НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Кафедра «ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ ТА ПРОГРАМУВАННЯ»

«Програмування ч.1»

*Звіт з лабораторної роботи №12*

*Тема: «Взаємодія з користувачем шляхом*

*механізму введення/виведення»*

Виконав:

ст. гр. КІТ-120А

Старовойтов Н.А.

Перевірив:

Челак В.В.

Харків – 2021

***Мета:*** Отримати навички взаємодії з користувачем шляхом механізму введення/виведення та його використанням у розв’язуванні задач.

***Індивідуальне завдання***

Робота на оцінку “відмінно”.

1) Програму, яка була розроблена у попередній лабораторній роботі (робота з покажчиками), змінити так, щоб:

• початкові дані вводилися з клавіатури;

• видача результуючих даних провадилася у консоль.

• при старті програми виводилась інформація об авторі, номері та темі лабораторної роботи;

• при запиті даних, користувач отримав повідомлення, що від нього очікують.

2) 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**

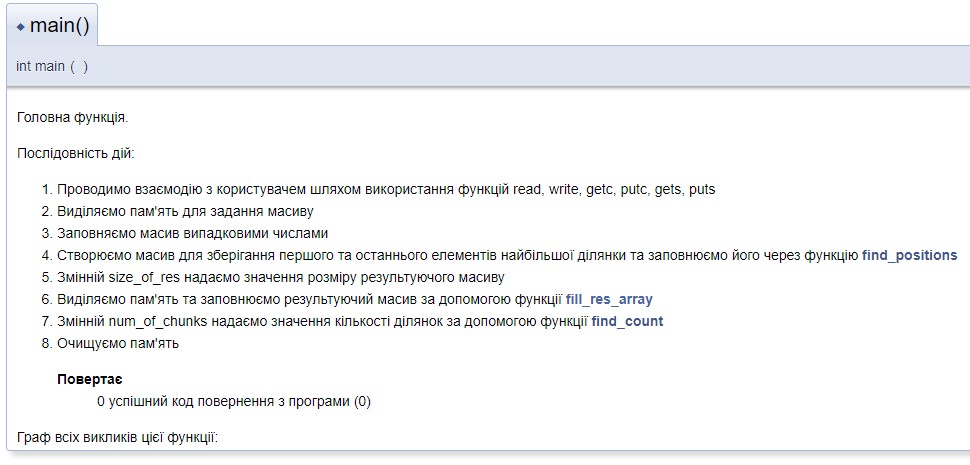
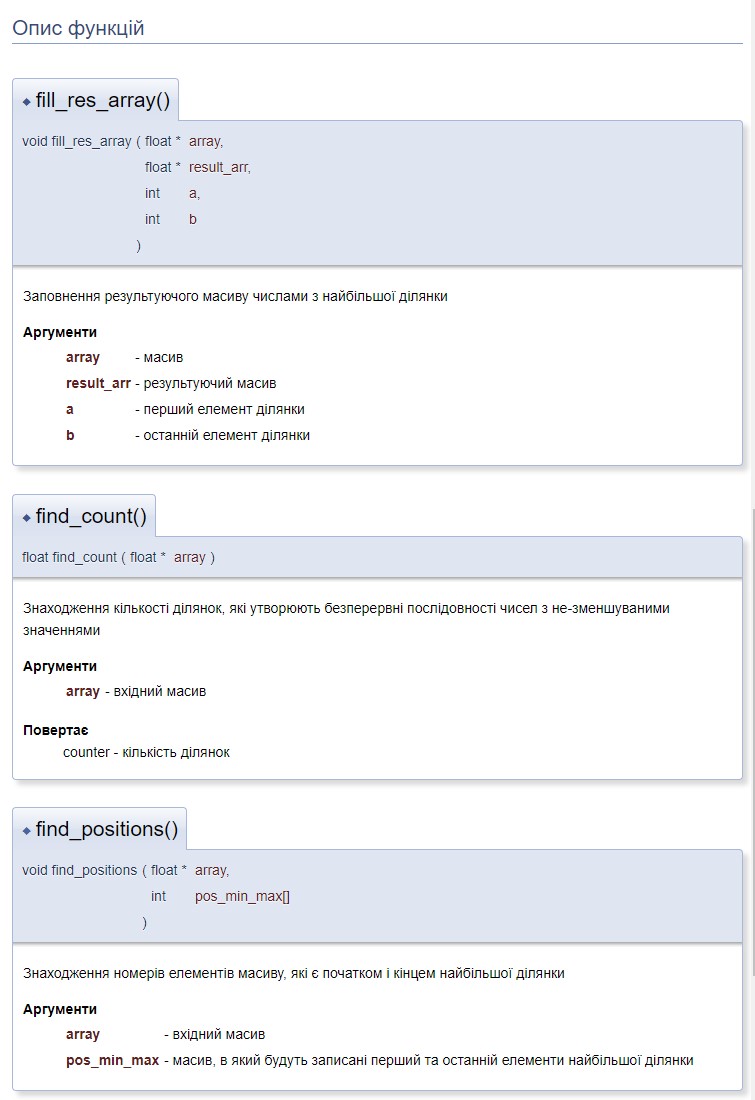


