НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Кафедра «ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ ТА ПРОГРАМУВАННЯ»

«Програмування ч.1»

Звіт з лабораторної роботи №11
Тема: «Вступ до показників»

Виконав:
ст. гр. КІТ-120А
Старовойтов Н.А.

Перевірив:
Челак В.В.

Харків – 2020

Мета: Отримати навички роботи з показниками та їх використанням у розв’язуванні задач, зокрема тих, що містять завдання на масиви.

Індивідуальне завдання

Робота на оцінку “відмінно”.

Завдання №3: «Дано масив з N речовинних чисел. Підрахувати кількість ділянок, які утворюють безперервні послідовності чисел з не-зменшуваними значеннями. Максимальну ділянку переписати у інший масив».

Опис програми

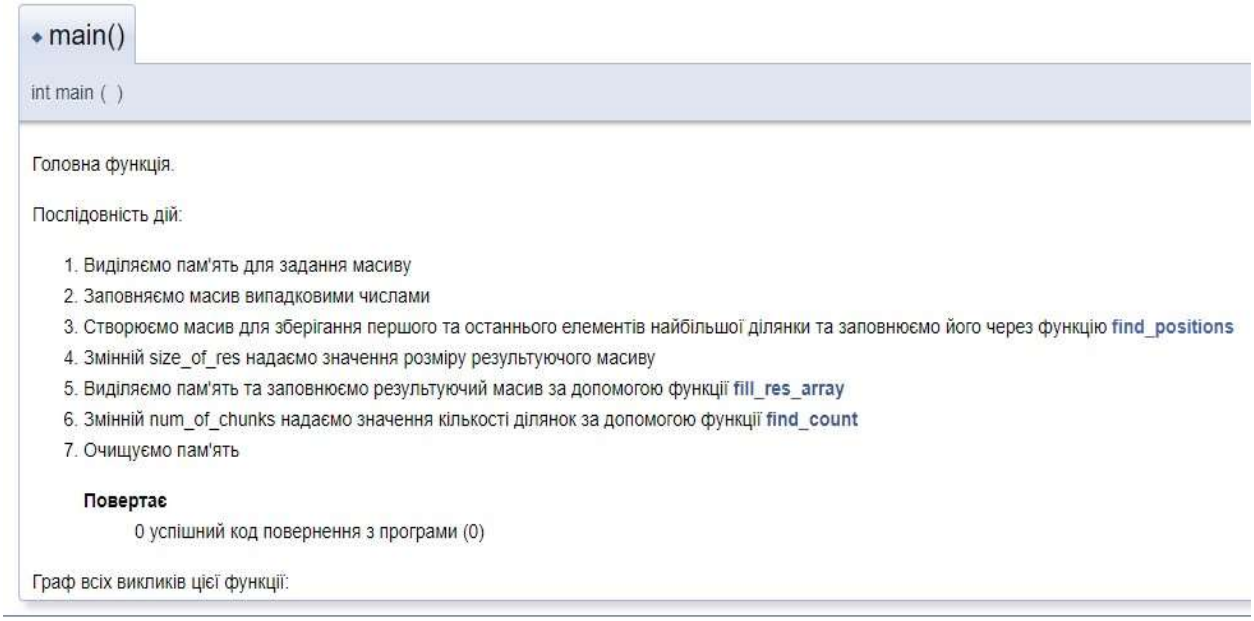

Рисунок 1 — Функція main

Опис функцій

fill_res_array()

```
void fill_res_array ( float *  array,
                      float *  result_arr,
                      int      a,
                      int      b
                    )
```

Заповнення результуючого масиву числами з найбільшої ділянки

Аргументи

array - масив

result_arr - результуючий масив

a - перший елемент ділянки

b - останній елемент ділянки

find_count()

```
float find_count ( float *  array )
```

Знаходження кількості ділянок, які утворюють безперервні послідовності чисел з не-зменшуваними значеннями

Аргументи

array - вхідний масив

Повертає

counter - кількість ділянок

find_positions()

```
void find_positions ( float *  array,
                      int      pos_min_max[]
                    )
```

Знаходження номерів елементів масиву, які є початком і кінцем найбільшої ділянки

Аргументи

array - вхідний масив

pos_min_max - масив, в який будуть записані перший та останній елементи найбільшої ділянки

