

Міністерство освіти і науки України
Національний технічний університет
«Харківський політехнічний інститут»
Кафедра «Обчислювальна техніка та програмування»

ЗВІТ
Про виконання лабораторної роботи №12
«Взаємодія з користувачем шляхом механізму введення/виведення»

Виконавець:
студент гр. КІТ-120В
Олексієнко Микита

Харків 2021

Лабораторна робота №11. Вступ до показчиків

1.Вимоги

1.1 Розробник

Олексієнко Микита Віталійович

студент групи КІТ-120В

17.01.2021

1.2 Загальне завдання

Реалізувати програму відповідно до індивідуального завдання.

1.3 Індивідуальне завдання

Разрахувати кількість ділянок, які утворюють безперервні послідовності чисел з не-зменшуваними значеннями. Максимальну ділянку переписати у інший масів.

2. Виконання роботи

2.1 Функціональне призначення

Програма призначена для того щоб обчислити безперервну послідовність чисел з не-зменшуваними значеннями.

2.2 Створення репозиторію

Створивши репозиторій на github, клоную його та створюю всі файли за вимогами структури проекту.

2.3 Написання коду програми

```
/**
```

```
 * @mainpage
```

```
 * # Загальне завдання
```

```
 * 1. **Разрахувати** кількість ділянок, які утворюють безперервні  
 послідовності чисел з не-зменшуваними значеннями. Максимальну  
 ділянку переписати у інший масів.
```

```
 * @author Alekseenko N.
```

```
 * @date 17-jan-2021
```

```
 * @version 1.0
```

```
 */
```

```

/**
 * @file main.c
 * @brief Файл з демонстрацією роботи структури та методів
 * оперування ним.
 *
 * @author Alekseenko N.
 * @date 17-jan-2021
 * @version 1.0
 */

#include <stdio.h>

#define N 4          // кількість чисел в масиві
#define M 10

/**
 * *Функція result*
 *
 * * Послідовність дій:
 *
 * - оголошення змінних
 *
 * \param r - показник на кінець максимальної послідовності
 * \param l - довжина поточної послідовності
 * \param c - кількість послідовностей
 * \param m - довжина максимальної послідовності
 *
 * - шукаємо позитивну послідовність з максимальною сумою
 *
 * - обробляємо останню послідовність
 *
 * - копіюємо максимальну послідовність у вихідний масив
 *
 * @return кількості послідовностей (c)
 */

```

```

int result (float *mas, float *res, int n, int *pm )
{
    float *p;

    float *r; // указатель
на конец максимальной последовательности

    float *q;

    int l = 1; // длина
текущей последовательности

    int c = 0; //
количество последовательностей

    int m = 0; // длина
максимальной последовательности

    for ( p = mas; p < mas + n - 1; p++ ) // поиск
положительной последовательности с максимальной суммой

    {

        if ( p[0] > p[1] ) // проверка
конца последовательности

        {

            c++;

            if (l > m) // ищем
максимальную последовательность

            {

                r = p;

                m = l;

                l = 1;

            }

        }

        else

        {

            l++;

        }

    }

    c++;

```

```

    if (l > m) // обробка останньої
последовательности
{
    r = p;
    m = l;
}
q = res;

for ( p = r - m + 1; p <= r ; p++ ) // копіювання максимальної
последовательности в вихідний масив
{
    *q++ = *p;
}

*pm = m;
return c;
}

/**
 * *Функція main*
 *
 * Послідовність дій:
 * - оголошення змінних
 * \param mas[] - вхідний масив
 * \param res[N] - результуючий масив
 * \param c - кількість послідовностей
 * \param m - довжина максимальної послідовності
 * @return успішний код повернення з програми (0)
 */
int main ()
{
    float mas[N] = {1.1 , 1.2 , 1.3 , 0.9 }; // вхідний
массив
    char name;

```

```

char lastname[M+1];

gets ( lastname );

printf("Ваша фамилия: ");

puts ( lastname );

name = getc ( stdin );

printf("Ваше имя: ");

putc ( name, stdout );

printf("\n\nВход в программу: 1\n");

printf("Выход из программы: 0\n");

char on;

read ( 0, &on, 1);

write ( 1, &on, 1);

if ( on == '0')
{
    printf("\n");

    return 0;
}

if ( on == '1')
{
    printf("\nВведите числа:\n");

    for ( int i = 0; i < N; i++)
    {
        scanf("%f", &mas[i] );
    }

    float res[ N ]; //
    результирующий массив

    int c = 1;

    int m = 0;

    c = result( mas, res, N, &m );

    //printf("c = %d\n", c);

```

```
printf("\n");
```

```
for ( int i = 0; i < m; i++) // ВЫВОД
```

```
{
```

```
printf("res[%d] = %f\n", i, res[i]);
```

```
}
```

```
return 0;
```

```
}
```

2.4. Зробимо опис функції:

