

## Лабораторна робота №22

### Показчики

**Мета:** набути навичок роботи з показчиками.

#### Література

Войтенко В. В., Морозов А. В. С\С++ Практика програмування. Навчально методичний посібник - Житомир: ЖДТУ, 2003. – 324 с.

#### Зміст роботи

**Завдання 1.** Написати програму з використанням показчиків.

1	Вивести адреси елементів масиву 4x4 , що знаходяться нижче головної діагоналі.
2	Сформувати масив із мінімальних елементів кожного стовпця матриці 4x4 і знайти адреси його елементів
3	Вивести адреси елементів матриці 4x4 , що знаходяться вище головної діагоналі.
4	Сформувати масив із максимальних елементів кожного стовпця матриці 4x4 і знайти адреси його елементів
5	Знайти адреси елементів матриці 4x4, що дорівнюють 0 .
6	Поміняти місцями перший і останній елемент матриці 4x4 і знайти його розмір в байтах.
7	Знайти адреси мінімальних елементів кожного рядка матриці 4x4 і сформувати із цих елементів одновимірний масив.
8	Вивести адреси елементів матриці 4x4 з парною сумою індексів
9	Визначити адреси елементів масиву розмірністю 4x4, що знаходяться на головній діагоналі.
10	Знайти максимальний елемент матриці 4x4 і сформувати новий масив, кожен елемент якого – це добуток елемента початкового масиву на максимальний елемент. Вивести адреси елементів нового масиву.

**Завдання 2.** Написати програму, в якій потрібно створити двовимірний динамічний масив і виконати обробку даних.

1	Видалити із масиву рядок із заданим номером і стовпець, де заходиться максимальний елемент.
2	Видалити із масиву стовпець із заданим номером і рядок, де заходиться мінімальний елемент.
3	Додати до масиву рядок із заданим номером і видалити стовпець, де заходиться мінімальний елемент.

4	Додати до масиву стовпець із заданим номером і видалити перший рядок, де заходиться 0 елемент.
5	Видалити із масиву рядок із заданим номером і рядок, де заходиться максимальний елемент, якщо рядки співпадають, видати повідомлення
6	Видалити із масиву рядок і стовпець із заданим номером
7	Додати до масиву рядок і стовпець із заданим номером
8	Видалити із масиву всі рядки, в яких зустрічається задане число.
9	Видалити із масиву всі стовпці, в яких зустрічається задане число
10	Видалити із масиву рядок і стовпець, на перетині яких знаходиться мінімальний елемент.
11	Видалити із масиву рядок який знаходиться до рядка з максимальним елементом і додати стовпець після стовпця з максимальним елементом
12	Видалити із масиву рядок до і рядок після рядка з мінімальним елементом
13	Видалити із масиву стовпчик до і стовпчик після стовпчика з мінімальним елементом
14	Видалити із масиву всі рядки, в яких зустрічається максимальне і мінімальне число
15	Видалити із масиву рядок і стовпець, на перетині яких знаходиться максимальний елемент

### Методичні рекомендації

Багатомірний масив у мові Сі розглядається як сукупність одновимірних масивів (його рядків). Для звертання до елементів багатомірного масиву можна використовувати нуль і більш індексів (індексних виразів):

*ім'я-масиву [вираз1][вираз2] ...*

Приклад оголошення двовимірного масиву значень типу *int*: *int a[n][m];*

Цей масив складається з *n* одновимірних масивів (рядків), у кожному з яких утримується *m* елементів (стовпців). При роботі з цим двовимірним масивом можна використовувати одно або 2 індексних вирази. Наприклад:

*a[i][j]*- містить 2 індекси; використовується для звертання до елемента *i*-рядка, *j*-стовпця масиву; обчислюються індексні вирази, визначається адреса елемента масиву і вилучається його значення;

$a[i]$  - містить 1 індекс; визначає адресу одновимірного масиву: адреса початку  $i$ -рядка масиву;

$a$  - не містить індексу  $i$  визначає адресу масиву, його нульового елемента.

Таким чином, звертання до двовимірних масивів за допомогою імені  $i$  тільки одного індексу визначає покажчик на початок відповідного рядка масиву (адреса його нульового елемента). Наприклад:

$$a[0] = \&a[0][0] = a + 0 * n * \text{sizeof}(int);$$

$$a[1] = \&a[1][0] = a + 1 * n * \text{sizeof}(int);$$

$$a[i] = \&a[i][0] = a + i * n * \text{sizeof}(int);$$

В ОП елементи масиву  $a$  розташовуються таким чином, що швидше всіх змінюється самий правий індекс, тобто в послідовності:  $a[0][0]$   $a[0][1]$   $a[0][2]$  ...  $a[2][2]$   $a[2][3]$ .

