

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”
ІНСТИТУТ КОМП’ЮТЕРНИХ ТЕХНОЛОГІЙ, АВТОМАТИКИ ТА
МЕТРОЛОГІЇ
КАФЕДРА КОМП’ЮТЕРИЗОВАНИХ СИСТЕМ АВТОМАТИКИ



ЗВІТ

про виконання лабораторної роботи № 3

з навчальної дисципліни:

«Алгоритмізація та програмування, частина 3»

Виконав:

студент групи ІР-24

Шийка Андрій

Прийняв:

Клиско Ю.В.

Львів - 2025

Мета роботи: вдосконалення навичок роботи з різними масивами (двовимірними, покажчиків, динамічними) та покажчиками на масиви.

Завдання 5.1.

Розробіть програму, в якій вводиться з клавіатури і виводиться на екран двовимірний масив цілих чисел заданого розміру та виконуються задані обчислення.

Обчисліть добуток елементів в діапазоні від 1 до 5 в стовпцях. Розмір масиву 4 на 5.

```
#include <stdio.h>

#include <stdlib.h>

int main()
{
    int arr[4][5];

    int product[5]; // Масив для зберігання добутків по стовпцях

    int i, j;

    // Введення масиву з клавіатури

    printf("Enter array elements 4x5:\n");

    for (i = 0; i < 4; i++)
    {
        printf("Row %d:\n", i + 1);

        for (j = 0; j < 5; j++)
        {
            printf("  arr[%d][%d] = ", i, j);

            scanf("%d", &arr[i][j]);

        }
    }

    // Виведення масиву на екран
```

```
printf("\nArray arr[4][5]:\n");

for (i = 0; i < 4; i++)
{
    for (j = 0; j < 5; j++)
        printf("%5d", arr[i][j]);

    printf("\n");
}

// Обчислення добутку елементів від 1 до 5 в кожному стовпці

printf("\nProduct of elements in range from 1 to 5 by columns:\n");

for (j = 0; j < 5; j++)
{
    product[j] = 1; // Ініціалізація добутку для кожного стовпця

    for (i = 0; i < 4; i++)
    {
        // Перевірка, чи елемент входить в діапазон [1, 5]

        if (arr[i][j] >= 1 && arr[i][j] <= 5)
        {
            product[j] *= arr[i][j];
        }
    }

    printf("Column %d: product = %d\n", j, product[j]);
}

return 0;
}
```

```

> ./lab_3/5_1
Enter array elements 4x5:
Row 1:
  arr[0][0] = 1
  arr[0][1] = 2
  arr[0][2] = 3
  arr[0][3] = 4
  arr[0][4] = 5
Row 2:
  arr[1][0] = 1
  arr[1][1] = 2
  arr[1][2] = 3
  arr[1][3] = 4
  arr[1][4] = 5
Row 3:
  arr[2][0] = 5
  arr[2][1] = 4
  arr[2][2] = 3
  arr[2][3] = 2
  arr[2][4] = 1
Row 4:
  arr[3][0] = 5
  arr[3][1] = 4
  arr[3][2] = 3
  arr[3][3] = 2
  arr[3][4] = 1

Array arr[4][5]:
  1   2   3   4   5
  1   2   3   4   5
  5   4   3   2   1
  5   4   3   2   1

Product of elements in range from 1 to 5 by columns:
Column 0: product = 25
Column 1: product = 64
Column 2: product = 81
Column 3: product = 64
Column 4: product = 25

```

Завдання 5.2.

Розробіть програму, в якій двовимірний масив заданого розміру заповнюється випадковими числами у заданому діапазоні і виводиться на екран. Потім виконуються задані обчислення, використовуючи різні способи доступу до елементів масиву.

Обчисліть кількість від'ємних чисел в масиві. Розмір масиву 5 на 4. Діапазон випадкових чисел від -50.50 до 20.50 .

```

#include <stdio.h>

#include <time.h>

#include <stdlib.h>

int main()
{
    const int N = 5;

    const int M = 4;

