**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"**

Інститут **КНІТ**

Кафедра **ПЗ**

**ЗВІТ**

До 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-15

Марущак А.С.

**Прийняла:**

асист. каф. ПЗ

Заводська Н.О.

« \_\_\_\_ » \_\_\_\_\_\_\_\_ 2021 р.

∑= \_\_\_\_ .

Львів – 2021

**Тема роботи:** робота з масиами в С.

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

**Теоретичні відомості**

Масив являє собою сукупність однотипних змінних, розміщених у послідовно пронумерованих суміжних комірках пам'яті. Номер елемента масиву задається індексом. Індексація елементів масиву в С починається з 0. Якщо у масиві N елементів, то перший елемент матиме індекс 0, а останній – індекс (N–1). За способом зв'язування індексів з комірками пам'яті виділяють три категорії масивів: статичні, фіксовані автоматичні та динамічні. Статичним називають масив, в якому зв’язування індексів та розміщення в пам'яті виконуються на етапі компіляції програми. Статичні масиви досить ефективні, оскільки для їх створення та знищення не потрібно додаткових операцій. Фіксованим автоматичним називають масив, в якому індекси зв'язуються статично, а розміщення в пам'яті виконується при обробці оголошень всередині функцій. Динамічним називають масив, де зв’язування індексів і розміщення в пам'яті виконується безпосередньо під час виконання програми. Цей спосіб організації даних достатньо гнучкий, оскільки розміри динамічного масиву можуть збільшуватися та зменшуватися під час виконання програми у міру необхідності.

Одновимірні масиви оголошуються таким чином:

тип ім’я\_масиву[розмір]

Наприклад, так оголошується статичний масив з іменем numbs, що складається з 10 елементів типу int:

int numbs[10];

У С доступ до елементів масиву здійснюється за допомогою спеціального оператора []. В квадратних дужках вказується індекс елемента масиву, а перед ними – його ім'я. Наприклад, присвоєння першому елементу масиву numbs значення 12 виконується так:

numbs[0] = 12;

Масиви тісно пов’язані із вказівниками. Ім’я масиву є вказівником на його перший елемент. Тому ім'я масиву можна використати в якості бази для зміщення вказівника. Наприклад, вираз

numbs[4] = 1

еквівалентний виразу

\*(numbs+4) = 1;

Часто за допомогою мов програмування виникає потреба обробляти таблиці або матриці. Для їх представлення у мові С передбачені багатовимірні масиви. Найпоширеніший варіант багатовимірного масиву – двовимірний масив, який можна представити у вигляді масиву одновимірних масивів:

тип ім’я\_масиву [розмір1][розмір2];

Звертаються до елементів масиву так:

ім’я\_масиву [індекс1][індекс2]

**Лабораторне завдання**

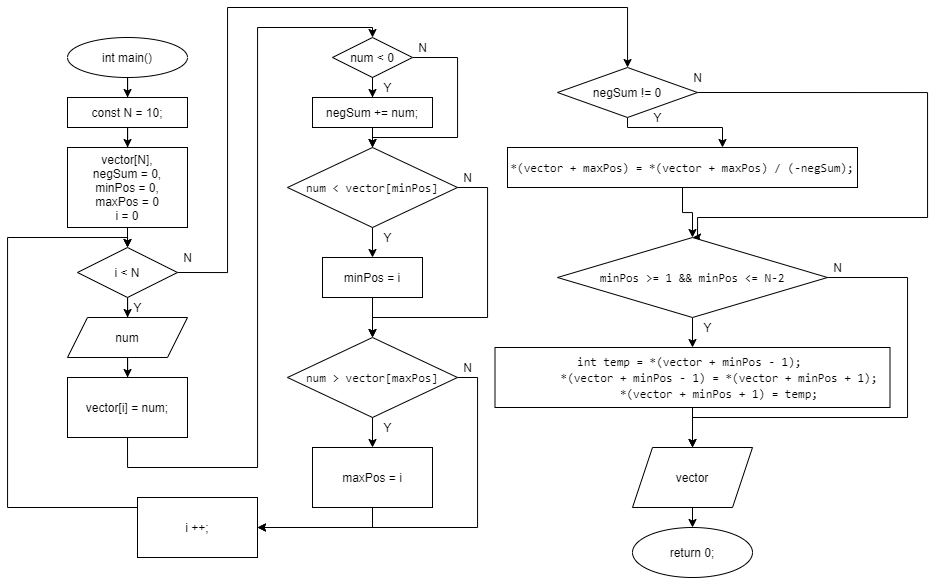
1. Ознайомитися з теоретичним матеріалом викладеним вище в даній інструкції і виконати приклади програм.
2. Одержати індивідуальне завдання з Додатку 1.
3. Розробити алгоритм розв’язання індивідуального завдання і подати його у вигляді блоксхеми
4. Скласти програму на мові С у відповідності з розробленим алгоритмом.
5. Виконати обчислення по програмі.
6. Одержати індивідуальне завдання з Додатку 2.
7. Розробити алгоритм розв’язання індивідуального завдання і подати його у вигляді блоксхеми.
8. Скласти програму на мові С у відповідності з розробленим алгоритмом
9. Виконати обчислення по програмі при різних значеннях точності і порівняти отримані результати.
10. Підготувати та здати звіт про виконання лабораторної роботи

