

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

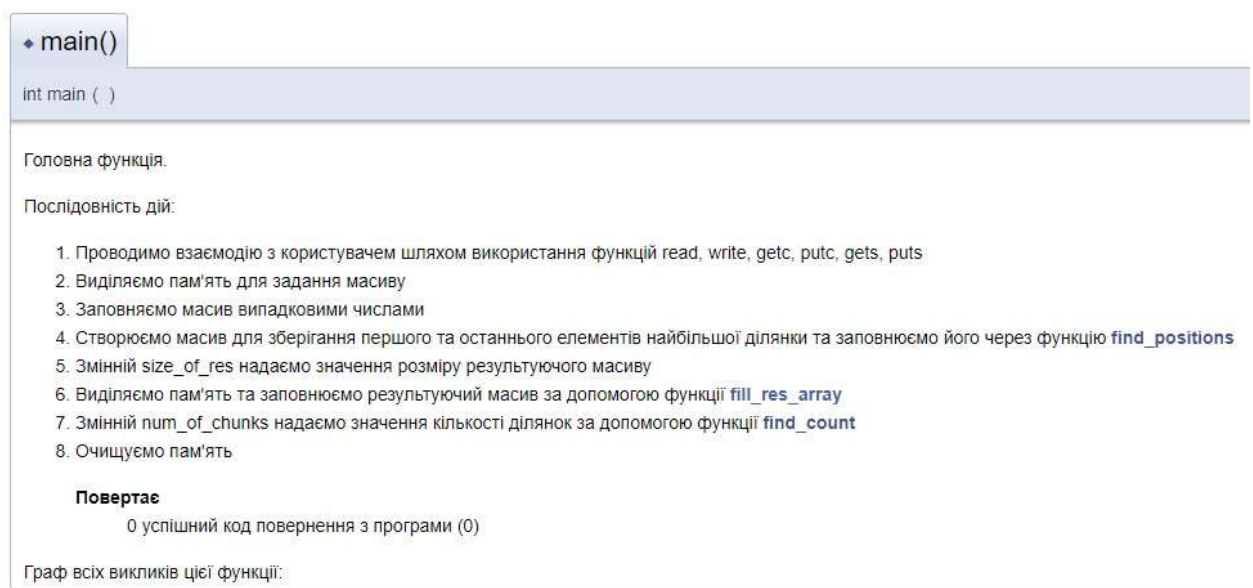


Рисунок 1 — Функція main

◆ 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

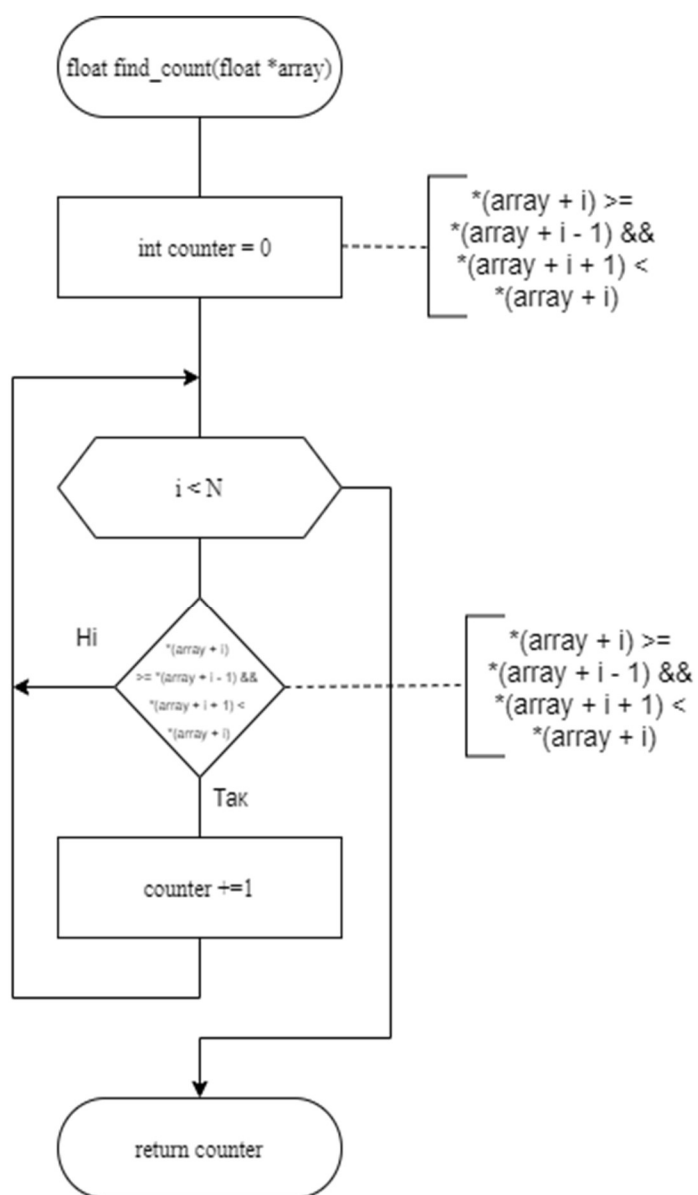


