МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра Вычислительной техники

ОТЧЕТ

по лабораторной работе № 1

по дисциплине «Программирование»

Тема: «Циклические вычисления на языке Си»

Студент гр. 3311 _____ Шарпинский Д. А.

Преподаватель _____ Хахаев И. А.

Санкт-Петербург

2023

Цель работы.

Целью работы является изучение циклических вычислений на языке программирования С и получение практических навыков в их применении.

Задание (вариант 4)

Разработать алгоритм и написать программу, определяющую количество участков последовательности, на которых значения вводимых чисел не уменьшаются, и вычисляющую максимальное количество чисел, образующих такой участок (такие участки не имеют общих элементов). Количество чисел в последовательности задано.

Исходные данные: количество чисел в последовательности, последовательность целых чисел.

Результаты: количество участков последовательности, на которых значения вводимых чисел не уменьшаются, максимальное количество чисел на таком участке.

Постановка задачи и описание решения

Для выполнения данной лабораторной работы необходимо разработать программу, которая будет анализировать последовательность целых чисел и выполнять две задачи:

Определить количество участков в последовательности, на которых значения вводимых чисел не уменьшаются.

Вычислить максимальное количество чисел, образующих такой участок (участки не имеют общих элементов).

Для достижения этой цели необходимо выполнить следующие шаги:

1. Инициализировать переменные: n, digit, globalCounter, counter, maxS, i и last. n будет использоваться для хранения количества чисел в последовательности, digit - для хранения текущего введенного числа, globalCounter - для подсчета количества участков, counter - для подсчета чисел в текущем участке, maxS - для хранения максимального количества чисел в участке, и last - для хранения последнего введенного числа.

2. Запросить у пользователя количество чисел в последовательности и сохранить это значение в переменной n.

3. Запросить у пользователя первое число в последовательности и сохранить его в переменной last. Уменьшить n на 1, так как одно число уже введено.

4. Создать цикл, который будет выполняться n раз (равно количеству оставшихся чисел в последовательности).

5. Внутри цикла запрашивать у пользователя следующее число и сохранять его в переменной digit.

6. Проверить, если digit меньше чем last, то это означает, что текущий участок завершился. В этом случае проверяем, если counter больше 1 (то есть участок содержал более одного числа), то увеличиваем globalCounter на 1 (потому что нашли новый участок) и проверяем, если maxS меньше counter, то обновляем maxS значением counter. Затем сбрасываем counter в 1, так как начинается новый участок.

7. Если digit не меньше last, то увеличиваем counter на 1, так как последовательность чисел продолжается.

8. Обновляем значение last на digit для следующей итерации цикла.

9. После завершения цикла проверяем, если counter больше 1 (чтобы учесть последний участок в последовательности), то выполняем те же действия, что и в шаге 6.

10. Выводим результаты: количество найденных участков (значение globalCounter) и максимальное количество чисел в участке (значение maxS).

Этот алгоритм позволит анализировать последовательность чисел и находить интересующие участки, учитывая условия задачи.

Описание переменных

№	Имя переменной	Тип	Назначение
1	n	int	Количество чисел в последовательности
2	digit	int	Текущее введенное число
3	globalCounter	int	Счетчик количества участков
4	counter	int	Счетчик чисел в текущем участке
5	maxS	int	Максимальное количество чисел в участке
6	last	int	Последнее введенное число
7	i	int	Переменная в цикле

Контрольные примеры

Пример 1:

Исходные данные: n = 10; Последовательность: 1, 1, 1, 0, -10, 2, 3, 4, 5

Результаты: globalCounter = 2; maxS = 6

Пример 2:

