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 іх 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**

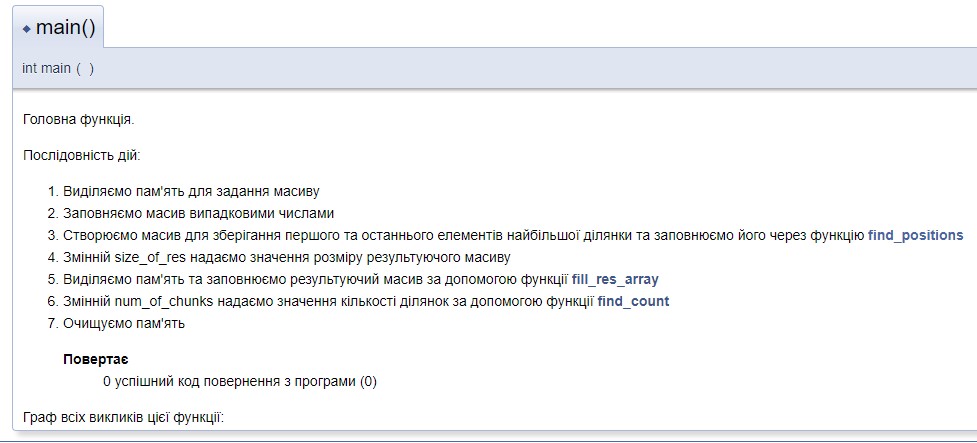
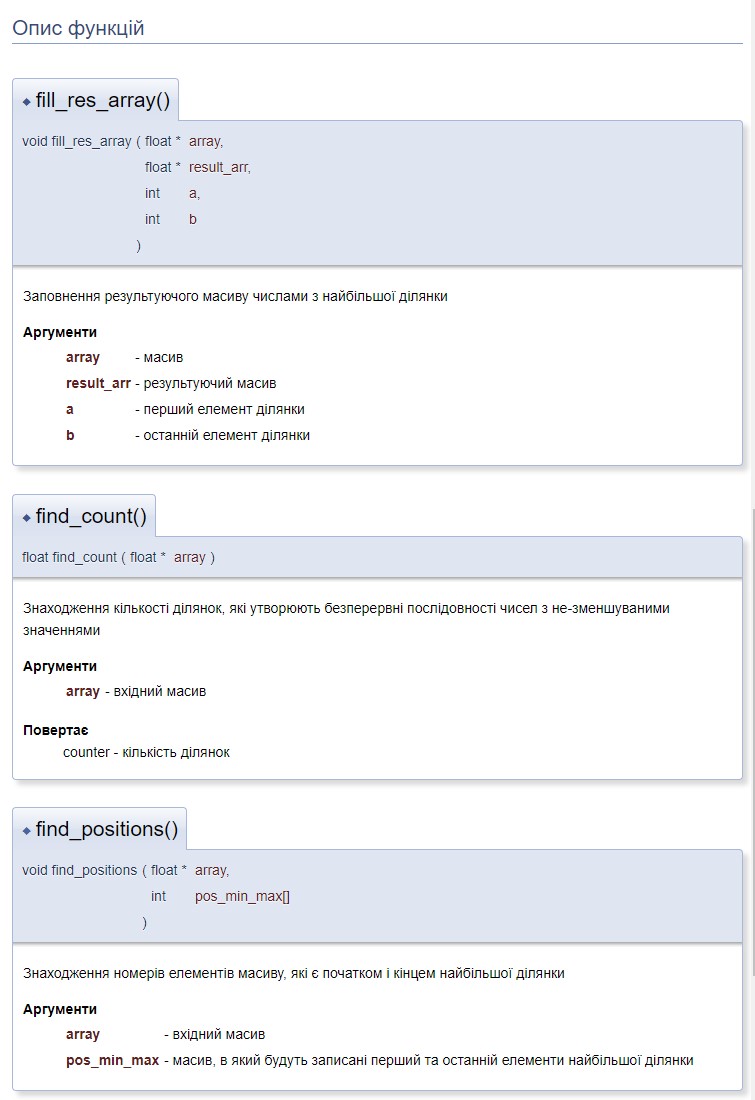


Рисунок 1 — Функція main

 Рисунок 2 — Функції fill\_res\_array, find\_count та find\_positions

**Схеми алгоритмів функцій**

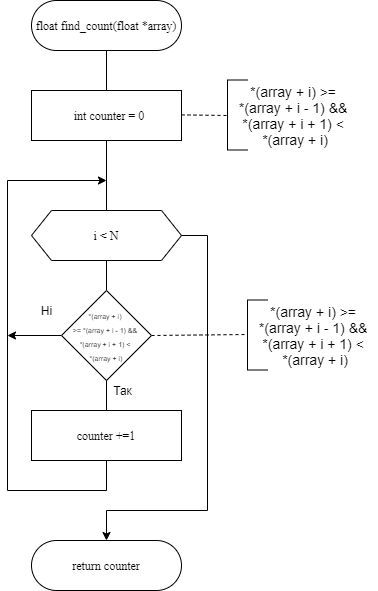
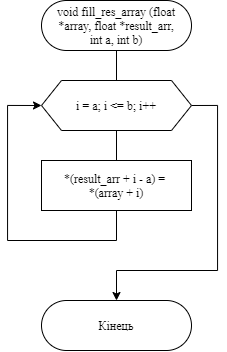
** **

