

## ЛАБОРАТОРНА РОБОТА № 11

**Тема:** Вказівники та динамічні масиви.

**Мета:** Навчитись динамічно виділяти пам'ять під масиви та використовувати вказівники.

### Порядок виконання роботи

#### та методичні рекомендації до її виконання:

- створити новий консольний проект (Win32 Console Application) для виконання лабораторної роботи та зберегти його на власному мережевому диску;
- написати програмний код для виконання поставленого завдання згідно індивідуального варіанту;
- провести тестування програми з різним набором вхідних даних;
- побудувати блок-схему до написаної програми;
- оформити звіт до лабораторної роботи.

### Теоретичні відомості

#### Вказівники та одновимірні масиви

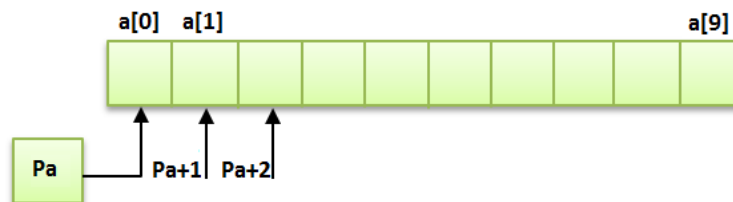
Вказівник - це змінна, яка містить адресу іншої змінної. Так як вказівник - це адреса деякого об'єкта, то через нього можна звертатися до даного об'єкта.

В C++ існує тісний зв'язок між вказівниками і масивами. Будь-який доступ до елементу масиву, здійснюваний операцією індексування, може бути виконаний за допомогою вказівника.

Декларація

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

визначає масив *a* розміру 10, тобто блок з десяти послідовних об'єктів, представлених на малюнку, з іменами *a[0]*, *a[1]*, ..., *a[9]*.



Запис *a[i]* відсилає нас до *i*-го елементу масиву. Якщо *pa* є вказівником, тобто визначений як *int \*pa;*, то в результаті присвоєння *pa = &a[0];* *pa* буде вказувати на нульовий елемент масиву *a*; інакше кажучи, *pa* буде містити адресу елемента *a[0]*. Тепер присвоювання *x = \*pa;* буде копіювати вміст *a[0]* в *x*.

Якщо  $pa$  вказує на певний елемент масиву, то  $pa+1$  по визначенню вказує на наступний елемент. Таким чином, якщо  $pa$  вказує на  $a[0]$ , то  $*(pa+1)$  являється вмістом  $a[1]$ ,  $pa+1$  - адресою  $a[1]$ ,  $*(pa+i)$  - вміст  $a[i]$ .

Оскільки ім'я масиву є не щось інше, як адреса його початкового елемента, присвоювання

```
 $pa = \& a[0];$ 
```

можна також записати в наступному вигляді:

```
 $pa = a;$ 
```

Так як  $pa$  - вказівник, то у виразах його можна використовувати з індексом, тобто запис  $pa[i]$  еквівалентна запису  $*(pa+i)$ . Елемент масиву однаково дозволяється зображати і у вигляді вказівника зі зміщенням, і у вигляді імені масиву з індексом.

Між ім'ям масиву і вказівником, що виступає в ролі імені масиву, існує одна відмінність. Вказівник - це змінна, тому можна написати  $pa = a$  або  $pa++$ . Але ім'я масиву не є змінною, і запис типу  $a = pa$  не допускається. Слід також розрізняти вирази  $*(a+2)$  і  $*a+2$ : перший повертає значення третього елемента масиву  $a$ ; другий - це додавання числа 2 до значення першого елемента масиву.

## Динамічні масиви

Динамічне виділення пам'яті необхідно для ефективного використання пам'яті комп'ютера. Наприклад, ми написали якусь програмку, яка обробляє масив. При написанні даної програми необхідно було оголосити масив, тобто поставити йому фіксований розмір (наприклад, від 0 до 100 елементів). Тоді дана програма буде не універсальною, адже може обробляти масив розміром не більше 100 елементів. А якщо нам знадобляться всього 20 елементів, але в пам'яті виділиться місце під 100 елементів, адже оголошення масиву було статичним, а таке використання пам'яті вкрай неефективне.