Рисунок 2 — Функції fill_res_array, find_count та find_positions

Схеми алгоритмів функцій

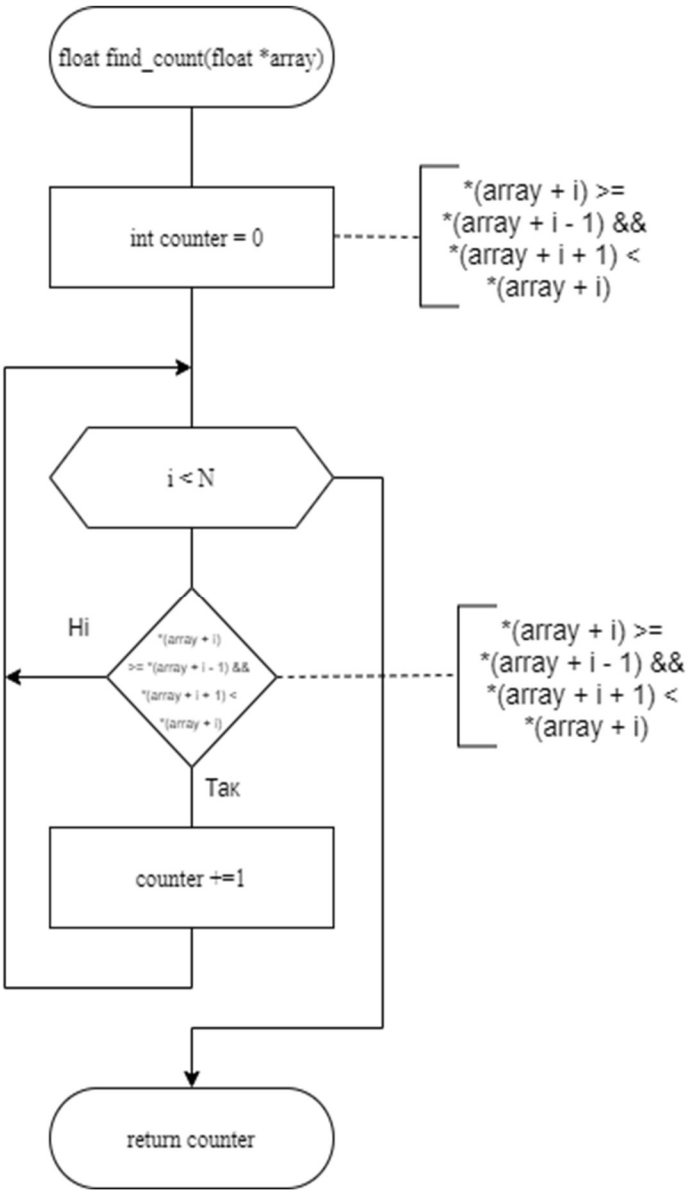

Рисунок 3 — Блок-схема функції find_count

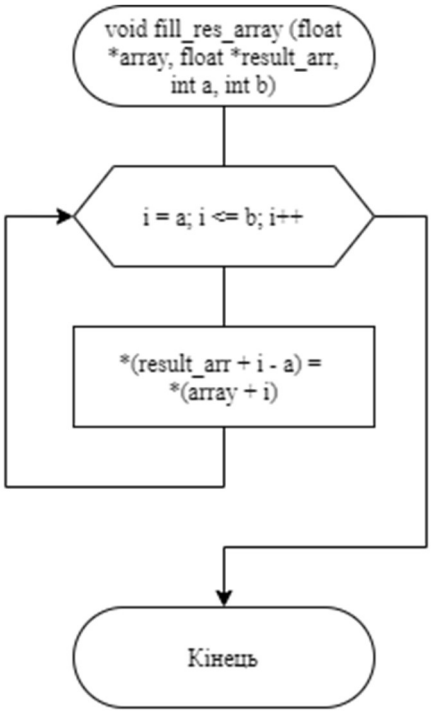

Рисунок 4 — Блок-схема функції fill_res_array

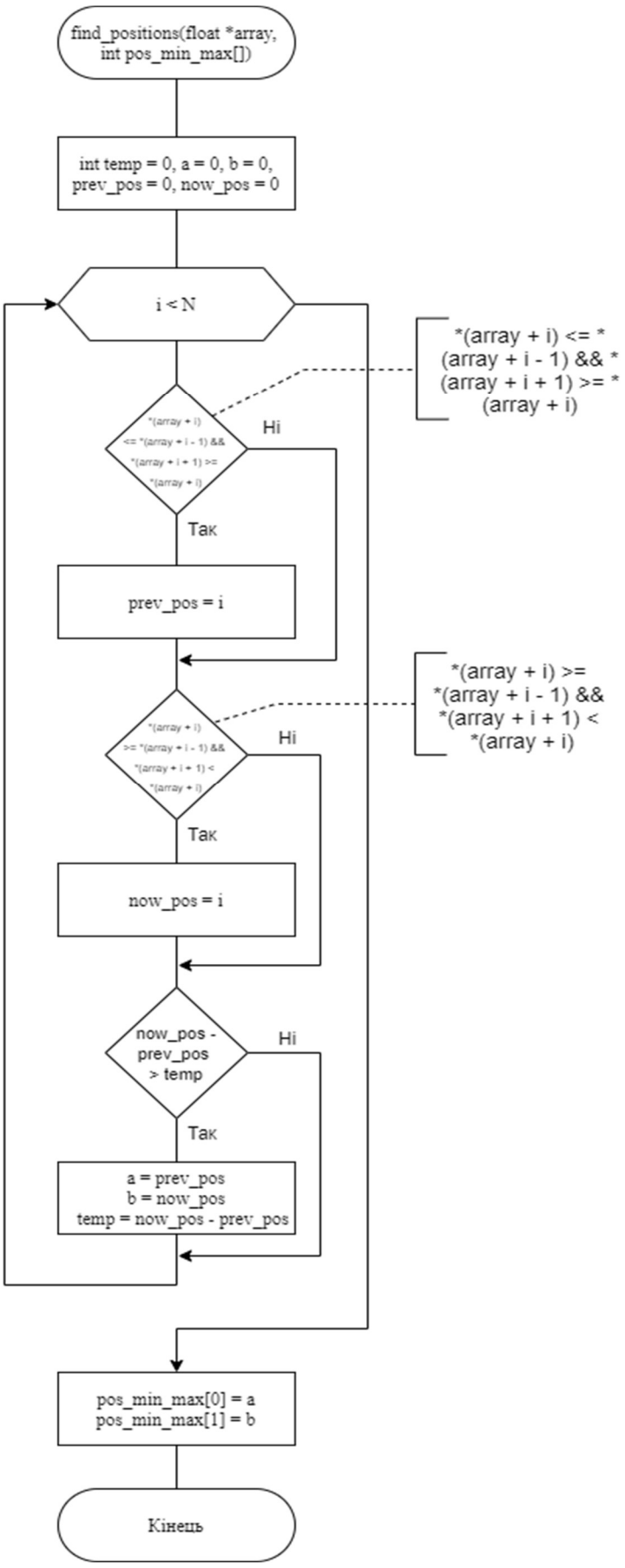

Рисунок 5 — Блок-схема функції find_positions

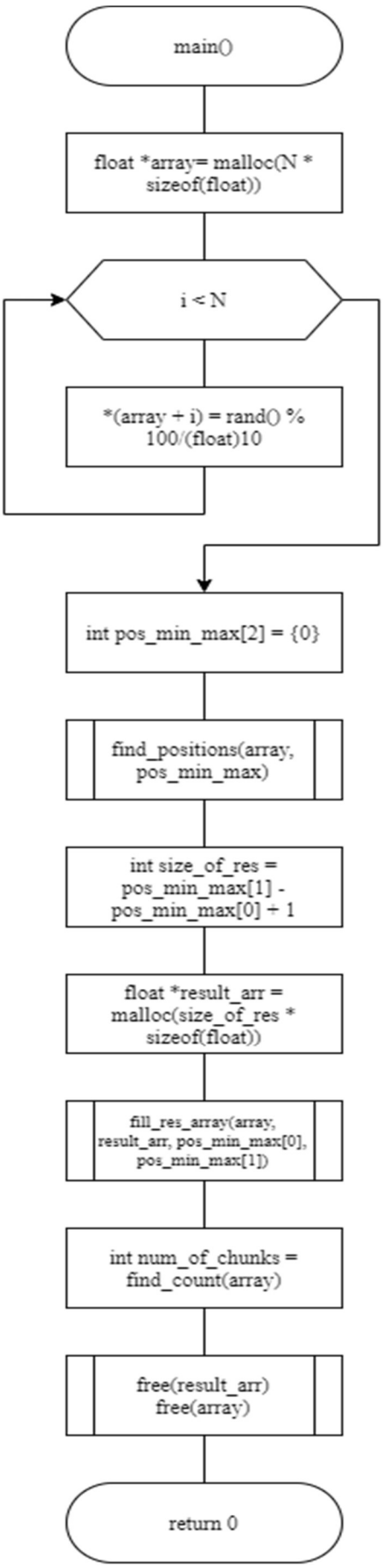

Рисунок 6 — Блок-схема функції main, яка складається з трох програм

Текст програми

```
#include <stdlib.h>
#define N 12

float find_count(float *array)
{   /*
    float DBG_array[N] = {0};
    for (int i = 0; i < N; i++){
        DBG_array[i] = *(array + i);
    }
    */
    int counter = 0;
    for (int i = 1; i < N; i++) {
        if (*(array + i) >= *(array + i - 1) && *(array + i + 1) < *(array + i)) {
            counter +=1;
        }
    }
    return counter;
}

void find_positions(float *array, int pos_min_max[])
{   /*
