемент съдържа броя на потребителите, свързани с този сървър. Програмата дава възможност на потребителя да въвежда номера на сървъра, след това прави справка с таблицата и докладва броя на потребителите:

```
#include <iostream>
```

```

        using namespace std;
int main(void)
{
    int ServerUsers[5][2] = {
        1, 14,
        2, 28,
        3, 19,
        4, 8,
        5, 15
    };
    int server;
    int i;
    cout<<"Enter the server number: ";
    cin>>server;
    //look it up in the table
    for(i=0; i<5; i++)
        if(server == ServerUsers[i][0]) {
            cout<<"There are users on server "<<ServerUsers[i][1]<< server;
            break;
        }
    //report error if not found
    if(i==5) cout<<"Server not listed.\n";
    return 0;
}

```

## ПОСТРОЯВАНЕ НА МАСИВИ ОТ НИЗОВЕ

Масивите от низове, наречени низови таблици, са много популярни в програмирането на C. Те се създават като двумерен масив.

**Пример 9:** Програмата въвежда 10 низа и дава възможност за показването им един по един в произволен ред. За да спрете програмата, въведете отрицателно число:

```

#include <iostream>
using namespace std;
int main(void)
{
    char text[10][80];
    int i;
    for(i=0; i<10; i++) {
        cout<<i+1;
        cin>>text[i];
    }
}

```

```

}
do {
    cout<<"Enter number of string (1-10) : ";
    cin>>i;
    i--; // adjust value to match array index
    if(i>=0 && i<10) cout<<text[i];
} while(i>=0);
return 0;
}

```

## ТЕСТ

**Зад.1** Напишете определение за едномерен масив и дайте пример как се декларира

**Зад. 2** Какъв ще бъде изходът на следната програма:

```

#include <iostream>
using namespace std;
int main(void)
{
    int sqrs[10];
    int i;
    for(i=1; i<11; i++) sqrs[i-1] = i*i;
    for(i=0; i<10; i++)
        cout<< sqrs[i]<<" ";
    return 0;
}

```

**Зад. 3** Как ще изглежда масива P[3][2] като въведете произволни стойности на елементите на този масив.

**Зад. 4** Покажете как ще се инициализира масив S[5][2]

**Зад. 5** Като гледате задачата по-долу, напишете нейното условие и коментарите на всеки ред

```

#include<iostream>
using namespace std;
int main()
{
    int k,br=0;
    cin>>k;

```

```
char s[100];  
for(int i=0;i<=k;i++)  
    cin>>s[i];  
for(int j=0;j<k;j++)  
    if(s[j]=='a') br++;  
cout<<br<<endl;  
return 0;  
}
```