

## Робота №3

### Розробка та реалізація програм з використанням багатовимірних динамічних масивів

**Мета роботи:** оволодіння навичками складання програм з використанням багатовимірних динамічних масивів.

#### Завдання

**Завдання 3.1.** Нехай є наступний фрагмент програми, поданий у варіантах (див. табл. 3.1). Поясніть, яку задачу реалізує вказаний фрагмент, при наступному початку програми (Вказівка: замість N підставити номер варіанта за списком групи).

```
#include <iostream>
#include <stdlib.h>
#include <iomanip>
const int N=32;
using namespace std;
int** form_matr(int n){
    int **matr=new int*[n];
    for(int i=0;i<n;i++) matr[i]=new int [n];
    return matr;}
void zapolnen (int ** matr,int n){
    int p=N;
    for(int i=0;i<n;i++)
        for(int j=0;j<n;j++){          matr[i][j]=p++ ;
    }}
void printArray (int ** matr,int n){
    for(int i=0;i<n;i++){
        for(int j=0;j<n;j++)
            cout<<setw(5)<<matr[i][j];
        cout<<"\n";    }}
int main(){
    int n;
    n=10+N%10;
    int **matr=form_matr(n);
    zapolnen(matr,n);
```

```
printArray(matr, n);
```

Таблиця 3.1 – Варіанти завдання 3.1

№	Фрагмент програми
1–5	<pre>int x=0; for(int i=0;i&lt;n;i++) for(int j=i;j&lt;=n-i-1; j++){     x+=matr[i][j] ; }</pre>
6–10	<pre>int t; for(int i=0;i&lt;n;i++) for(int j=i;j&lt;=n-i-1; j++) { t=matr[i][j] ; matr[i][j]=matr[n-i-1][j]; matr[n-i-1][j]=t;} </pre>
11–15	<pre>int x=matr[1][n-1], m=0; for(int i=0;i&lt;n;i++){ for(int j=n-1-i;j&lt;n;j++) if (matr[i][j]&lt;x){     x=matr[i][j]; m=i;}}</pre>
16–20	<pre>int x=matr[1][n-1], m=0; for(int i=0;i&lt;n;i++){ for(int j=0;j&lt;=n-1-i; j++) if (matr[i][j]&gt;x){     x=matr[i][j]; m=i;}}</pre>
21–25	<pre>int t; for(int j=0;j&lt;n/2;j++) for(int i=j;i&lt;=n-j-1; i++){     t=matr[i][n-1-j] ; matr[i][n-1-j]= matr[i][j]; matr[i][j]=t;} </pre>
26–30	<pre>double x=1; for(int i=0;i&lt;n;i++){     for(int j=0;j&lt;=i;j++)         x*=matr[i][j]; }</pre>

**Завдання 3.2.** Розробити програму, яка виконує обчислення над двовимірними динамічними масивами, при написанні програми користуватися принципами структурного програмування (описати такі функції: що створює масив, що заповнює його випадковими числами, що друкує, що реалізує завдання з таблиці 3.2).

Таблиця 3.2 – Варіанти завдання 3.2

№	Завдання
1	Дана дійсна матриця розміром 7 * 8. Знайти найбільший елемент матриці. Поміняти рядок, що містить найбільший елемент з першим рядком матриці.