Рисунок 3 — Блок-схема функції find\_count Рисунок 4 — Блок-схема функції fill\_res\_array

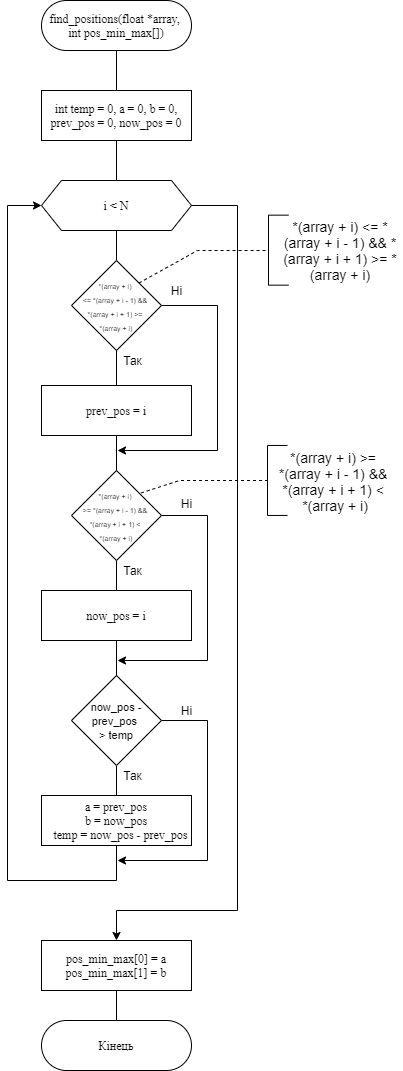
****

Рисунок 5 — Блок-схема функції find\_positions

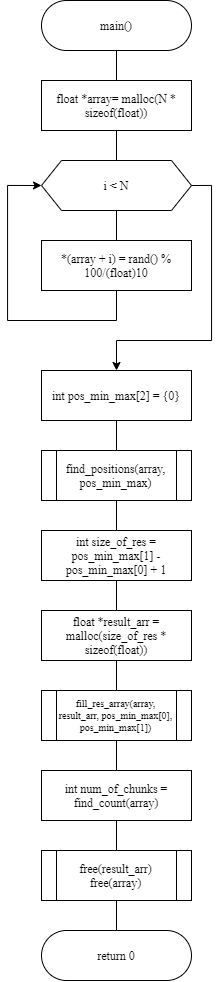
****

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**

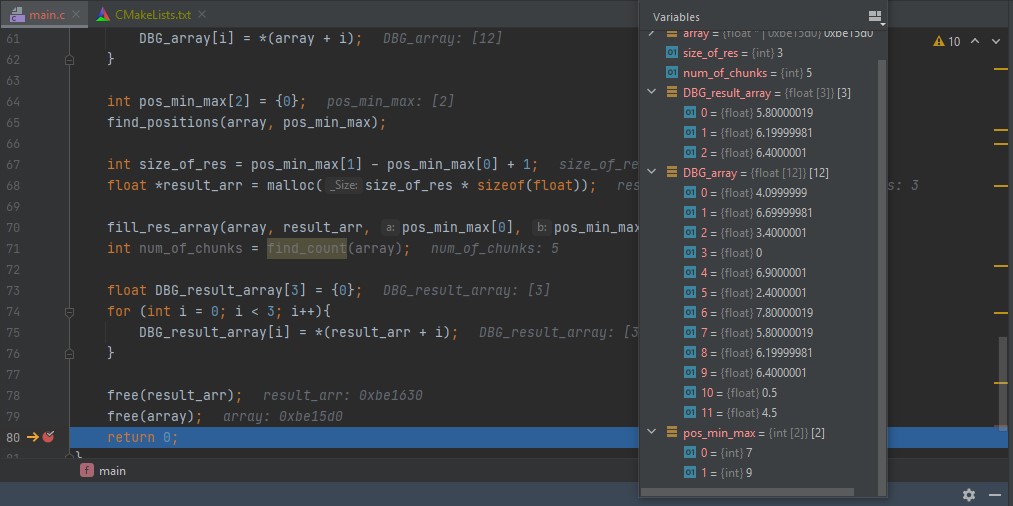


Рисунок 7 — Результат успішного виконання програми

**Висновки**

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