**Індивідуальне завдання**

**№1**

21. Ввести вектор дійсних чисел. Поміняти місцями сусідні елементи мінімального елемента, а максимальний елемент поділити на суму абсолютних значень від’ємних елементів

**Блок-схема:**



**Код програми: (назва файлу Lab03\_01.c):**

#include <stdio.h>

#include <limits.h>

#define LIMIT 1000

int main()

{

    double vector[LIMIT], num = 0, negSum = 0;

    int minPos = 0, maxPos = 0, number = 0;

    do{

        printf("How many numbers should be in vector?");

        scanf("%i", &number);

    }while(number <= 0);

    printf("Enter %i numbers: ", number);

    for(int i = 0; i < number; i++)

    {

        scanf("%lf", &num);

        if(num < 0){

            negSum += num;

        }

        vector[i] = num;

        minPos = num < vector[minPos] ? i : minPos;

        maxPos = num > vector[maxPos] ? i : maxPos;

    }

    if(negSum != 0)

    {

        \*(vector + maxPos) = \*(vector + maxPos) / (-negSum);

    }

    if(minPos >= 1 && minPos <= number-2)

    {

        int temp = \*(vector + minPos - 1);

        \*(vector + minPos - 1) = \*(vector + minPos + 1);

        \*(vector + minPos + 1) = temp;

    }

    printf("Result: ");

    for (int i = 0; i < number; i++)

    {

        printf("%lf ", vector[i]);

    }

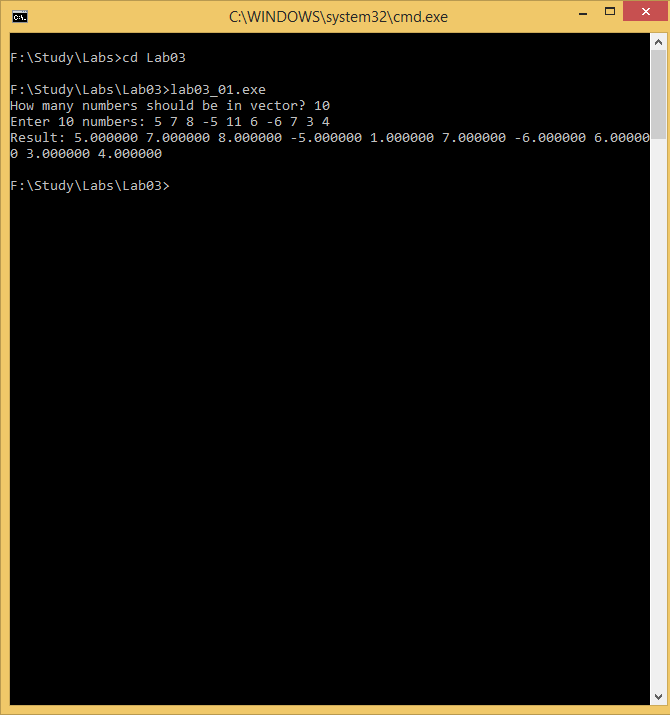
    printf("\n");

    return 0;

}

**Протокол роботи:**

Програма отримує n чисел, ділить максимальне на модуль суми всіх від’ємних елементів, а сусідні до мінімального міняє місцями.



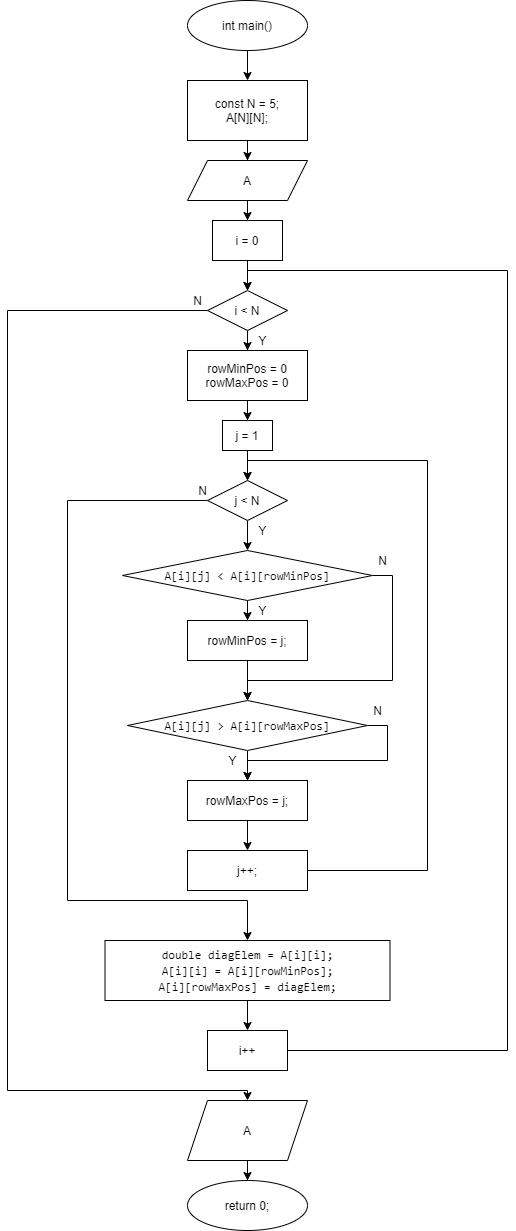
І дійсно, 1, а елементи навколо -6 помінялись місцями.