2	Дана дійсна матриця розміром $7 * 8$ . Знайти найбільший елемент матриці. Поміняти стовпчик, що містить найбільший елемент з першим стовпцем матриці.
3	Скласти програму заміни всіх від'ємних елементів матриці <b>A</b> (9,9) на 0, якщо сума мінімального і максимального елементів цієї матриці виявиться менше $P$ , де $P$ вводиться з клавіатури.
4	Скласти програму знаходження максимального елемента в кожному стовпчику матриці <b>A</b> (15,15).
5	Скласти програму знаходження мінімального додатнього елемента в кожному стовпці матриці <b>A</b> (11,11).
6	Скласти програму знаходження кількості рядків матриці <b>A</b> (10,10), сума елементів яких від'ємна.
7	Скласти програму знаходження кількості рядків матриці <b>A</b> (8, 8), кількість від'ємних елементів в яких більше за $P$ , де $P$ вводиться з клавіатури.
8	Скласти програму формування вектора <b>B</b> (16), якщо $b_i$ – сума мінімального і максимального елементів $i$ -го рядку матриці <b>A</b> (16,5).
9	Скласти програму заміни всіх від'ємних елементів матриці <b>A</b> (12,12) на елемент цієї матриці, що має максимальне значення.
10	Є двовимірний масив цілих чисел розмірності $6 * 15$ . Знайти номер рядка, для якого середньоарифметичне значення елементів максимально.
11	В двовимірному масиві цілих чисел розмірності $9 * 9$ поміняти місцями рядок і стовпчик, номер якого вводиться користувачем.
12	Дано масив <b>C</b> (14,14). Визначити кількість "особливих" елементів масиву, вважаючи елемент "особливим", якщо він більше суми інших елементів свого стовпчика. Надрукувати індекси "особливих" елементів.
13	Дано масив <b>C</b> (11,11). Визначити кількість "особливих" елементів масиву, вважаючи елемент "особливим" якщо в його рядку зліва від нього знаходяться елементи менші за нього, а праворуч більші.
14	Знайти мінімальний елемент серед максимальних елементів рядків двовимірного масиву цілих чисел. Визначити номер рядка і стовпчика такого елемента.
15	Видалити стовпець двовимірного масиву цілих чисел, в якому знаходиться максимальний елемент цього масиву.
16	Дано масив <b>C</b> (10,10). Знайти всі неповторювані елементи двовимірного масиву цілих чисел.
17	Дано масив <b>C</b> (7,10). Видалити стовпчики, розташовані між стовпчиками з мінімальним та максимальним елементами.
18	Дано масив <b>C</b> (9,11). Парні рядки масиву зсунути циклічно на $K$ елементів вправо, де $K$ вводиться з клавіатури.
19	Дано масив <b>C</b> (9,12). Непарні рядки масиву зсунути циклічно на $K$ елементів вліво, де $K$ вводиться з клавіатури.

20	Створіть двовимірний масив цілих чисел $C$ (11,9). Видаліть з нього рядок $i$ і стовпець, на перетині яких розташований мінімальний елемент.
----	--

**Завдання 3.3.** Скласти програму, яка модифікує двовимірні динамічні масиви. При написанні програми користуватися принципами структурного програмування (описати такі функції: створити масив, сформувати масив, друкувати на екрані, модифікувати масив, згідно завдання з таблиці 3.3, звільнити пам'ять ).

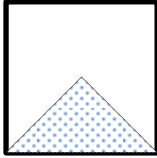
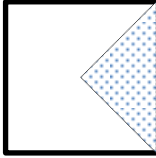



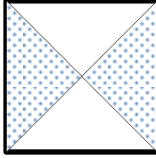
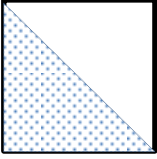
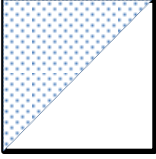
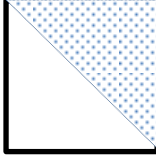
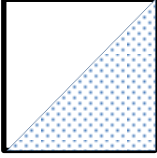
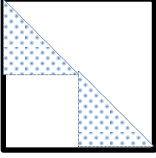
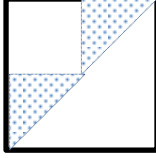
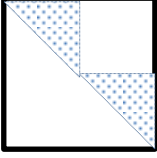
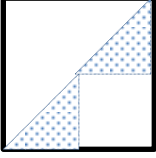
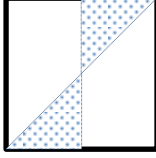
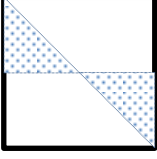
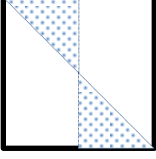
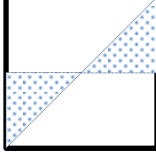
Таблиця 3.3 – Варіанти завдання 3.3