Рисунок 3 — Блок-схема функції find_count

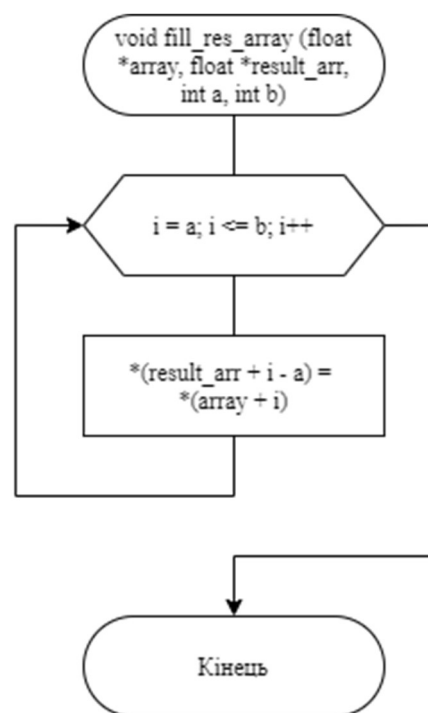


Рисунок 4 — Блок-схема функції fill_res_array

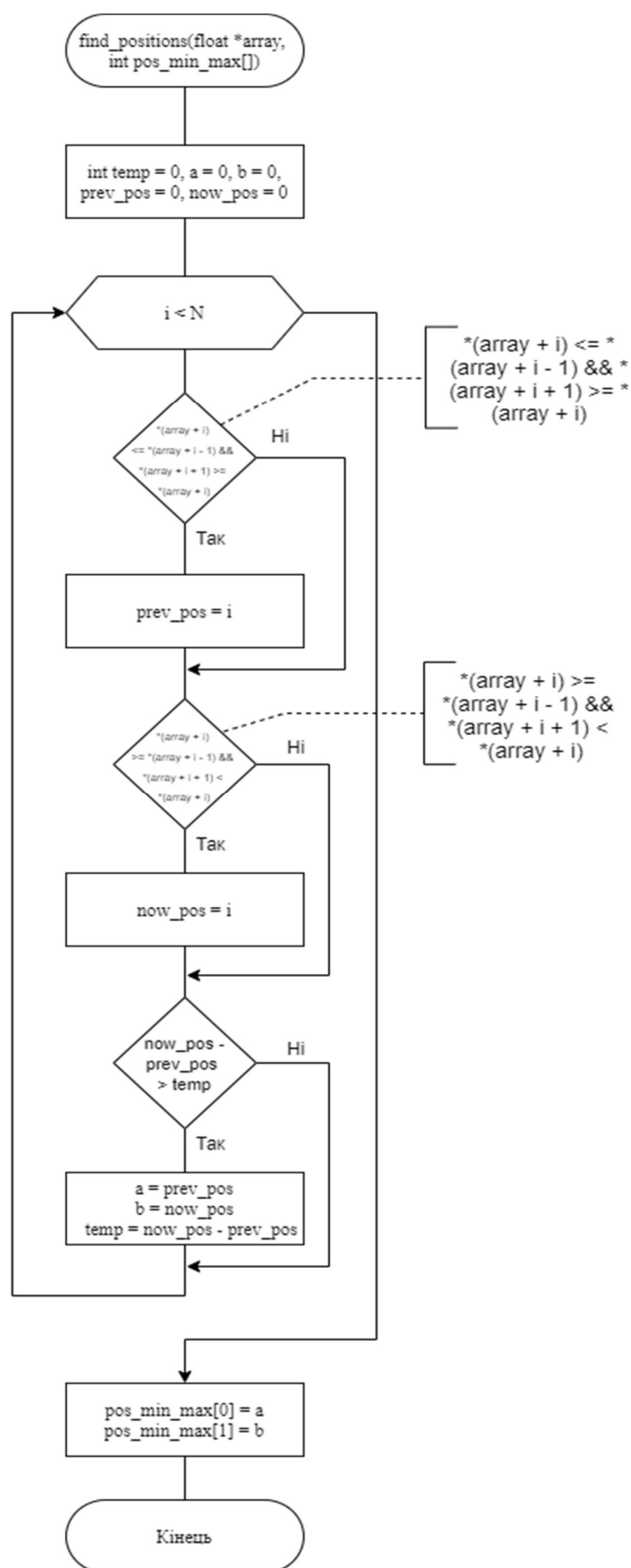


Рисунок 5 — Блок-схема функції `find_positions`