Исходные данные: n =5; Последовательность: 5, 4, 3, 2, 1

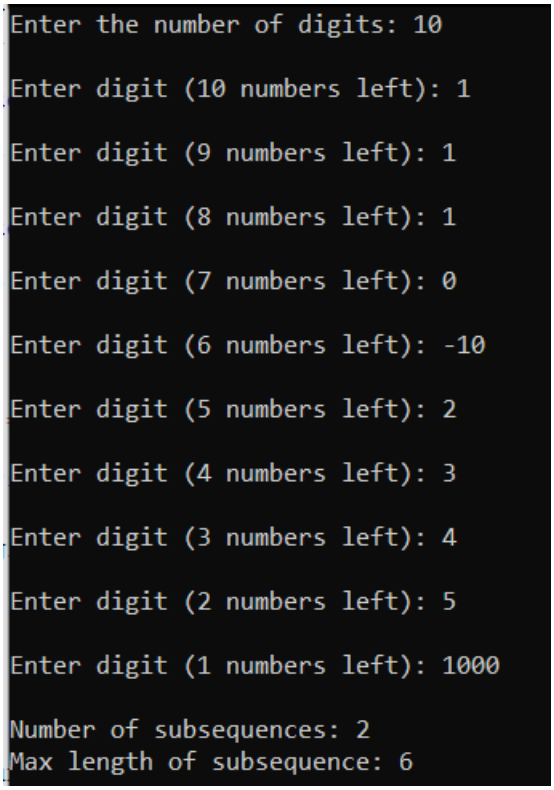
Результаты: globalCounter = 0; maxS = 0

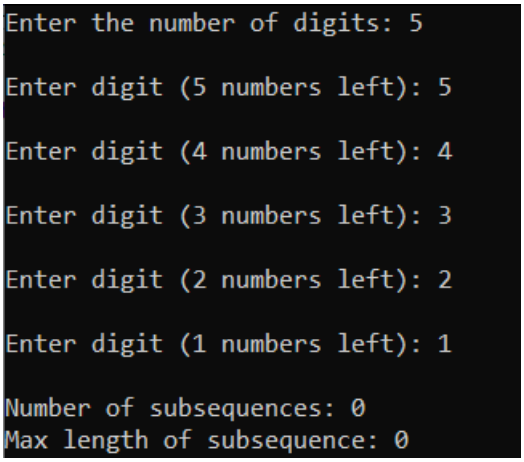
Пример 3:

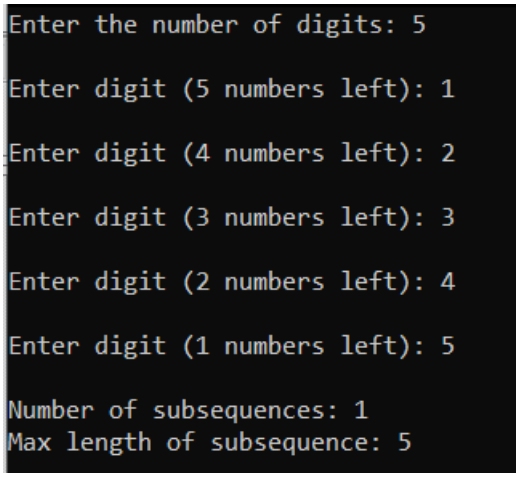
Исходные данные: n =5; Последовательность: 1, 2, 3, 4, 5

