

Міністерство освіти і науки України
Національний університет “Львівська політехніка”
Інститут комп’ютерних наук та інформаційних технологій
Кафедра програмного забезпечення



Звіт

Про виконання лабораторної роботи №3
на тему:

«Робота з масивами в С»

з дисципліни «Основи програмування»

Лектор:

ст. викл. каф. ПЗ
Муха Т.О.

Виконав:

ст. гр. ПЗ-11
Ясногородський Н.В.

Прийняв:

асист. каф. ПЗ
Дивак І.В.

« __ » _____ 2021 р.

Σ = _____ .

Львів – 2021

Тема: Робота з масивами в С

Мета: Навчитися організовувати такі структури даних як масиви та освоїти основні методи програмування алгоритмів обробки масивів даних засобами мови С.

ЗАВДАННЯ

Завдання 1.

Написати програму для обробки даних, організованих у масив, згідно завдання наведеного варіанту. У програмі використати такі форми звертання до елементів масиву:

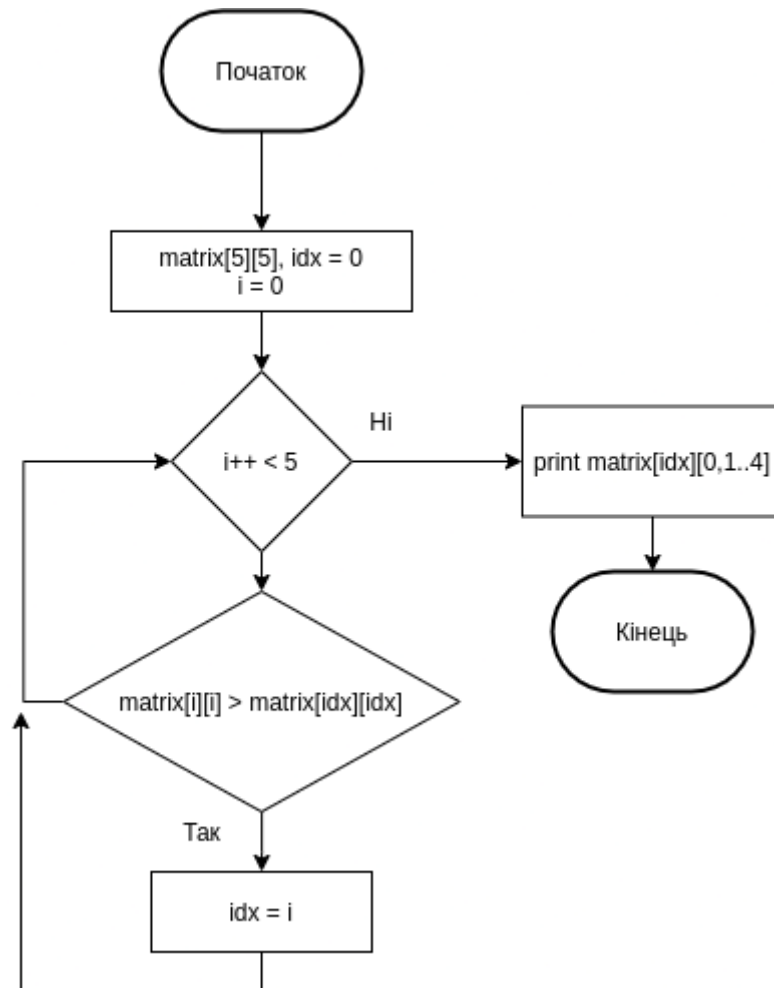
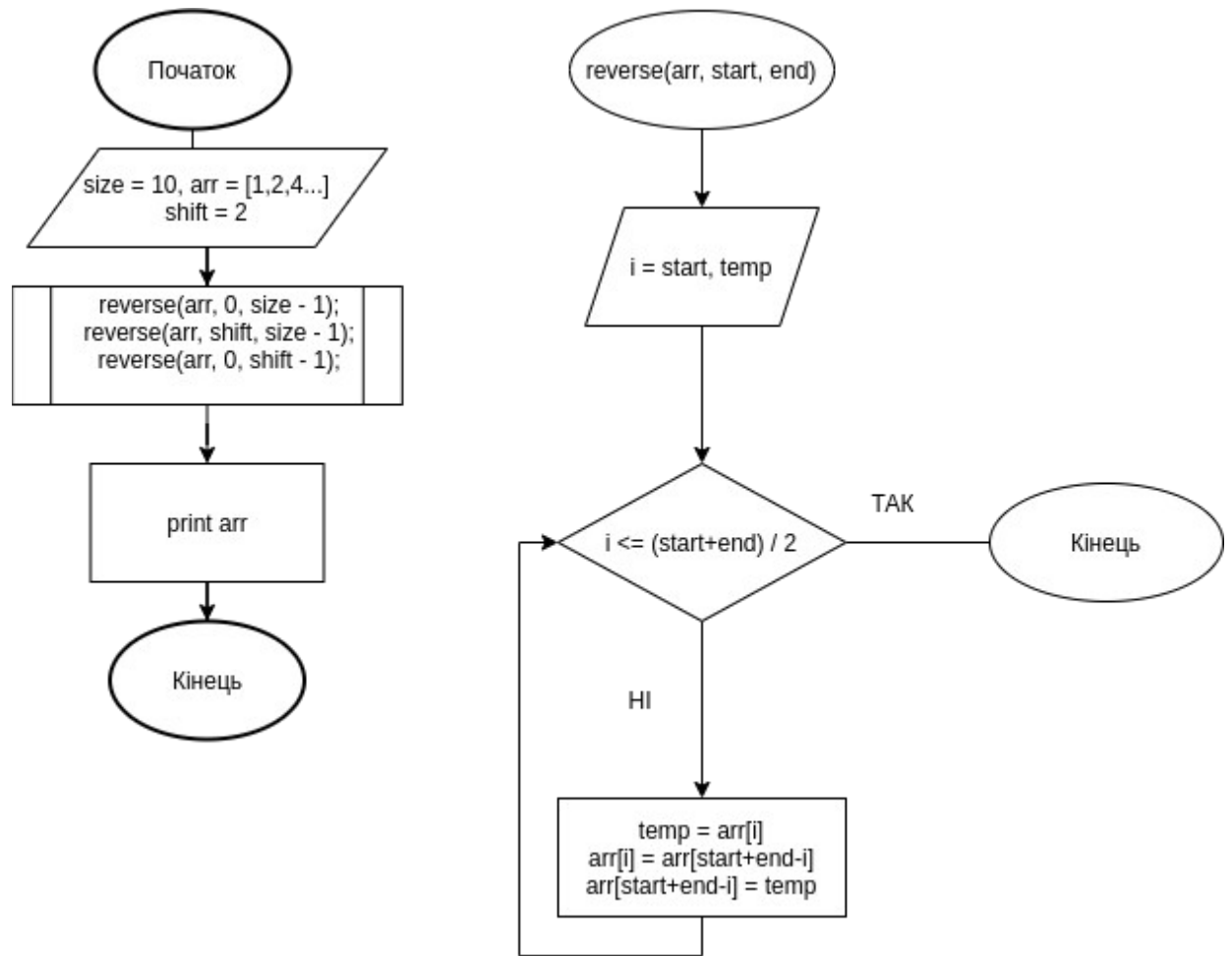
5. Задано масив цілих чисел. Циклічно зсунути його елементи на 2 позиції вправо. *Підказка:* циклічний зсув передбачає, що старші елементи масиву переходять на місце зсунутих молодших.

Завдання 2.

Написати програму для обробки даних, організованих у масив, згідно завдання наведеного варіанту. У програмах використати різні форми звертання до елементів багатовимірних масивів, у тому числі за допомогою конструкцій “вказівник на масив” і “вказівник на вказівник”. Враховувати, що кожен рядок матриці можна опрацьовувати як окремий елемент.

5. Знайти найбільший елемент головної діагоналі матриці $A[5][5]$ і вивести на друк весь рядок, де він знаходиться.