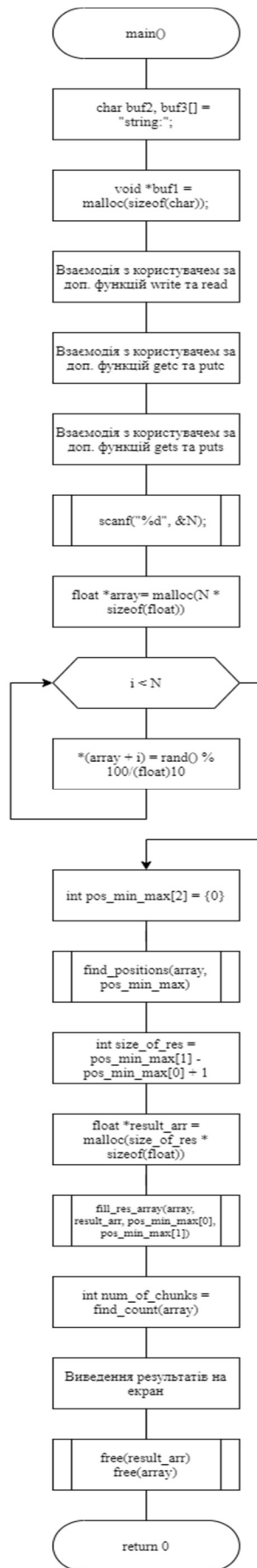


Рисунок 6 — Блок-схема функції main

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\n"Author: 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);

