

# ЛАБОРАТОРНА РОБОТА № 17

## Робота з двовимірними масивами

**Мета:** набуття практичних навичок роботи з двовимірними масивами.

### Хід роботи:

Завдання 1. Дана квадратна матриця.

13.	Розмістити елементи вказаних рядків у порядку зростання.
-----	--

Рис. 1. Завдання для написання першої програми

Лістинг програми:

```
#include <stdio.h>
#include <math.h>
#include <windows.h>
#include <stdlib.h>
#include <time.h>
#define N 50

int main()
{
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    srand(time(NULL));
    HANDLE h = GetStdHandle(STD_OUTPUT_HANDLE);
    int matr[N][N], n, r, fl, tmp, x = 0;
    printf("Введіть кількість елементів в рядку квадратної матриці: ");
    scanf_s("%d", &n);
    printf("Матриця %d x %d:\n\n", n, n);
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            matr[i][j] = rand() % 21 - 10;
            printf("%6d", matr[i][j]);
        }
        printf("\n");
    }
    printf("\n");
    do {
        printf("Введіть номер рядка матриці для сортування: ");
        scanf_s("%d", &r);
        printf("\n");
        if (r > 0 && r <= n) {
            do
            {
                fl = 0;
                for (int j = 1; j < n; j++)
                    if (matr[r - 1][j - 1] > matr[r - 1][j]) {
                        tmp = matr[r - 1][j]; matr[r - 1][j] = matr[r - 1][j - 1];
                        matr[r - 1][j - 1] = tmp;
                    }
            } while (fl == 0);
        }
    } while (r > 0 && r <= n);
}
```

					ДУ «Житомирська політехніка».22.122.13.000 – Лр17						
Змн.	Арк.	№ докум.	Підпис	Дата							
Розроб.		Черниш М.			Звіт з лабораторної роботи			Лім.	Арк.	Аркушів	
Перевір.		Терещук С.О.							1	7	
Керівник								ФІКТ Гр. КН-22-3[2]			
Н. контр.											
Зав. каф.											

```

        fl = 1;
    }
} while (fl);
for (int i = 0; i < n; i++) {
    for (int j = 0; j < n; j++) {
        if (i == r - 1) SetConsoleTextAttribute(h, 3);
        else SetConsoleTextAttribute(h, 7);
        printf("%6d", matr[i][j]);
    }
    printf("\n");
}
SetConsoleTextAttribute(h, 7);
printf("\nНатисніть 1, щоб продовжити роботу та 0, щоб завершити роботу.\n");
scanf_s("%d", &x);
}
else printf("\nПомилка: введене число не відповідає номеру рядка матриці!\n");
} while (x == 1);
return 0;
}

```

Результат виконання програми:

```

Microsoft Visual Studio Debug Console
Введіть кількість елементів в рядку квадратної матриці: 5
Матриця 5 x 5:
    3    3    1    4   -5
   -9    8    4   -4  -10
    7   -2    7   -5   -2
    0   -4   -1    0  -10
   -9   -6   -4   -8   -3

Введіть номер рядка матриці для сортування: 1
   -5    1    3    3    4
   -9    8    4   -4  -10
    7   -2    7   -5   -2
    0   -4   -1    0  -10
   -9   -6   -4   -8   -3

Натисніть 1, щоб продовжити роботу та 0, щоб завершити роботу.
1
Введіть номер рядка матриці для сортування: 3
   -5    1    3    3    4
   -9    8    4   -4  -10
   -5   -2   -2    7    7
    0   -4   -1    0  -10
   -9   -6   -4   -8   -3

Натисніть 1, щоб продовжити роботу та 0, щоб завершити роботу.
0

```

Рис. 2. Результат виконання першої програми

## Завдання 2. Дана прямокутна матриця.

13.

Ущільнити задану матрицю, вилучаючи із неї рядки і стовпці, заповнені нулями.

Рис. 3. Завдання для написання другої програми

Лістинг програми:

```
#include <stdio.h>
#include <math.h>
#include <windows.h>

int main()
{
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    int arr[5][5] = { 0, 12, 3, -14, -6, 0, 0, 0, 0, 0, 0, 4, 10, 6, -8, 0, 17, 8, -9, 0,
0, 5, 7, 5, 22 };
    int a, l = 0, k = 0, n = 5, m = 5;
    printf("Прямокутна матриця розміру 5 x 5: \n\n");
    for (int i = 0; i < 5; i++) {
        for (int j = 0; j < 5; j++) {
            printf("%6d", arr[i][j]);
        }
        printf("\n");
    }
    for (int i = 0; i < 5; i++) {
        a = 0;
        for (int j = 0; j < 5; j++) {
            if (arr[i][j] == 0) a++;
            if (a == 4) {
                l = i; n--;
            }
        }
    }
    for (int j = 0; j < 5; j++) {
        a = 0;
        for (int i = 0; i < 5; i++) {
            if (arr[i][j] == 0) a++;
            if (a == 4) {
                k = j; m--;
            }
        }
    }
    printf("\n\nУщільнена матриця розміру %d x %d: \n\n", n, m);
    for (int i = 0; i < 5; i++) {
        if (i == l) continue;
        for (int j = 0; j < 5; j++) {
            if (j == k) continue;
            printf("%6d", arr[i][j]);
        }
        printf("\n");
    }
    return 0;
}
```

Результат виконання програми:

```

Microsoft Visual Studio Debug Console
Прямокутна матриця розміру 5 x 5:

0    12    3   -14   -6
0     0     0     0     0
0     4    10     6   -8
0    17     8    -9     0
0     5     7     5    22

Ущільнена матриця розміру 4 x 4:

12     3   -14   -6
 4    10     6   -8
17     8    -9     0
 5     7     5    22

D:\КН-22-3[2]\Основи програмування\Лабораторні