Опис функцій

◆ main()

```
int main ( )
```

Функція *main*

Послідовність дій:

- оголошення змінних

Аргументи

mas[] - вхідний масив

res[N] - результуючий масив

r - показчик на кінець максимальної послідовності

l - довжина поточної послідовності

c - кількість послідовностей

m - довжина максимальної послідовності

- шукаємо позитивну послідовність з максимальною сумою
- обробляємо останню послідовність
- копіюємо максимальну послідовність у вихідний масив

Рисунок 1 — Опис функції

2.5 Перевірка правильності роботи програми за допомогою Nemiver:

The image shows the Nemiver IDE interface. The top pane displays the source code of a C program named `main.c`. The code includes a `for` loop for copying data and a `printf` statement for output. The bottom pane shows the 'Локальные переменные' (Local Variables) window, which lists variables and their current values and types.

```
1 {
2     r = p;
3     m = l;
4 }
5
6 q = res;
7
8 for ( p = r - m + 1; p <= r ; p++ )    // копирование максимальной последовательности в выходной массив
9 {
10     *q++ = *p;
11 }
12
13 printf("c = %d\n", c);
14
15 for ( int i = 0; i < m; i++)          // вывод
16 {
17     printf("res[%d] = %f\n", i, res[i]);
18 }
19
20 return 0;
21 }
```

Имя	Переменная	Значение	Тип
Локальные переменные			
	mas	[4]	float [4]
	res	[4]	float [4]
	0	1.10000002	float
	1	1.20000005	float
	2	1.29999995	float
	3	3.0611365e-41	float
	p	0x7fffffffdeec	float *
	r	0x7fffffffdee8	float *
	q	0x7fffffffdefc	float *
	l	1	int
	c	2	int
	m	3	int
	i	1	int

Рисунок 2 — Код у Nemiver.

2.6. Блок-схема моєї програми виглядає наступним чином:

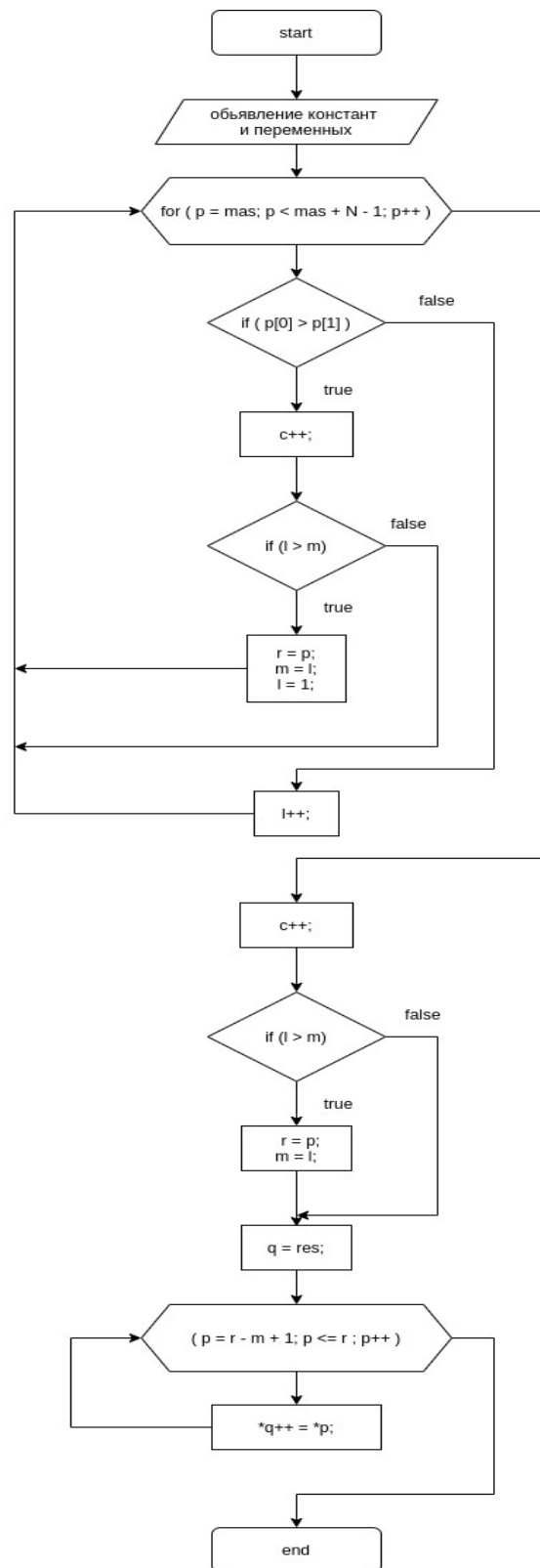


Рисунок 3 — Блок-схема

Висновок:

Я створив програму, яка обчислює безперервну послідовність чисел з не-зменшуваними значеннями.