```

```
float A[N][M];

srand((unsigned)time(NULL)); // Запуск генератора випадкових чисел

int i, j;

int count = 0; // Лічильник від'ємних чисел

// Заповнення масиву випадковими числами від -50.50 до 20.50

printf("Array A[%d][%d]:\n", N, M);

for (i = 0; i < N; i++)
{
    for (j = 0; j < M; j++)
    {
        // Генерація випадкового числа в діапазоні [-50.50, 20.50]

        (*(A + i) + j) = (rand() % 7101) * 0.01f - 50.50f;

        printf("%8.2f", *(A[i] + j));

    }

    printf("\n");
}

// Спосіб 1: Звичайна індексація A[i][j]

count = 0;

for (i = 0; i < N; i++)
{
    for (j = 0; j < M; j++)
    {
        if (A[i][j] < 0)

            count++;

    }
}
```

```
printf("Method 1 (A[i][j]): number of negative numbers = %d\n", count);

// Спосіб 2: Використання вказівників *(A[i] + j)

count = 0;

for (i = 0; i < N; i++)
{
    for (j = 0; j < M; j++)
    {
        if (*(A[i] + j) < 0)
            count++;
    }
}

printf("Method 2 (*(A[i] + j)): number of negative numbers = %d\n", count);

// Додатково: Виведення списку від'ємних чисел

printf("\nNegative numbers in array:\n");

for (i = 0; i < N; i++)
{
    for (j = 0; j < M; j++)
    {
        if (A[i][j] < 0)
            printf("A[%d][%d] = %7.2f\n", i, j, A[i][j]);
    }
}

return 0;
}
```

```

> ./lab_3/5_2
Array A[5][4]:
  16.75   -0.77   19.95  -22.21
 -29.58  -45.95  -42.59  -10.65
  -6.12  -29.25  -42.54  -21.45
  16.44  -29.21  -25.00   9.79
  -7.27  -29.51  -14.78  -7.43
Method 1 (A[i][j]): number of negative numbers = 16
Method 2 (*(A[i] + j)): number of negative numbers = 16

Negative numbers in array:
A[0][1] = -0.77
A[0][3] = -22.21
A[1][0] = -29.58
A[1][1] = -45.95
A[1][2] = -42.59
A[1][3] = -10.65
A[2][0] = -6.12
A[2][1] = -29.25
A[2][2] = -42.54
A[2][3] = -21.45
A[3][1] = -29.21
A[3][2] = -25.00
A[4][0] = -7.27
A[4][1] = -29.51
A[4][2] = -14.78
A[4][3] = -7.43

```

Завдання 5.3.

Розробіть програму, в якій визначається масив покажчиків на три задані рядки. Ці рядки виводяться на екран, визначається їх розмір та розмір масиву покажчиків. Потім виконуються задані обчислення.

Визначте масив з 3-х покажчиків на рядки з назвами книги, глави та параграфа. У другому рядку підрахувати кількість букв 'с'.

```

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

int main()
{
    char string_1[] = "C Programming Language";
    char string_2[] = "Arrays and Pointers Section";
    char string_3[] = "Working with Strings and Characters";

    // Масив вказівників на 3 рядки (книга, розділ, параграф)
    char *book_parts[] = {&string_1, &string_2, &string_3};
}

```

```
// Виведення рядків на екран

printf("Book:      %s\n", book_parts[0]);
printf("Chapter:   %s\n", book_parts[1]);
printf("Paragraph: %s\n", book_parts[2]);


// Визначення розмірів

printf("Size of pointer array: %d bytes\n", sizeof(book_parts));
printf("Size of string 1: %d bytes\n", sizeof(book_parts[0]));
printf("Size of string 2: %d bytes\n", sizeof(book_parts[1]));
printf("Size of string 3: %d bytes\n", sizeof(book_parts[2]));


// Виведення рядків з їх довжиною

for (int i = 0; i < 3; i++)
{
    printf("String %d: %-40s Length: %d\n", i + 1, book_parts[i],
strlen(book_parts[i]));
}


// Підрахунок кількості літер 'c' у другому рядку (розділ)

int count_c = 0;

printf("Second string (chapter): \"%s\"\n\n", string_2);


// Підрахунок літери 'c' (і малої, і великої)

for (int i = 0; i < strlen(string_2); i++)
{
    if (string_2[i] == 'c' || string_2[i] == 'C')
    {
        count_c++;

        printf("Found 'c' at position %d\n", i);
    }
}
```



```

}

printf("\nTotal number of 'c' letters in the second string: %d\n",
count_c);

return 0;
}

```

```

● > ./lab_3/5_3
Book:      C Programming Language
Chapter:   Arrays and Pointers Section
Paragraph: Working with Strings and Characters
Size of pointer array: 24 bytes
Size of string 1: 8 bytes
Size of string 2: 8 bytes
Size of string 3: 8 bytes
String 1: C Programming Language           Length: 22
String 2: Arrays and Pointers Section       Length: 27
String 3: Working with Strings and Characters Length: 35
Second string (chapter): "Arrays and Pointers Section"

Found 'c' at position 22

Total number of 'c' letters in the second string: 1

```

Завдання 5.4.

Розробіть програму, в якій створюється динамічний масив для змінних заданого типу з використанням операції `new` або функції `malloc`. Розмір масиву ввести з клавіатури. Масив заповнити випадковими числами у заданому діапазоні та виводиться на екран. Потім звільняється пам'ять.