В C++ операції `new` і `delete` призначені для динамічного розподілу пам'яті комп'ютера. Операція `new` виділяє пам'ять з області вільної пам'яті, а операція `delete` вивільняє виділену пам'ять. Виділювана пам'ять, після її використання повинна вивільнятися, тому операції `new` і `delete` використовуються парами:

```
int *ptr = new int;
```

```
...
```

```
delete ptr;
```

Операція `new` створює об'єкт заданого типу, виділяє йому пам'ять і повертає вказівник відповідного типу на дану ділянку пам'яті. Якщо пам'ять неможливо виділити, наприклад, у разі відсутності вільних ділянок, то повертається нульовий

показчик, тобто вказівник поверне значення 0. Виділення пам'яті можливо під будь-який тип даних: `int`, `float`, `double`, `char` і т. д.

Як було сказано раніше, масиви також можуть бути динамічними. Найчастіше операції `new` і `delete` застосовуються для створення динамічних масивів, а не для створення динамічних змінних. Розглянемо фрагмент коду створення одновимірного динамічного масиву.

```
float *p_array = new float [10];
```

Після того як динамічний масив став непотрібним, потрібно звільнити ділянку пам'яті, який під нього виділявся.

```
delete [] p_array;
```

Після оператора `delete` ставляться квадратні дужки, які говорять про те, що вивільняється ділянку пам'яті, яка відводилась під одновимірний масив.

Тепер розглянемо фрагмент коду, в якому показано, як оголошується двовимірний динамічний масив.

```
float **p_array = new float* [2];  
for (int i = 0; i < 2; count++)  
    p_array[i] = new float [5];
```

Спочатку оголошується вказівник другого порядку `float **p_array`, який посилається на масив вказівників `float* [2]`, де розмір масиву дорівнює двом. Після чого в циклі `for` кожному рядку масиву оголошеного в рядку 2 виділяється пам'ять під п'ять елементів. У результаті виходить двовимірний динамічний масив `p_array[2][5]`. Розглянемо приклад вивільнення пам'яті виділеної під двовимірний динамічний масив.

```
for (int i = 0; i < 2; count++)  
    delete [] p_array[i];
```

Оголошення і видалення двовимірного динамічного масиву виконується за допомогою циклу, так як показано вище.

## Зразок виконання з одновимірним масивом

В одновимірному масиві замінити усі від'ємні та нульові елементи на одиниці. Елементи масиву згенерувати випадковим чином у вказаному діапазоні. Розмір масиву задається з консольного вікна користувачем. Вивести на екран початковий масив та видозмінений масив.

## Програмний код:

```
#include <iostream>
#include <time.h>
using namespace std;
void main()
{
    setlocale(LC_ALL, "Ukrainian");
    cout << "+-----+" << endl;
    cout << "                Лабораторна робота №11" << endl;
    cout << "                Вказівники та динамічні масиви" << endl;
    cout << "+-----+" << endl;

    int *Array; //оголошення вказівника
    int size;   //змінна для збереження розміру майбутнього масиву

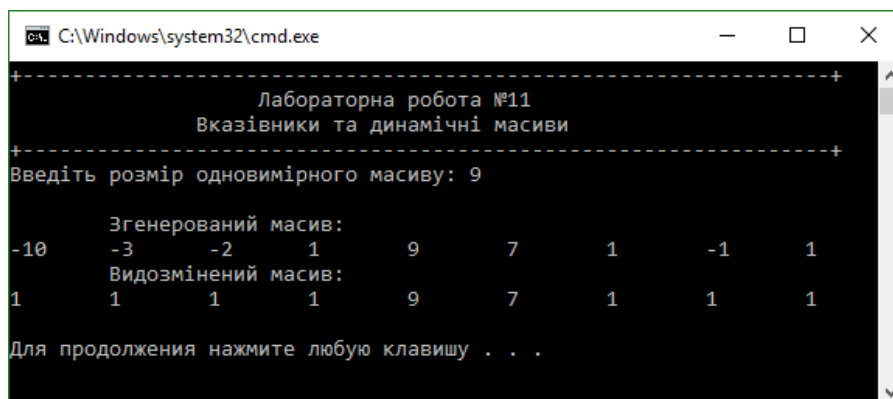
    cout << "Введіть розмір одновимірного масиву: ";
    cin >> size;
    /*виділення пам'яті під динамічний масив
    та присвоєння вказівнику адреси першого елемента масиву*/
    Array = new int[size];

    //генерування елементів матриці випадковим чином від -10 до 10
    float min = -10, max = 10;
    srand(time(0));
    for (int i = 0; i < size; i++)
        Array[i] = min + (rand() % (int)(max - min + 1));

    cout << "\n\tЗгенерований масив: " << endl;
    for (int i = 0; i < size; i++)
    {
        /*Класичний спосіб виводу елементів масиву також вірний
        cout << Array[i] << '\t';*/

        //Вивід елементів масиву через розіменований вказівник
        cout << *(Array + i) << '\t';
    }
    cout << "\n\tВидозмінений масив: " << endl;
    for (int i = 0; i < size; i++)
    {
        if (Array[i] <= 0)
            Array[i] = 1;
        cout << Array[i] << '\t';
    }
    cout << endl;
    system("pause");
}
```