Рисунок 1 — Функція main

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

**Схеми алгоритмів функцій**

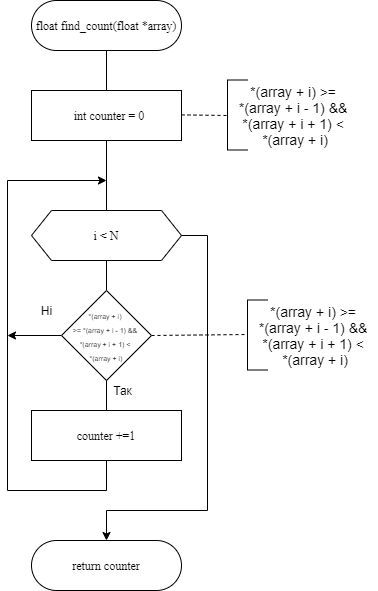
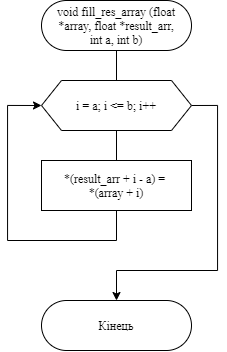
** **

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

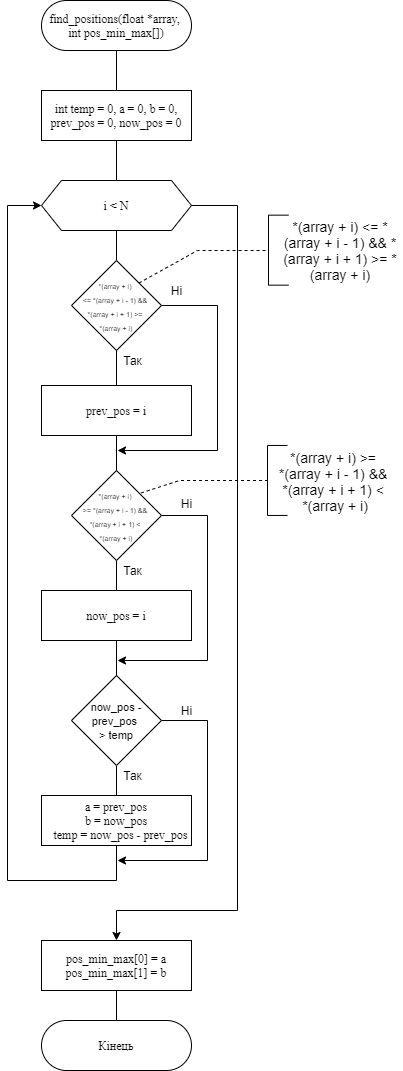
****

Рисунок 5 — Блок-схема функції find\_positions

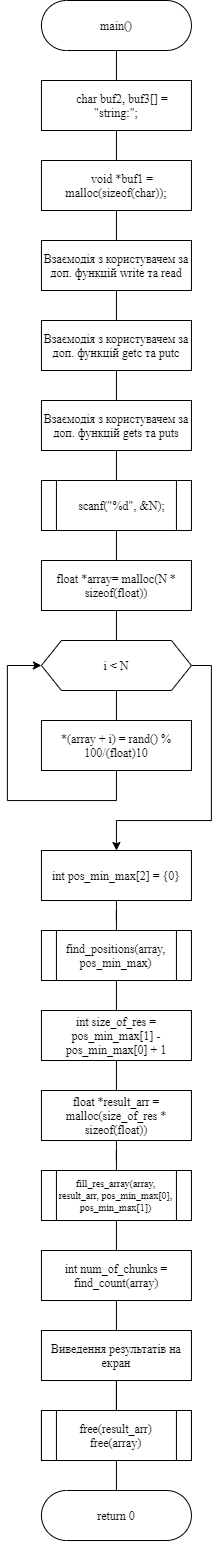
****

Рисунок 6 — Блок-схема функції main

**Текст програми**

#include <stdlib.h>

#include <stdio.h>

#include <io.h>

#include <string.h>

unsigned int N;

int 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()

{

printf("Lab 12\n\"Interaction with the user using the I/O mechanism\"\nAuthor: Starovoytov N.\n\n");

char buf2, buf3[] = "string:";

void \*buf1 = malloc(sizeof(char));

char test1[] = "Testing user interaction using \"write\" and \"read\" functions. \nPlease, enter any character/string:";

write(1, test1, strlen(test1));

read(0, buf1, 100);

printf("OK\n\nTesting user interaction using \"getc\" an \"putc\" functions. \nPlease, enter any character");

putc(':', stdout);

buf2 = getc(stdin);

fseek(stdin, 1, SEEK\_CUR);