**Висновок:** ми написали програму, яка зчитує вектор чисел, проводить над ним операції та виводить результат. Це допомогло нам навчитися використовувати масиви: оголошувати їх, вводити в нього значення, змінювати їх та виодити.

**№2**

21. Знайти найменші елементи кожного рядка квадратної матриці А і помістити їх на головну діагональ, а діагональні елементи на місця найбільших

**Блок-схема:**

****

**Код програми: (назва файлу Lab03\_02.c):**

#include <stdio.h>

#define LIMIT 100

int main()

{

    double A[LIMIT][LIMIT];

    int size;

    do{

        printf("Size pf matrix: ");

        scanf("%d", &size);

    }while(size <= 0);

    printf("Enter matrix %dx%d:\n", size, size);

    for(int i = 0; i < size; i++){

        for(int j = 0; j < size; j++){

            scanf("%lf", (\*(A+i) + j));

        }

    }

    for(int i = 0; i < size; i++){

        int rowMinPos = 0, rowMaxPos = 0;

        for(int j = 1; j < size; j++){

            rowMinPos = A[i][j] < A[i][rowMinPos] ? j : rowMinPos;

            rowMaxPos = A[i][j] > A[i][rowMaxPos] ? j : rowMaxPos;

        }

        double diagElem = A[i][i];

        A[i][i] = A[i][rowMinPos];

        A[i][rowMaxPos] = diagElem;

    }

    for(int i = 0; i < size; i++){

        for(int j = 0; j < size; j++){

            printf("%lf ", \*(\*(A+i) + j));

        }

        printf("\n");

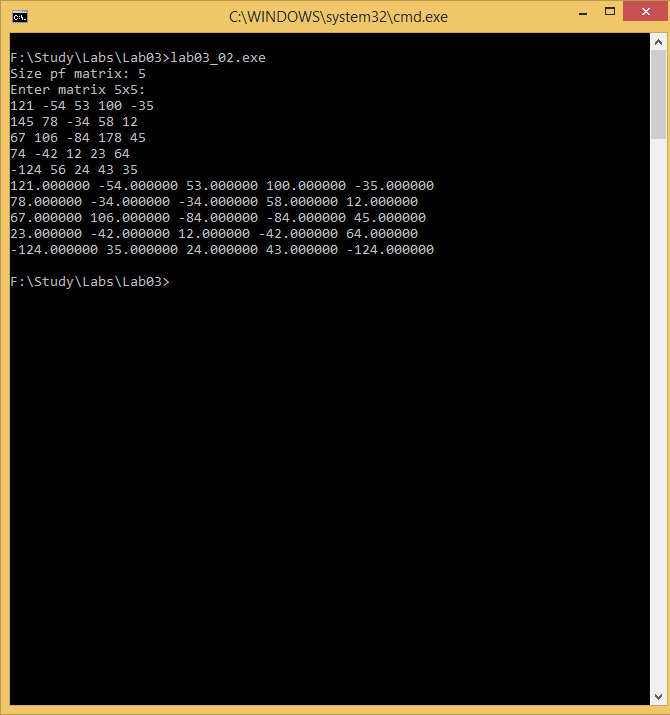
    }

    return 0;

}

**Протокол роботи:**

Програма зчитує, якого розміру матрицю треба ввести, зчитує необхідну к-ть чисел і виводить результат згідно з правилом.



У першому рядку елемент -54 став на діагональ, але діагональний елемент є найбільшим, тому він повернув собі попереднє значення. У другому найменший елемент став на діагональ, а діагональний замінив найбільший. В інших рядках програма також відпрацювала без помилок.

**Висновок:** ми написали програму, що зчитує масив, редагує і виводить результат. Це допомогло нам використовувти же двовимірні масиви, що корисно для представлення табиць, матриць і т.п.