№	Завдання
1	Додати рядок з вказаним номером $K$
2	Додати рядок з початку матриці
3	Додати стовпчик з початку матриці
4	Додати $K$ рядків з початку матриці
5	Додати $K$ стовпців з початку матриці
6	Видалити рядок з номером $K$
7	Видалити стовпець з номером $K$
8	Видалити рядки, починаючи з рядка $K_1$ і до рядка $K_2$
9	Видалити стовпці, починаючи з стовпця $K_1$ і до стовпця $K_2$
10	Видалити всі парні рядки
11	Видалити всі парні стовпці
12	Видалити всі рядки, в яких є хоча б один нульовий елемент
13	Видалити всі стовпці, в яких є хоча б один нульовий елемент
14	Видалити рядок, в якому знаходиться найбільший елемент матриці
15	Додати рядки після кожного парного рядку матриці
16	Додати стовпці після кожного парного стовпця матриці
17	Додати $K$ рядків, починаючи з рядка з номером $N$
18	Додати $K$ стовпців, починаючи зі стовпця з номером $N$
19	Додати рядок після рядка, що містить найбільший елемент
20	Додати стовпчик після стовпчика, що містить найбільший елемент

**Завдання 3.4.** Розробити програму, яка заповнює двовимірний масив наступним чином: елементи, що належать заштрихованій області, генеруються випадковим чином; всі інші дорівнюють  $N$ , де  $N$  – номер варіанту. Упорядкувати всі випадкові елементи масиву, які належать заштрихованій області. Парні варіанти упорядковуються за зростанням,

непарні – за спаданням. При написанні програми користуватися принципами структурного програмування.

Таблиця 3.4 – Варіанти завдання 3.4

№	Рисунок	№	Рисунок	№	Рисунок
1		2		3	
4		5		6	
7		8		9	
10		11		12	
13		14		15	
16		17		18	

### Короткі теоретичні відомості

Окремим випадком багатовимірного масиву є двовимірний масив, або матриця. Двовимірний масив являє собою сукупність рядків і стовпців, на перетину яких перебуває конкретне значення. Для оголошення двовимірного масиву необхідно вказати кількість рядків і стовпців. При цьому діють ті ж правила, що й при оголошенні одномірного масиву:

тип\_даних ім'я\_масиву [число\_рядків][число\_стовпців];

Незважаючи на те, що ми уявляємо двовимірний масив у вигляді матриці, у пам'яті будь-який двовимірний масив розташовується по рядкам: спочатку нульовий рядок, потім перший і так далі. Про це слід пам'ятати, тому що вихід за межі масиву може викликати некоректну роботу програми, при цьому компілятор не повідомлює про помилку.

Звернення до певного елемента масиву здійснюється за номером рядка та номером стовпця, наприклад: `array [2][1]`.

Багатомірний масив в C++ по своїй суті одномірний. Операції `new` та `delete` дозволяють створювати й видаляти динамічні масиви, підтримуючи при цьому ілюзію довільної розмірності. Діяльність по організації динамічного масиву вимагає додаткової уваги, однак характеристики масиву (операнди операції `new`) можуть не бути константними виразами. Це дозволяє створювати багатомірні динамічні масиви довільної конфігурації. Більш докладно робота з операціями `new` та `delete` була розглянута у роботі №2.

Організація двовимірного динамічного масиву виконується у два етапи: спочатку створюється одномірний масив вказівників, а потім кожному елементу цього масиву присвоюється адреса одномірного масиву:

```
int size_row = 5, size_col = 5;
int **pArr = new int*[ size_row];
for (int i = 0; i < size_row; i++)
pArr[i] = new int[size_col];
```

Знищення двовимірного масиву відбувається у зворотній послідовності:

```
for (int i = 0; i < size_row; i++)
    delete[]pArr[i];
delete[]pArr;
```

### **Приклад виконання роботи**

**Завдання 3.1.** Пояснити, що виконує наступний фрагмент програми, де N=32.

```
int x=0;
```

```
for(int i=0;i<n;i++){
    for(int j=0;j<=n-1-i;j++)
        x+=matr[i][j];
```

### Розв'язання

Цей фрагмент програми реалізує обчислення суми елементів, що розташовані на та над другою діагоналлю матриці. Після виконання цього фрагмента  $x=208$ .