Створіть динамічний масив для змінних типу `double` з використанням операції `new` (`malloc` для C). Масив заповніть випадковими числами у діапазоні від -15.000 до 15.000 .

```

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

int main()
{
    int n, i;

    double *ptr; /* вказівник на масив чисел типу double */

```

```
/* Ініціалізація генератора випадкових чисел */

srand(time(NULL));

printf("Input size array: ");

scanf("%d", &n);

/* Перевірка коректності розміру */

if (n <= 0)

{

    printf("Error: size must be positive\n");

    return -1;

}

/* Запит пам'яті */

ptr = (double *)malloc(n * sizeof(double));

/* Перевірка виділення пам'яті */

if (ptr == NULL)

{

    printf("Error: memory allocation failed\n");

    return -1;

}

/* Заповнення масиву випадковими числами від -15000 до 15000 */

printf("\nGenerating random array...\n");

for (i = 0; i < n; i++)

{

    /* Генерація випадкового числа в діапазоні [-15000, 15000] */

    ptr[i] = ((double)rand() / RAND_MAX) * 30000.0 - 15000.0;

}
```

```

}

/* Виведення масиву */
printf("\nArray:\n");
for (i = 0; i < n; i++)
{
    printf("%.3f ", ptr[i]);
}
printf("\n");

/* Звільнення пам'яті */
free(ptr);

printf("\nMemory freed successfully.\n");

return 0;
}

```

```

● > ./lab_3/5_4
Input size array: 5

Generating random array...

Array:
-9017.692 -356.655 5706.655 1756.247 -2759.626

Memory freed successfully.
ap_p3_labs_iot_nulp on 🍃 lab/3 [!?] took 3s
○ > █

```

Завдання 5.5.

Доповніть програму, створивши третю динамічну матрицю. Виконайте задані відповідно варіанту дії. Результат занесіть в третю матрицю та виведіть на екран.

Першу матрицю заповніть випадковими числами від -100 до +100. В третю матрицю занесіть суму абсолютних значень елементів перших двох матриць.

```
#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#include <math.h>


int main()

{

    int i, j, N;


    printf("Input the size of the square matrix: ");

    scanf("%d", &N);


    /* Перевірка коректності розміру */

    if (N <= 0)

    {

        printf("Error: size must be positive\n");

        return -1;

    }


    /* Перша матриця - вказівник на масив вказівників */

    int **matr;

    matr = (int **)malloc(N * sizeof(int *));

    if (matr == NULL)

    {

        printf("Error: memory allocation failed for matr\n");

        return -1;

    }


    for (i = 0; i < N; i++)
```

```
{

    matr[i] = (int *)malloc(N * sizeof(int));

    if (matr[i] == NULL)

    {

        printf("Error: memory allocation failed for matr[%d]\n", i);

        return -2;

    }

}

/* Друга матриця */

int **mas;

mas = (int **)malloc(N * sizeof(int *));

if (mas == NULL)

{

    printf("Error: memory allocation failed for mas\n");

    return -1;

}

for (i = 0; i < N; i++)

{

    mas[i] = (int *)malloc(N * sizeof(int));

    if (mas[i] == NULL)

    {

        printf("Error: memory allocation failed for mas[%d]\n", i);

        return -2;

    }

}

/* Третя матриця (результуюча) */
```

```
int **result;

result = (int **)malloc(N * sizeof(int *));

if (result == NULL)

{

    printf("Error: memory allocation failed for result\n");

    return -1;

}


for (i = 0; i < N; i++)

{

    result[i] = (int *)malloc(N * sizeof(int));

    if (result[i] == NULL)

    {

        printf("Error: memory allocation failed for result[%d]\n", i);

        return -2;

    }

}


/* Ініціалізація генератора випадкових чисел */

srand((unsigned)time(NULL));


/* Заповнення першої матриці випадковими числами від -100 до +100 */

printf("\n\tMatrix matr (random from -100 to +100):\n");

for (i = 0; i < N; i++)

{

    for (j = 0; j < N; j++)

    {

        matr[i][j] = rand() % 201 - 100; /* діапазон [-100, 100] */

        printf("%8d", matr[i][j]);
```

```
}

printf("\n");

}

/* Заповнення другої матриці */

printf("\n\tMatrix mas:\n");

printf("Input the elements of the matrix mas:\n");

for (i = 0; i < N; i++)
{
    for (j = 0; j < N; j++)
    {
        scanf("%d", &mas[i][j]);
    }
}

/* Виведення другої матриці */

printf("\n\tMatrix mas:\n");

for (i = 0; i < N; i++)
{
    for (j = 0; j < N; j++)
    {
        printf("%8d", mas[i][j]);
    }

    printf("\n");
}

/* Обчислення третьої матриці - сума абсолютних значень */

printf("\n\tMatrix result (sum of absolute values):\n");

for (i = 0; i < N; i++)
```

```
{

    for (j = 0; j < N; j++)

    {

        result[i][j] = abs(matr[i][j]) + abs(mas[i][j]);

        printf("%8d", result[i][j]);

    }

    printf("\n");

}

/* Звільнення пам'яті */

for (i = 0; i < N; i++)

{

    free(matr[i]);

    free(mas[i]);

    free(result[i]);

}

free(matr);

free(mas);

free(result);

printf("\nMemory freed successfully.\n");

return 0;

}
```



```
• > ./lab_3/5_5
Input the size of the square matrix: 2

      Matrix matr (random from -100 to +100):
-38   -85
 5     50

      Matrix mas:
Input the elements of the matrix mas:
-100
100
20
30

      Matrix mas:
-100   100
 20    30

      Matrix result (sum of absolute values):
138   185
25    80

Memory freed successfully.
```