## Результати виконання



```
C:\Windows\system32\cmd.exe
+-----+
                Лабораторна робота №11
                Вказівники та динамічні масиви
+-----+
Введіть розмір одновимірного масиву: 9

        Згенерований масив:
-10      -3      -2      1      9      7      1      -1      1
        Видозмінений масив:
1        1        1        1      9      7      1      1      1

Для продовження натисніть будь-яку клавішу . . .
```

## Зразок виконання з двовимірним масивом

У двовимірному масиві замінити усі елементи у стовпці з максимальним елементом на цей максимальний елемент. Кількість рядків та стовпців масиву задаються користувачем.

### Програмний код:

```
#include <iostream>
#include <time.h>
using namespace std;
void main()
{
    float **matrix;           //оголошення вказівника другого порядку
    int rowCount, colCount;    //змінні для збереження розміру майбутнього масиву
    cout << "Введіть розмір двовимірного масиву"<<endl;
    cout << "Кількість рядків: ";
    cin >> rowCount;
    cout << "Кількість стовпців: ";
    cin >> colCount;

    //виділення пам'яті під задану к-сть рядків
    //кожен рядок, в свою чергу, також буде масивом
    matrix = new float*[rowCount];

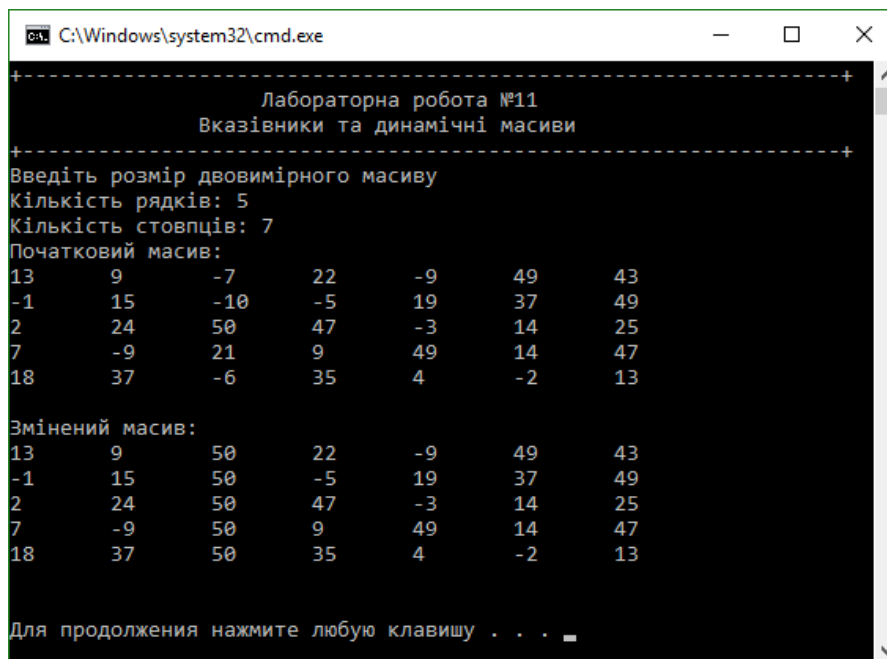
    float min = -10, max = 50;
    srand(time(0));
    cout << "Початковий масив: " << endl;
    for (int i = 0; i < rowCount; i++)
    {
        //виділення пам'яті під елементи кожного рядка
        matrix[i] = new float[colCount];
        for (int j = 0; j < colCount; j++)
        {
            matrix[i][j] = min + (rand() % (int)(max - min + 1));
            cout << matrix[i][j] << "\t";
        }
        cout << "\n";
    }

    //пошук максимального елементу та номера стовпця цього елементу
    int searchColIndex = 0;
    float maxElement = matrix[0][0];
    for (int i = 0; i < colCount; i++)
        for (int j = 0; j < rowCount; j++)
            if (matrix[j][i] > maxElement)
            {
                maxElement = matrix[j][i];
                searchColIndex = i;
            }

    //Заміна усіх елементів у стовпці, що містить максимальний елемент,
    //на цей максимальний елемент
    for (int i = 0; i < rowCount; i++)
        matrix[i][searchColIndex] = maxElement;
    cout << endl;

    cout << "Змінений масив: " << endl;
    for (int i = 0; i < rowCount; i++)
    {
        for (int j = 0; j < colCount; j++)
            cout << matrix[i][j] << "\t";
        cout << "\n";
    }
    cout << endl<<endl;
    system("pause");
}
```