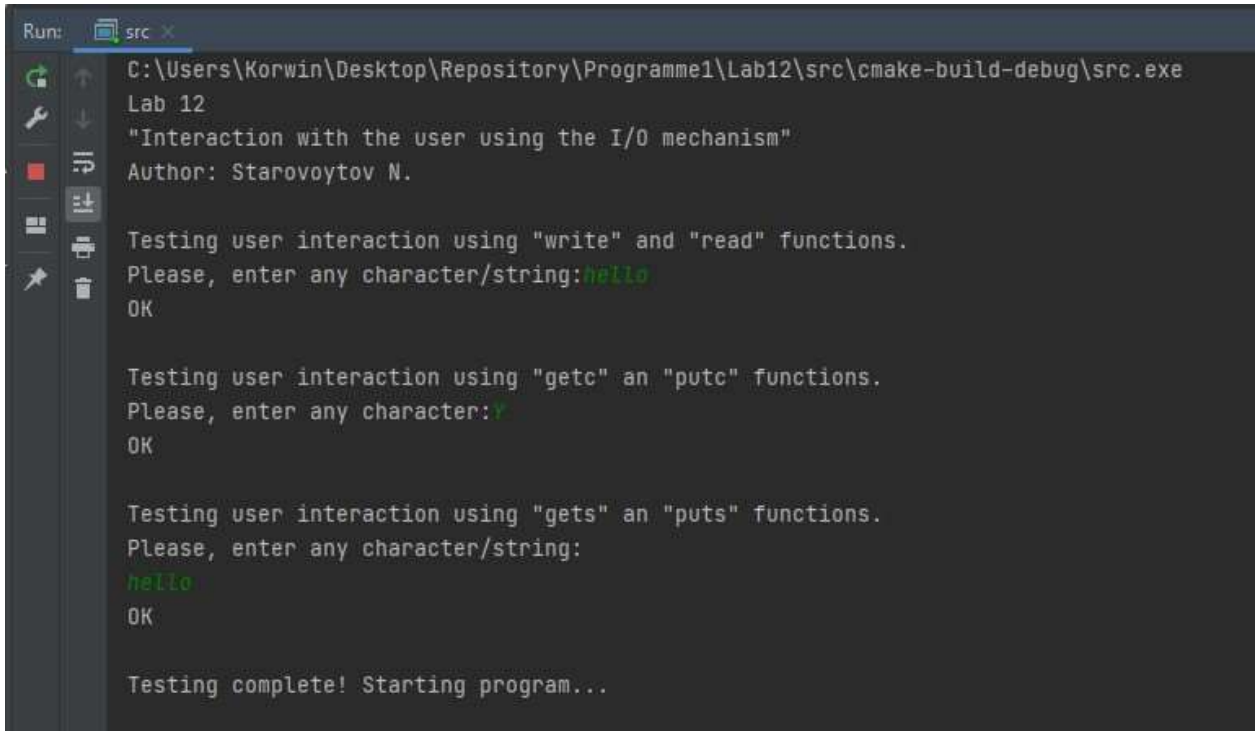
    int size_of_res = pos_min_max[1] - pos_min_max[0] + 1;

    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



```
Run: src x
C:\Users\Korwin\Desktop\Repository\Programme1\Lab12\src\cmake-build-debug\src.exe
Lab 12
"Interaction with the user using the I/O mechanism"
Author: Starovoytov N.

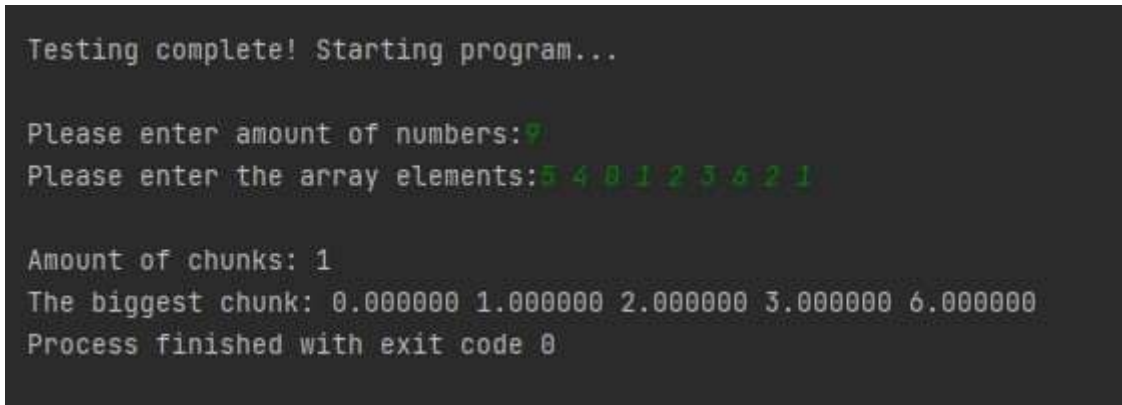
Testing user interaction using "write" and "read" functions.
Please, enter any character/string:hello
OK

Testing user interaction using "getc" an "putc" functions.
Please, enter any character:Y
OK

Testing user interaction using "gets" an "puts" functions.
Please, enter any character/string:
hello
OK

Testing complete! Starting program...
```

Рисунок 7 — Результат успішного виконання першої частини програми програми



```
Testing complete! Starting program...

Please enter amount of numbers:9
Please enter the array elements:5 4 0 1 2 3 6 2 1

Amount of chunks: 1
The biggest chunk: 0.000000 1.000000 2.000000 3.000000 6.000000
Process finished with exit code 0
```

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.