

**Масиви**

# Що є масив?

Масив являє собою сукупність даних,  
що організована певним чином.

Тобто масив – це структура даних.

# Властивості масивів(1)

- масив складається з елементів, які мають однаковий тип;
- елементи масиву послідовно, без проміжків, розташовані в одній ділянці оперативної пам'яті;
- кожен з елементів масиву має свій порядковий номер, що зветься індексом;
- нумерація елементів починається з 0;
- до елементів масиву можна звертатися використовуючи ім'я масиву і індекс у квадратних дужках.

# Властивості масивів(2)

- масив може бути одновимірним, або багатовимірним,
- У двовимірному масиві кожний з елементів є одновимірним масивом;
- У тривимірному масиві кожний з елементів є двовимірним масивом;

# Оголошення одновимірного масиву

<тип елементів> <ім'я масиву> [<кількість елементів>];

Приклади оголошень масивів:

```
int arr [10];
```

```
int A [10] = {2, 5, 10};
```

```
int A [] = {2, 5, 10, 3, 6, 0, 9, 4, 5, 7};
```

# Доступ до елементів масиву

Для доступу до елементів масиву використовується синтаксична конструкція, що складається з імені масиву та індексу, який записується у квадратних дужках.

Наприклад,  $A[i]$ ,  $A[0]$ ,  $A[9]$ .

Індексовані елементи масиву можуть бути використані так само, як і прості змінні. Вони можуть бути операндами у виразах, їм можна присвоювати значення, відповідні їх типу.

# Доступ до елементів масиву

```
int arr[ ] = {2, 5, 10, 3, 6, 0, 9, 4, 5, 7};
```

Імена елементів масиву



Значення елементів масиву

# Масиви символів

- У мові С символний рядок розглядається як масив символів.
- Для масиву символів у пам'яті виділяється ділянка, розмір якої на один байт більше, ніж кількість символів у рядку.
- У цей додатковий байт записується ознака кінця рядка - символ '\0'.



# Оголошення рядка символів

`char <ім'я символного рядка> [<кількість байтів >];`

Приклад:

**`char str[80];`**

**`char str[10] ={'H', 'e', 'l', 'l', 'o', '!', '\0'};`**

**`char str[10] = "Hello!";`**

**`char str[] = "Hello!";`**

# Особливості роботи з масивами в C, C++

- Ніякого контролю за значеннями індексів, що використовуються для доступу до елементів масиву, нема.
- Ви можете звернутися до «елементу масиву» з номером, який більше максимального, але отримаєте невідомо що.
- Ще гірше, якщо ви щось напишете за цим номером

# Масиви і функції

- У мовах C, C++ не існує такого типу як «масив». Не можна написати, наприклад, `int[ ]`, як тип.
- З цієї причини у функції не можна вказати масив, як тип того, що повертається функцією.
- Але масив можна повернути через параметри функції.

# Масиви, як параметри функцій

- Масив, як формальний параметр функції, оголошується майже так само, як і проста змінна, але після імені параметру слід поставити пусті квадратні дужки. Тип цього параметру має співпадати з типом елементів масиву.

# Масиви, як параметри функцій

- Масиви у функції завжди передаються за посиланням.
- Але символ & ставити не потрібно, бо ім'я масиву і так є адресом першого елемента масиву.

# Масиви, як параметри функцій

- Масив «не знає», скільки у нього елементів.
- Передаючи масив до функції слід передавати і кількість елементів масиву, що має бути оброблена.
- Це число не повинно перевищувати кількість елементів під які виділено пам'ять під час оголошення масиву.

# Приклад функції для створення масиву чисел Фібоначі

```
void fibo(int ar[], int size) {  
    ar[0] = 0;  
    ar[1] = 1;  
    for (int i=2 ; i<size; i++)  
        ar[i] = ar[i - 2] + ar[ i - 1 ];  
}
```

# Виклик функції з масивом

```
#define MAX 100
void fibo(int ar[], int size);
int main() {
    int fiboArray[MAX];
    cout<<"Скільки чисел сформувати: ";
    int n; cin>>n;
    fibo(fiboArray,n);
    for(int i = 0; i < n; i++)
        cout<<fiboArray[i]<<"\n";
}
```



# ФУНКЦІЇ ОБРОБКИ МАСИВІВ ЧИСЕЛ

# Формування масиву випадкових чисел

```
void createRndAr(int ar[], int size, int mod){  
    for(int i=0; i<size; i++){  
        ar[i] = rand()%mod;  
    }  
}
```

