

# Многомерни масиви



# Деклариране на многомерни масиви

---

тип име [брой1][брой2];

- **тип** – типа на елементите в масива; нарича се базов тип
- **име** – име на масива
- **брой1, брой2** - броят на елементите във всяка посока (*например по редове и колони*)
- **пример:**  
`int a[10][5];`

# Стандартни алгоритми с многомерни масиви

---

Алгоритмите за работа с многомерни масиви са същите както при едномерни масиви, с единствената разлика, че имаме няколко вложени цикъла за обхождане на елементите на масива.

# Въвеждане на масив от клавиатурата

---

```
1. int a[100][100];
2. int rcount, ccount;
3. cout<<"Въведете брой редове и колони:";
4. cin>>rcount>>ccount;
5. cout<<"Въведете елементите на масива:";
6. for(int r=0; r<rcount; r++)
7.     for(int c=0; c<ccount; c++) {
8.         cout<<"Елемент [" , r, "]"[" , c, "]=";
9.         cin>>a[r][c];
10.     }
```

# Отпечатване на масив на екрана

---

```
1. cout<<"Съдържание на масива:";
2. for(int r=0; r<rcount; r++) {
3.     for(int c=0; c<ccount; c++)
4.         cout<<setw(6)<<a[r][c]<<' ';
5.     cout<<endl;
6. }
```

# Намиране на сума на целия масив

---

1. `int sum=0;`
2. `for(int r=0; r<rcount; r++)`
3.     `for(int c=0; c<ccount; c++)`
4.         `sum+=a[r][c];`
5. `cout<<"Сумата на всички елементи е:"`  
   `<<sum<<endl;`

# Намиране на сума по редове

---

```
1. for(int r=0; r<rcount; r++) {  
2.     int sum=0;  
3.     for(int c=0; c<ccount; c++)  
4.         sum+=a[r][c];  
5.     cout<<"Сумата за ред "<<r  
        <<" е "<<sum<<endl;  
6. }
```

# Намиране на сума по колони

---

```
1. for(int c=0; c<ccount; c++) {  
2.     int sum=0;  
3.     for(int r=0; r<rcount; r++)  
4.         sum+=a[r][c];  
5.     cout<<"Сумата за колона "<<c  
        <<" е "<<sum<<endl;  
6. }
```



# Особености при многомерни масиви

---

Трябва да се обърне специално внимание да не се разменя мястото на различните индекси и граници в масива (например на номерът и броят на редовете и колоните при двумерен масив)

Край