## Результати виконання



```
C:\Windows\system32\cmd.exe
+-----+
+-----+
Лабораторна робота №11
Вказівники та динамічні масиви
+-----+
Введіть розмір двовимірного масиву
Кількість рядків: 5
Кількість стовпців: 7
Початковий масив:
13      9      -7      22      -9      49      43
-1      15     -10      -5      19      37      49
2       24      50      47      -3      14      25
7       -9      21      9       49      14      47
18      37      -6      35      4       -2      13

Змінений масив:
13      9      50      22      -9      49      43
-1      15      50      -5      19      37      49
2       24      50      47      -3      14      25
7       -9      50      9       49      14      47
18      37      50      35      4       -2      13

Для продовження натисніть будь-яку клавішу . . .
```

## Індивідуальні завдання

Розробити програми для роботи з одновимірними та двовимірними масивами. Розміри масивів задаються користувачем під час виконання програми.

### Варіант №1

1. Створити динамічний одновимірний масив. Знайти максимальний елемент і його номер.
2. Створити динамічний двовимірний масив В. Знайти суму діагональних елементів.

### Варіант №2

1. Створити динамічний одновимірний масив А. Знайти суму додатніх елементів.
2. Створити динамічний двовимірний масив С. Знайти суму елементів над головною діагоналлю.

### Варіант №3

1. Створити динамічний одновимірний масив С. Знайти добуток елементів, які відмінні від нуля.
2. Створити динамічний двовимірний масив А. Знайти середнє значення елементів масиву та 5-го стовпця

### Варіант №4

1. Створити динамічний одновимірний масив М. Знайти кількість від'ємних елементів.
2. Створити динамічний двовимірний масив Т. Знайти суму елементів парних стовпців.

#### **Варіант №5**

1. Створити динамічний одновимірний масив Т. Знайти різницю між максимальним та мінімальним елементом.
2. Створити динамічний двовимірний масив К. Знайти добуток кожного рядка масиву.

#### **Варіант №6**

1. Створити динамічний одновимірний масив Н. Знайти середнє значення парних елементів.
2. Створити динамічний двовимірний масив М. Вивести кількість елементів, які попадають в проміжок [2,19].

#### **Варіант №7**

1. Створити динамічний одновимірний масив Р. Знайти кількість від'ємних елементів.
2. Створити динамічний двовимірний масив У. Знайти максимальний елемент масиву.

#### **Варіант №8**

1. Створити динамічний одновимірний масив. Знайти суму непарних елементів.
2. Створити динамічний двовимірний масив В. Знайти мінімальний елемент та його індекси.

#### **Варіант №9**

1. Створити динамічний одновимірний масив О. Знайти мінімальний елемент і присвоїти його значення першому елементу.
2. Створити динамічний двовимірний масив В. Знайти добуток додатніх елементів масиву.

#### **Варіант №10**

1. Створити динамічний одновимірний масив В. Заміни всі його від'ємні елементи на нуль.

2. Створити динамічний двовимірний масив. Знайти максимальний елемент кожного стовпця.

### Варіант №11

1. Створити динамічний одновимірний масив А. Знайти суму елементів з парними індексами.
2. Створити динамічний двовимірний масив Т. Утворити новий масив Т1, кожний елемент якого рівний  $T1 = T_{ij}^2 + 3$

### Варіант №12

1. Створити динамічний одновимірний масив К. Знайти кількість елементів, що знаходяться на проміжку від 1,5 до 10.
2. Створити динамічний двовимірний масив Е. Знайти суму елементів парних стовпців.

### Варіант №13

1. Створити динамічний одновимірний масив Е. Збільшити елементи масиву у два рази.
2. Створити динамічний двовимірний масив Р. Знайти мінімальний елемент кожного рядка.

### Варіант №14

1. Створити динамічний одновимірний масив Н. Додати до всіх елементів масиву, що мають індекс 6 і більше число 100.
2. Створити динамічний двовимірний масив Н. Знайти середнє значення кожного стовпця.

### Варіант №15

1. Створити динамічний одновимірний масив К. Збільшити значення від'ємних елементів на 10.
2. Створити динамічний двовимірний масив С. Знайти кількість парних і кількість не парних елементів у масиві.