# Операції над масивами

- формування масиву випадкових чисел;
- введення/виведення масиву чисел;
- пошук суми елементів масиву;
- пошук максимального та мінімального елементів масиву та їх індексів;
- пошук індексу елемента масиву за його значенням;

# Введення чисел з консолі(1)

```
getArFromConByStep(int ar[], int &n){  
    cout<<"Скільки чисел ввести? ";  
    cin>>n;  
    for(int i=0; i<n; i++){  
        cout<<"Введіть число № "  
            <<i<<" => ";  
        cin>>ar[i];  
    }  
}
```

# Введення чисел з консолі (2)

```
void getArFromConAsLine(int ar[], int &size){  
    char s[80];  
    cout<<"Введіть елементи масиву"  
        <<" через пробіли"<<endl;  
    cin.getline(s,80); //gets(s);  
    strToArr(s,ar,size);  
}
```

# Перетворення рядка у масив

```
void strToArr(char s[], int ar[], int &size){
    int i(0); size=0;
    forever{
        //Цикл пошуку цифри
        while( s[i] != '\0' && !isdigit(s[i])) i++;
        if(s[i]=='\0') return;
        //Формуємо наступне число
        for(ar[size]=0; isdigit(s[i]); i++ )
            ar[size]=ar[size]*10+(s[i]-48);
        size++;
    }
}
```

# Особливості використання cin.getline() та gets()

```
forever{
    system("cls");
    cout<<"1. Масив випадкових чисел.\n";
    cout<<"2. Input array as string.\n";
    ...
    cout<<"\nYou chois is: ";
    cin>>v; getchar();
    switch(v){
        ...
        case 2: getArFromConAsLine(arr, size);
                arToConsole(arr,size);
                break;
        ...
    }
```

# Виведення масиву на консоль

```
void arToConsole(int ar[], int size){  
    for(int i=0; i<size; i++){  
        cout<<ar[i];  
        if(i<size-1) cout<<" ";  
    }  
    cout<<endl;  
}
```



# Обчислення суми елементів масиву

```
int sumAr(int ar[ ],int n) {  
    int sum=0;  
    for (int i = 0; i < n; ++i) {  
        sum+=ar[ i ];  
    }  
    return sum;  
}
```

# Пошук найбільшого елемента у масиві

```
int maxInAr(int m[], int size){  
    int mx=INT_MIN;  
    for(int i=0; i<size;i++)  
        if (m[i]>mx) mx=m[i];  
    return mx;  
}
```

# Ще раз про виклик функції

```
void getArFromConAsLine(int m[], int &size);  
int maxInAr(int m[], int size);  
int main(){  
    int arr[100], sizeArr;  
    getArFromConAsLine(arr, sizeArr);  
    cout<<"Найбільше число у масиві: "  
        <<maxInAr(arr, sizeArr) <<"\n";  
    return 0;  
}
```

# Пошук найменшого елемента у масиві та його індексу

```
void idxMn(int m[], int size, int &mn, int &idx){  
    mn=INT_MAX; idx= -1;  
    for(int i=0; i<size;i++)  
        if (m[i]<mn) {  
            mn=m[i];  
            idx = i;  
        }  
    }  
}
```

# І знов про виклик функції

```
void idxMn(int m[], int sz, int &mn, int &idx);  
int main(){  
    int arr[100], sizeArr;  
    getArFromConAsLine(arr, sizeArr);  
    int mn, idx;  
    idxMn(arr, sizeArr, mn, idx);  
    cout<<"min="<<mn  
        <<" ; index="<<idx<<"\n";  
    return 0;  
}
```

# Визначення позиції елемента у масиві

```
int findPos(int x, int m[], int size){  
    for(int i=0; i<size; i++)  
        if(m[i]==x) return i;  
    return -1;  
}
```

# ПЕРЕСТАНОВКИ ЕЛЕМЕНТІВ МАСИВІВ

# Вилучення елемента із масиву

```
void delElement(int element, int ar[], int &size){
    int i=0;
    while(i < size){
        if(ar[i] == element) {
            size = size - 1;
            for(int j = i; j<size; j++)
                ar[ j ] = ar[ j+1 ];
        }
        else i++;
    }
}
```