printf("OK\n\nTesting user interaction using \"gets\" an \"puts\" functions. \nPlease, enter any character/");

puts(buf3);

gets(&buf2);

fflush(stdin);

printf("OK\n\nTesting complete! Starting program...\n\nPlease enter amount of numbers:");

scanf("%d", &N);

float \*array= malloc(N \* sizeof(float));

printf("Please enter the array elements:");

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

scanf("%f", (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);

printf("\nAmount of chunks: %d", num\_of\_chunks);

printf("\nThe biggest chunk: ");

for (int i = 0; i < size\_of\_res; i++){

printf("%f ", \*(result\_arr + i));

}

free(result\_arr);

free(array);

return 0;

}

**Результати роботи програми**

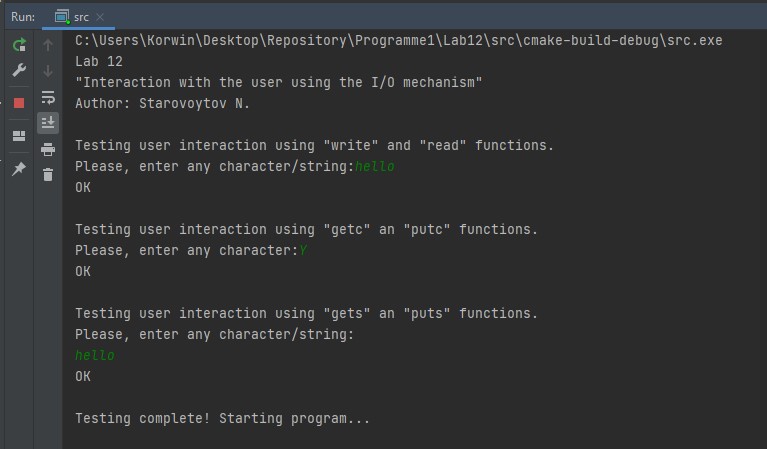


Рисунок 7 — Результат успішного виконання першої частини програми програми

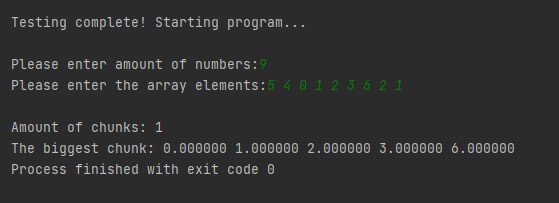


Рисунок 8 — Результат успішного виконання другої частини програми програми

**Висновки**

Під час виконання даної лабораторної роботи було отримано навички взаємодії з користувачем шляхом механізму введення/виведення та його використанням у розв’язуванні задач.