Блок схеми (до 1,2 завдання)



ТЕКСТ ПРОГРАМИ

Завдання 1

```
#include <stdio.h>

#define ARRAY_SIZE 10

void swapArrayElements(int arr[], int oldIdx, int newIdx)
{
    int temp = arr[oldIdx];
    arr[oldIdx] = arr[newIdx];
    *(arr + newIdx) = temp;
}

void reverse(int arr[], int start, int end)
{
    for (int i = start; i <= (start + end) / 2; i++)
    {
        swapArrayElements(arr, i, start + end - i);
    }
}

void cyclicShift(int arr[], int size, int shift)
{
    reverse(arr, 0, size - 1);
    reverse(arr, shift, size - 1);
    reverse(arr, 0, shift - 1);
}

void printArray(int arr[], int size)
{
    printf("array[%d]{", size);

    for (int i = 0; i < size; i++)
    {
        printf("%d", arr[i]);
        if (i != size - 1)
        {
            printf(", ");
        }
    }

    printf("}\n");
}

int main(void)
{
    printf("Task 5 from 1st section\n");
    int array[ARRAY_SIZE] = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9};

    printf("Array before cyclic 2 shift\n");
    printArray(array, ARRAY_SIZE);

    cyclicShift(array, ARRAY_SIZE, 2);

    printf("Array after\n");
    printArray(array, ARRAY_SIZE);
    return 0;
}
```

Завдання 2

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

int genRandomInt()
{
    return rand() % 10000;
}

int **generateMatrix(int x, int y)
{
    int **matrix = (int **)malloc(y * sizeof(int *));

    for (int i = 0; i < y; i++)
    {
        matrix[i] = (int *)malloc(x * sizeof(int));
    }

    for (int i = 0; i < y; i++)
    {
        for (int j = 0; j < x; j++)
        {
            (*(matrix + i) + j) = genRandomInt();
        }
    }

    return matrix;
}

void freeMatrix(int **matrix, int x, int y)
{
    for (int i = 0; i < x; i++)
    {
        free(matrix[i]);
    }
    free(matrix);
}

void printRowWithBiggestDiagNum(int **matrix, int x, int y)
{
    if (x != y)
    {
        printf("Error: matrix must be square\n");
        return;
    }

    int maxElementRowIdx = 0;
    for (int i = 0; i < x; i++)
    {
        if (matrix[i][i] > matrix[maxElementRowIdx][maxElementRowIdx])
        {
            maxElementRowIdx = i;
        }
    }

    printf("\nPrinting row with biggest diagonal number: \n");
    for (int i = 0; i < x; i++)
    {
        printf("%d ", matrix[maxElementRowIdx][i]);
    }
    printf("\n");
}
```

```

void printMatrix(int **matrix, int x, int y)
{
    for (int i = 0; i < y; i++)
    {
        for (int j = 0; j < x; j++)
        {
            printf("%d\t", matrix[i][j]);
        }
        printf("\n");
    }
}

int main(void)
{
    printf("Task 5 from 2nd section\n\n");

    const MATRIX_SIZE = 5;
    int **matrix = generateMatrix(MATRIX_SIZE, MATRIX_SIZE);

    printMatrix(matrix, MATRIX_SIZE, MATRIX_SIZE);

    printRowWithBiggestDiagNum(matrix, MATRIX_SIZE, MATRIX_SIZE);

    freeMatrix(matrix, MATRIX_SIZE, MATRIX_SIZE);
    return 0;
}

```

РЕЗУЛЬТАТИ

```

[keep-simple@pc ~d/c/labs/1_semester/3]$ ./"1"
Task 5 from 1st section
Array before cyclic 2 shift
array[10]{0, 1, 2, 3, 4, 5, 6, 7, 8, 9}
Array after
array[10]{8, 9, 0, 1, 2, 3, 4, 5, 6, 7}

```

Рис 1. Результат виконання програми №1

```

[keep-simple@pc ~d/c/labs/1_semester/3]$ ./"2"
Task 5 from 2nd section

9383      886      2777      6915      7793
8335      5386      492       6649      1421
2362       27      8690       59       7763
3926      540      3426      9172      5736
5211      5368      2567      6429      5782

Printing row with biggest diagonal number:
9383 886 2777 6915 7793

```

Рис 2. Результат виконання програми №2

ВИСНОВКИ

Закріплено поняття масивів та освоєно основні методи програмування алгоритмів обробки масивів даних засобами мови C.