Результаты: globalCounter = 1; maxS = 5

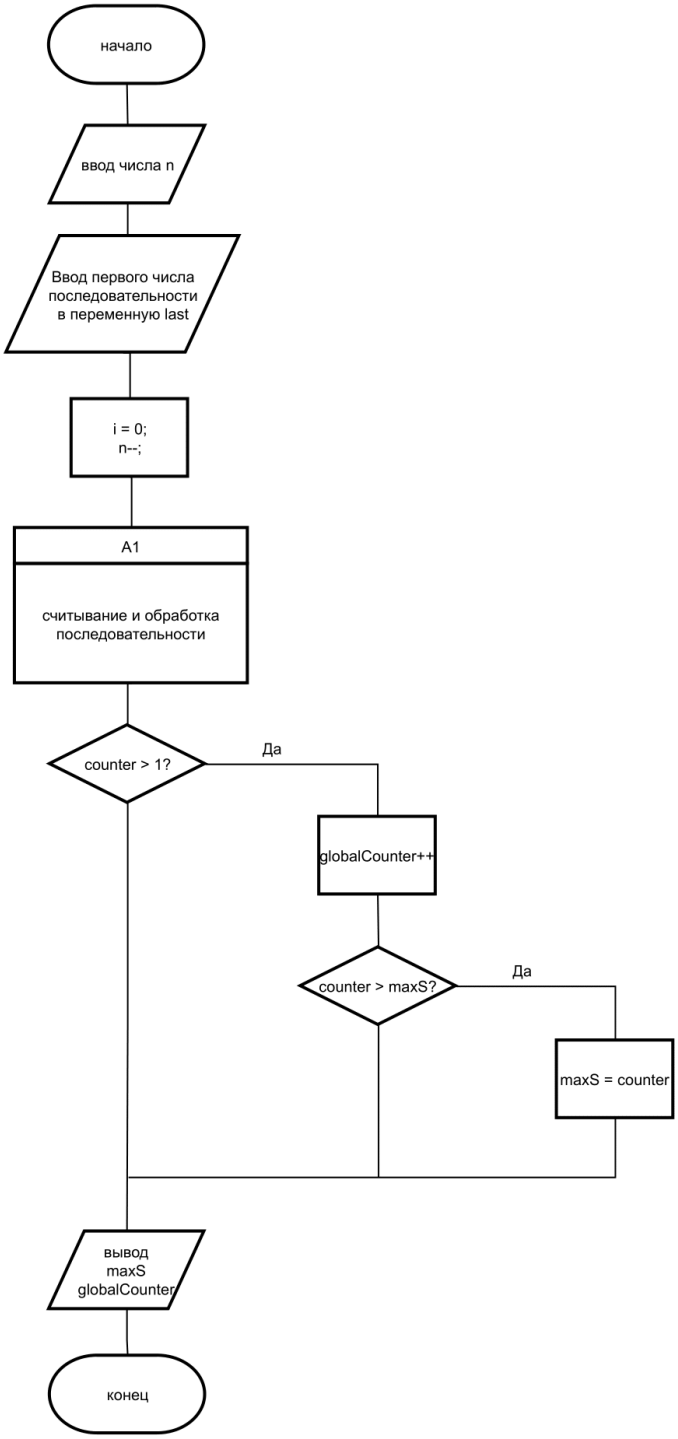
Примеры выполнения программы

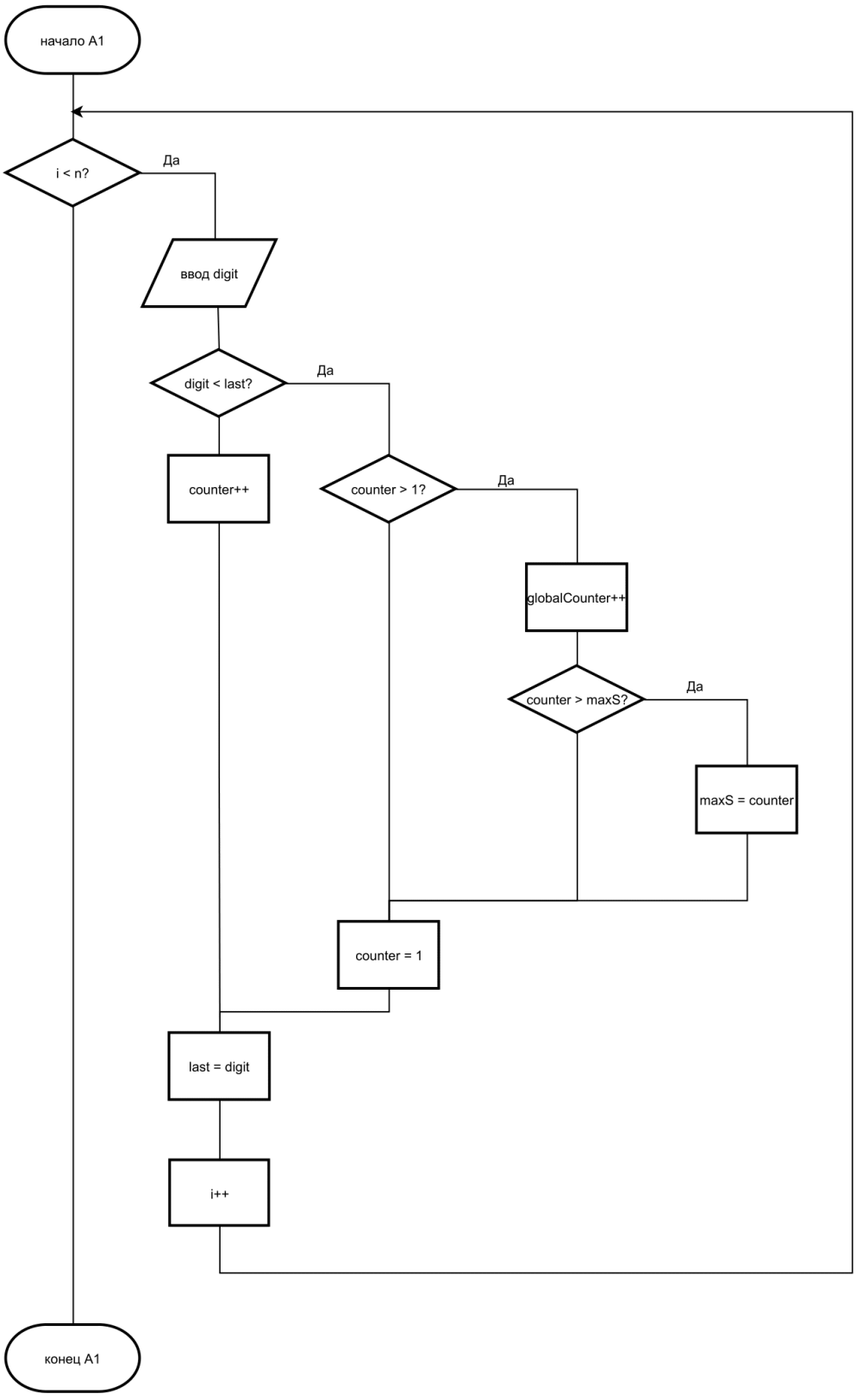






```c
#include<stdio.h>

int main() {

    int n, digit, globalCounter, counter, maxS, last, i;
    globalCounter = maxS = 0;
    counter = 1;

    printf("Enter the number of digits: ");
    scanf("%d", &n);
    printf("\n");

    printf("Enter digit (%d numbers left): ", n);
    scanf("%d", &last);
    printf("\n");

    n--;

    for (i=0; i < n; i++) {
        printf("Enter digit (%d numbers left): ", n-i);
        scanf("%d", &digit);
        printf("\n");
        if (digit < last) {
            if (counter > 1) {
                globalCounter++;
                if (maxS < counter) {
                    maxS = counter;
                }
            }
            counter = 1;
        } else {
            counter++;
        }
        last = digit;
    }

    if (counter > 1) {
        globalCounter++;
        if (maxS < counter) {
            maxS = counter;
        }
    }

    printf("Number of subsequences: %d", globalCounter);
    printf("\n");
    printf("Max length of subsequence: %d", maxS);
    printf("\n");


    return 0;
}

```


начало
ввод числа n
Ввод первого числа последовательности в переменную last
i = 0;
n--;
A1
считывание и обработка последовательности
counter > 1?
Да
globalCounter++
counter > maxS?
Да
maxS = counter
вывод maxS globalCounter
конец


начало A1
i < n?
Да
ввод digit
digit < last?
Да
counter++
counter > 1?
Да
globalCounter++
counter > maxS?
Да
maxS = counter
counter = 1
last = digit
i++
конец A1

Выводы.

В результате выполнения работы изучены циклические вычисления на языке программирования С и получены практические навыки в их применении.