    float DBG_array[N] = {0};
    for (int i = 0; i < N; i++){
        DBG_array[i] = *(array + i);
    }
    */
    int temp = 0, a = 0, b = 0;
    int prev_pos = 0;
    int now_pos = 0;
    for (int i = 1; i < N; i++) {
        if (*(array + i) <= *(array + i - 1) && *(array + i + 1) >= *(array + i))
            prev_pos = i;
        if (*(array + i) >= *(array + i - 1) && *(array + i + 1) < *(array + i)){
            now_pos = i;
            if (now_pos - prev_pos > temp) {
                a = prev_pos;
                b = now_pos;
                temp = now_pos - prev_pos;
            }
        }
    }
    pos_min_max[0] = a;
    pos_min_max[1] = b;
}

void fill_res_array (float *array, float *result_arr, int a, int b){
    for (int i = a; i <= b; i++ ){
        *(result_arr + i - a) = *(array + i);
    }
}

int main()
{
    float *array= malloc(N * sizeof(float));
    for (int i = 0; i < N; i++){
        *(array + i) = rand() % 100/(float)10 ;
    }
    /*
```

```
    float DBG_array[N] = {0};
    for (int i = 0; i < N; i++){
        DBG_array[i] = *(array + i);
    }
     */
    int pos_min_max[2] = {0};
    find_positions(array, pos_min_max);

    int size_of_res = pos_min_max[1] - pos_min_max[0] + 1;
    float *result_arr = malloc(size_of_res * sizeof(float));

    fill_res_array(array, result_arr, pos_min_max[0], pos_min_max[1]);
    int num_of_chunks = find_count(array);
    /*
    float DBG_result_array[N] = {0};
    for (int i = 0; i < N; i++){
        DBG_result_array[i] = *(result_arr + i);
    }
    */
    free(result_arr);
    free(array);
    return 0;
}
```

Результати роботи програми

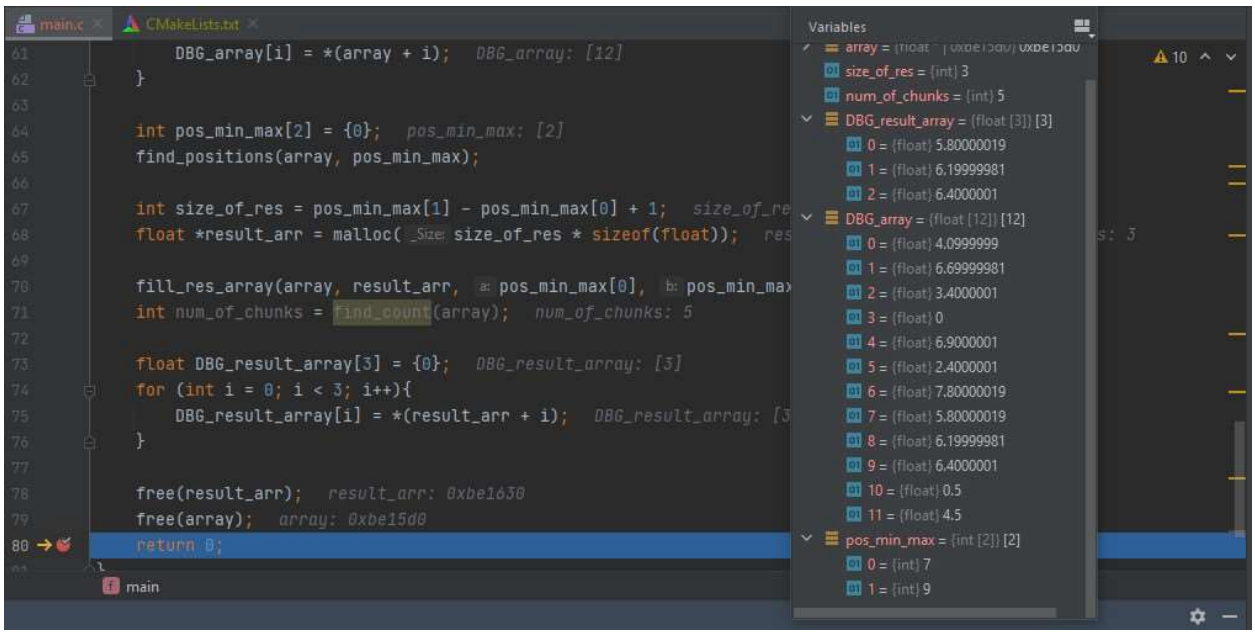

Рисунок 7 — Результат успішного виконання програми

Висновки

Під час виконання даної лабораторної роботи було отримано навички роботи з показниками та їх використанням у розв’язуванні задач, зокрема тих, що містять завдання на масиви.