**Завдання 3.2.** Скласти програму. Видалити з двовимірного динамічного масиву рядок, що містить максимальний елемент.

### Розв'язання

#### 1. Постановка задачі

Скласти програму, що видаляє з двовимірного динамічного масиву рядок, що містить максимальний елемент, на мові C++.

#### 2. Алгоритм розв'язання задачі

Алгоритм розв'язання задачі можна представити у вигляді такої послідовності дій:

Дія 1. Виділити динамічну пам'ять для елементів масиву.

Дія 2. Заповнити масив випадковими числами.

Дія 3. Вивести елементи масиву на екран.

Дія 4. Визначити індекс рядка, в якому розташований максимальний елемент.

Дія 5. Виділити динамічну пам'ять для нового масиву.

Дія 6. У новий масив записати тільки ті рядки, що відрізняються від знайденого індексу.

Дія 7. Звільнити пам'ять, що була виділена під старий масив.

Дія 8. Замінити значення вказівки

#### 3. Текст програми

```
#include <iostream>
#include <stdlib.h>
#include <iomanip>
```

```

using namespace std;
/*виділення динамічної пам'яті під двовимірний динамічний
масив*/
int** form_matr(int n,int m){
int **matr=new int*[n];
for(int i=0;i<n;i++)
    matr[i]=new int [m];
    return matr;//повертаємо вказівку на масив вказівок
}
void zapolnen (int ** matr,int n,int m){
//заповнення матриці
for(int i=0;i<n;i++)
for(int j=0;j<m;j++)
    matr[i][j]=rand()%100;
}
void printArray (int ** matr,int n,int m){
for(int i=0;i<n;i++){
for(int j=0;j<m;j++)
    cout<<setw(5)<<matr[i][j];
    cout<<"\n";}
}
void dell(int ** matr,int n,int m){//звільнення пам'яті
for(int i=0;i<n;i++)
    delete [] matr[i];
delete[]matr;
}
//пошук індексу рядка з максимальним елементом
int maxIndex (int ** matr,int n,int m){
int max=matr[0][0],imax=0;
for(int i=0;i<n;i++)
for(int j=0;j<m;j++)
    if (max<matr[i][j]) {    max=matr[i][j];
                            imax=i;}
return imax;
}
int ** deleteMax (int ** matr,int &n,int m){

```



```

int**temp=form_matr(n-1,m);
    //заповнення нової матриці
int t=0,k=maxIndex(matr,n,m);
for(int i=0;i<n;i++)
    if(i!=k)
    {
        for(int j=0;j<m;j++)
            temp[t][j]=matr[i][j];
        t++;
    }
dell(matr, n, m);
    n--;
    return temp;
}
int main()
{
    int n,m;//розмір матриці
    cout<<"\nEnter n=";
    cin>>n;//рядки
    cout<<"\nEnter m=";
    cin>>m;//стовпці
    int **matr=form_matr(n,m);
    zapolnen(matr,n,m);
    cout<<"old array:\n";
    printArray(matr,n,m);
    matr=deleteMax(matr,n,m);
    cout<<"new array:\n";
    printArray(matr,n,m);
    dell(matr, n, m);
    return 0;
}

```

#### 4. Результати роботи програми

Enter n=5

Enter m=5

old array:

41	67	34	0	69
24	78	58	62	64
5	45	81	27	61
91	95	42	27	36
91	4	2	53	92

new array:

41	67	34	0	69
24	78	58	62	64
5	45	81	27	61
91	4	2	53	92

### **Контрольні питання**

1. Як оголосити двовимірний масив?
2. Які існують варіанти ініціалізації двовимірного масиву?
3. Продемонструйте взаємозв'язок між посиланнями та масивами.
4. Що таке динамічна пам'ять?
5. Яка існує особливість динамічного виділення пам'яті у випадку двовимірного масиву?
6. Яка існує особливість звільнення пам'яті у випадку двовимірного масиву?