```

Рис. 4. Результат виконання другої програми

Завдання 3. Здійснити сортування двовимірного масиву з наступними параметрами:

13	обміном, методом вибору	довге ціле	З початку	За спаданням
----	----------------------------	------------	-----------	--------------

Рис. 5. Завдання для написання третьої програми

Лістинг програми:

```

#include <stdio.h>
#include <math.h>
#include <windows.h>
#include <stdlib.h>
#include <time.h>
#define N 100

int main()
{
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    srand(time(NULL));
    long mas[N][N], arr[N];
    int n, m, fl, tmp, imax, method, l = 0;
    printf("Введіть кількість рядків двовимірного масиву: ");
    scanf_s("%d", &n);
    printf("Введіть кількість стовпців двовимірного масиву: ");

```

```

scanf_s("%d", &m);
printf("\nДвовимірний масив [%d][%d]:\n\n", n, m);
for (int i = 0; i < n; i++) {
    for (int j = 0; j < m; j++) {
        mas[i][j] = rand() % 21 - 10;
        printf("%6d", mas[i][j]);
    }
    printf("\n");
}
for (int i = 0; i < n; i++)
    for (int j = 0; j < m; j++) {
        arr[l] = mas[i][j];
        l++;
    }
printf("\nОберіть метод сортування: 1 - обміном, 2 - методом вибору: ");
scanf_s("%d", &method);
switch (method) {
case 1: do {
        fl = 0;
        for (int i = 1; i < n * m; i++)
            if (arr[i - 1] < arr[i])
            {
                tmp = arr[i];
                arr[i] = arr[i - 1];
                arr[i - 1] = tmp;
                fl = 1;
            }
    } while (fl); break;
case 2: for (int i = 0; i < n * m - 1; i++)
    {
        imax = i;
        for (int j = i + 1; j < n * m; j++)
            if (arr[j] > arr[imax]) imax = j;
        tmp = arr[i];
        arr[i] = arr[imax];
        arr[imax] = tmp;
    } break;
default: printf("Помилка: введене число не є номером методу сортування!");
}
l = 0;
for (int i = 0; i < n; i++)
    for (int j = 0; j < m; j++) {
        mas[i][j] = arr[l];
        l++;
    }
printf("\nДвовимірний масив [%d][%d], відсортований за спаданням:\n\n", n, m);
for (int i = 0; i < n; i++) {
    for (int j = 0; j < m; j++) {
        printf("%6d", mas[i][j]);
    }
    printf("\n");
}
return 0;
}

```

		Черниш М.			ДУ «Житомирська політехніка».22.122.13.000 – Лр17	Арк.
		Герецьук С.О.				5
Змн.	Арк.	№ докум.	Підпис	Дата		

## Результат виконання програми:

```

Microsoft Visual Studio Debug Console
Введіть кількість рядків двовимірного масиву: 5
Введіть кількість стовпців двовимірного масиву: 6

Двовимірний масив [5][6]:

-8    7    2   -1    1   -6
 9   -3   -4   -4   10    9
-7   -7    5   -3   -1   10
-6   10   -2   -7   -4   -2
-4    9   -7    1    5    0

Оберіть метод сортування: 1 - обміном, 2 - методом вибору: 2

Двовимірний масив [5][6], відсортований за спаданням:

10    10    10    9    9    9
 7    5     5    2    1    1
 0   -1   -1   -2   -2   -3
-3   -4   -4   -4   -4   -6
-6   -7   -7   -7   -7   -8

D:\КН-22-3[2]\Основи програмування\Лабораторні роботи з ОП\ОР_Lab_17

```

Рис. 6. Результат виконання третьої програми

## Завдання 4. Встановіть відповідність:

1	Масиви можуть бути:	локальні	int A [10]; void main () { ... }
		глобальні	void main () { int A [5] [5]; ... }

Рис. 7. Результат виконання першого пункту четвертого завдання

2	Масиви бувають	1	двовимірні	int A [3] [4] [5]; 2
				int A [1]; 3
		2	багатовимірні	int A [5]; 3
				int A [1] [5]; 1
		3	одновимірний	int A [3] [4] [5] [6]; 2
				int A [3] [5]; 1

Рис. 8. Результат виконання другого пункту четвертого завдання

3	У разі ... масиви має наступний вигляд	повної ініціалізації 1	int A [4] = {1, 4}; 2
			char x[][3]={9,8,7},{6,5,4},{3,2,1}}; 1
			int mas [ ][5]={ { 1, 5, 3, 7, 4 },{10, 11, 13, 14, 25 } }; 1
			int A [] = {0}; 1
			int A [4] [4] = {{2, 6}, {1, 4}, {6} }; 2
			int A [2] [4] = {1, 4, 4, 7, 2, 1, 4, 4, 2}; 1
		часткової ініціалізації 2	inta[5][3] = { {4, 7, 8}, {9, 66, -1},{5, - 5, 0},{3, -3, 30},{1, 1, 1} }; 1
			int mas [2][5]={ 1, 5, 3, 7, 4,10, 11, 13, 14, 25 }; 1
			int mas [ ][5]={ 1, 5, 3, 7, 4, 10, 11, 13, 14, 25 }; 1
			int A [4] = {1, 4, 2, 6}; 1
			int A [] = {1, 4, 2, 6}; 1
			int Mass[3][2][4]={ { {1,2,3,4},{5,6,7,8}}, { {9,10,11,12},{13,14,15,16}}, { {17,18,19,20},{21,22,23,24}}}; 1
			int z[6][3][2][5][8][4] 2

Рис. 9. Результат виконання третього пункту четвертого завдання