# Вставка елемента до масиву

```
void insert(int x,int pos, int ar[], int &size){  
    if(pos<0 || pos>size)return;  
    for(int i=size;i>pos;i--)  
        ar[i]=ar[i-1];  
    ar[pos]=x;  
    size++;  
}
```

# Переворот массиву

```
void transArray(int ar[], int size){  
    for(int i=0; i<size/2; i++){  
        int tmp = ar[i];  
        ar[i] = ar[size-1-i];  
        ar[size-1-i] = tmp;  
    }  
}
```

# Циклічний зсув масиву праворуч

```
void shiftRight(int ar[], int size){  
    int buf=ar[size-1];  
    for(int i=size-1;i>0;i--)  
        ar[i]=ar[i-1];  
    ar[0]=buf;  
}
```

Інші задачі

# Масив у ролі лічильника

```
// Тут ми підраховуємо скільки разів з'явиться  
// кожне з 10 (від 0 до 9) випадкових чисел  
// у вибірці обсягом v
```

```
void arrayAsCount(int ar[], int v){  
    for(int i=0;i<10;i++)  
        ar[i]=0;  
    for(int i=0;i<v;i++)  
        ar[rand()%10]++;  
}
```

# Масив накопичених значень (інтеграл по індексу)

```
void accumAr(int ar[], int size){  
    for(int i=1; i<size; i++)  
        ar[i]=ar[i-1]+ar[i];  
}
```

# Обробка рядків СИМВОЛІВ

# Операції над рядками СИМВОЛІВ

- Пошук позиції символу у рядку;
- пошук позиції під рядка у рядку;
- копіювання частини рядка;
- вилучення частини рядка;
- вставка підрядка у рядок;
- перетворення числа у рядок символів;
- перетворення рядка символів у число.



# Пошук позиції символу у рядку

```
int posChar(char str[], char ch){  
    for(uint i=0; str[i] != '\0'; i++)  
        if(str[i]==ch) return i;  
    return -1;  
}
```

# Пошук позиції під рядка у рядку

```
int posSubstr(char str[], char sub[]){  
    for(uint i=0; i<=strlen(str)-strlen(sub); i++){  
        uint j=0;  
        while(str[i+j]==sub[j] && j<strlen(sub))  
            j++;  
        if (j==strlen(sub))return i;  
    }  
    return -1;  
}
```

# Копіювання частини рядка

```
void subStr(char str[], char sub[],
            uint pos, uint n){
    //Якщо позиція за межами рядка
    if (pos>strlen(str)) n=0;
    //Якщо кількість більша за можливе
    else if(pos+n > strlen(str))
        n=strlen(str)- pos;
    for(uint i=0; i<n; i++)
        sub[i]=str[pos+i];
    sub[n] = '\0';
}
```

# Вилучення частини рядка

```
void del(char str[], uint pos, uint n){  
    //Якщо позиція за межами рядка  
    if (pos >= strlen(str)) return;  
    //Якщо кількість більша за можливе  
    if(pos+n > strlen(str))  
        n = strlen(str) - pos;  
    for(uint i=pos+n; i<=strlen(str); i++)  
        str[i-n]=str[i];  
}
```

# Вставка під рядка у рядок

```
void insert(char str[],char sub[], uint pos){  
    //Якщо позиція за межами рядка  
    if (pos>strlen(str)) return;  
    //Зсув праворуч частини рядка  
    for(uint i=strlen(str); i>=pos ;i--)  
        str[i+strlen(sub)]=str[i];  
    //Вставка  
    for(uint i=0; i<strlen(sub); i++)  
        str[pos+i]=sub[i];  
}
```

# Ціле число у рядок СИМВОЛІВ

```
void intToStr(int x, char str[]){
    int i=0, z=abs(x); str[0]='\0';
    do{ //Звільняємо місце для цифри
        for(int j=i++; j>=0; j--) str[j+1]=str[j];
        str[0]=z%10+48; //Наступна цифра
    }while(z/=10);
    if(x<0){
        for(int j=i; j>=0; j--) str[j+1]=str[j];
        str[0]='-';
    }
}
```

# Перетворення рядка символів у число

```
int strToInt(char s[]){  
    int x=0, i=0;  
    if(s[0] == '-') i=1;  
    for( ; isdigit(s[i]); i++ )  
        x=x*10+(s[i]-48);  
    return s[0]=='-' ? -x : x;  
}
```