При цьому для звертання до масиву  $a$  можна використовувати імена:

$\&a = a = \&a[0][0] = *a$  - адреса  $a[0][0]$  - елемента 0 рядка 0 стовпця масиву  $a$ ;

$**a = *(\&a[0][0]) = a[0][0]$  - значення елемента 0 рядка 0 стовпця масиву  $a$ ;

$$a[i] = (a + i) = *(a + i) = \&a[i][0] - \text{адреса елемента } i\text{-рядка } 0\text{-стовпця};$$

$$*a[i] = **a + i = *(\&a[i]) = a[i][0] - \text{значення } 0 \text{ елемента } i\text{-рядка};$$

$$a[i][j] = (*(a + i) + j) = *(a[i] + j) = a[i][j] - \text{значення елемента } i\text{-рядка } j\text{-стовпця масиву } a;$$

$$\text{де: } (a + i) = *(a + i) = a[i] - \text{адреса } 0 \text{ елемента } i\text{-рядка} = \&a[i][0];$$

$$(*(a + i) + j) - \text{адреса } j\text{-елемента } i\text{-рядка} = \&a[i][j];$$

$$**a + j - \text{значення } j\text{-елемента } i\text{-рядка} = a[i][j].$$

Значення адреси початку  $i$ -рядка (адреси 0-елемента  $i$ -рядка) на машинному рівні формується у виді:

$a[i] = a + i == (a + i * n * \text{sizeof}(int))$ , де  $n$  - кількість значень в одному рядку.

Таким чином, адреса  $(i+1)$ -рядка знаходиться від  $i$ -рядка на  $(n * \text{sizeof}(int))$  байтів, тобто на відстань одного рядка масиву.

Вираз  $a[i][j]$  компілятор Сі переводить в еквівалентний вираз:  $*(a + i) + j$ . Запис  $a[i][j]$  більш традиційний у математиці і більш наочний.

До елементів двовимірного масиву можна звернутися і за допомогою скалярного покажчика на масив. Наприклад, після оголошення:

$$\text{int } a[n][m], *p = a;$$

$$*(p + i * m + j) - \text{значення } j - \text{елемента } i\text{-рядка};$$

де:  $m$  - кількість елементів у рядку;

$i*m + j - a[i][j]$ - елемент відносно початку масиву  $a$ .

У наведеній нижче програмі використовуються звертання до  $p[i][j]$  - елементів масиву у вигляді:  $*(p+i*m+j)$  - адреса  $p[i][j]$ -елемента масиву.

```
#include ...
void main()
{
    srand(time(NULL));
    int *p, n=4, m=5;
    p = (int*) malloc (n*m * sizeof (int));
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < m; j++ )
        {
            *(p+i*m+j) = rand()%10;
            printf("%3d", *(p+i*m+j));
        }
        printf ("\n" );
    }
    free( p );
    getch();
}
```

### *Динамічне виділення пам'яті під масиви*

Для створення двовимірного масиву спочатку потрібно розподілити пам'ять для масиву покажчиків на одномірні масиви, а потім розподіляти пам'ять для одномірних масивів. Наприклад, потрібно створити масив  $arr[n][m]$ :

```
#include ...
void main()
{
    srand(time(NULL));
    int **arr=NULL;
    int n,m;
    printf("n= ");scanf("%d",&n);
    printf("m= ");scanf("%d",&m);
    /*arr=(int **)calloc(n,sizeof(int *)); // calloc
    for (int i=0; i<n; i++)
        arr[i]=(int *)calloc(m,sizeof(int));*/

    arr = (int**) malloc(n * sizeof(int*)); // malloc
    for (int i = 0; i < n; i++)
        arr[i] = (int*) malloc(m * sizeof(int));
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < m; j++ )
        {
            arr[i][j]= rand()%10;
            printf("%3d", arr[i][j]);
        }
        printf ("\n" );
    }
    for (int i = 0; i < n; i++)
        free(arr[i]);
    free(arr);
    getch();
}
```

## Контрольні питання:

1. Що таке покажчик?
2. Як можна використовувати покажчики у багатовимірних масивах?
3. Що таке динамічна пам'ять?
4. Як оголосити одновимірний динамічний масив?
5. Як оголосити динамічний двовимірний масив?
6. В чому різниця між статичними та динамічними масивами?
7. Коли і звідки видяється пам'ять для статичних і динамічних масивів?
8. Наведіть фрагмент коду для заповнення динамічного масиву цілими (дійсними) випадковими числами.
9. Наведіть фрагмент коду для обчислення суми (добутку) елементів одновимірного динамічного масиву?
10. Наведіть фрагмент коду для підрахунку суми (добутку) елементів певного рядка (стовпця) двовимірного масиву?

## Завдання на самостійну роботу:

Опрацювати лекцію №14, 15, 16

1. Оголошено змінні: `char c; int * pi; float ** ppf;`

Вкажіть типи та базові типи виразів, якщо вони існують  
`& c, * (& c), & pi [0], & (p +10),`  
`& & ppf, ppf [10], (* ppf) [3]`

2. Є покажчик :

```
int n = 5, m = 20;
```

```
int * const pi = &n;
```

Які оператори синтаксично невірні?

```
* Pi = 10;
```

```
pi = &m;
```

```
* Pi + +;
```

```
(* Pi) + +;
```