

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ

**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ  
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В. Г. ШУХОВА»  
(БГТУ им. В.Г. Шухова)**

Кафедра программного обеспечения вычислительной техники и автоматизированных систем

## **Лабораторная работа №4**

по дисциплине: Основы программирования  
тема: «Циклы. Введение в тестирование»

Выполнил: ст. группы ПВ-223  
Пахомов Владислав Андреевич

Проверили:  
Притчин Иван Сергеевич  
Черников Сергей Викторович

Код-ревьюер: ст. группы ПВ-223  
Голуцкий Георгий Юрьевич

Белгород 2022 г.

# Лабораторная работа № 4

## Содержание отчёта:

- Тема лабораторной работы
- Цель лабораторной работы
- Решения задач
  - Условие задачи
  - Для задач с звездочкой приложить блок-схему
  - Задачи с двумя звездочками допускается пропустить.
  - Тестовые данные, на которых проверялось приложение
  - Исходный код
- Код-ревью
- Работа над ошибками (код-ревью)
- Вывод

**Тема лабораторной работы:** Циклы. Введение в тестирование

**Цель лабораторной работы:** получение навыков написания циклических алгоритмов и проведения ручного тестирования.

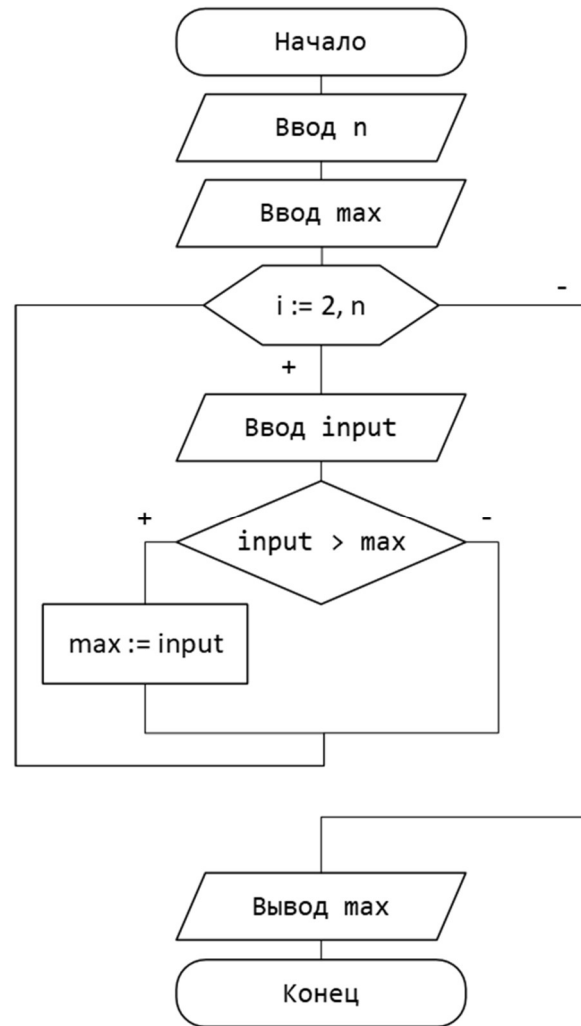
## Решения задач:

### 1. Задача №1

\* С клавиатуры вводятся  $n$  ( $n > 0$ ) чисел. Найти максимальное значение

Входные данные	Ожидаемый результат	Пояснение
1	1	Последовательность из одного элемента, который сам по себе является максимумом
-12312	-12312	Последовательность из одного отрицательного элемента, который сам по себе является максимумом
1 2	2	Максимум обновляется в процессе его поиска, последовательность содержит только положительные числа
-12 -7	-7	Максимум обновляется в процессе его поиска, последовательность содержит только отрицательные числа
-12 7	7	Максимум обновляется в процессе его поиска, последовательность содержит отрицательные и положительные числа
3 2 4 5 4 5	5	Максимум обновляется в процессе его поиска, последовательность содержит только положительные числа
-12 -9 -33 -5 -1 -1	-1	Максимум обновляется в процессе его поиска, последовательность содержит только отрицательные числа
-12 9 -33 0 11	11	Максимум обновляется в процессе его поиска, последовательность содержит отрицательные и положительные числа

199 9 -33 1 11 199	Максимум содержится в начале последовательности
3 9 -33 1 199 199	Максимум содержится в конце последовательности



```

#include <stdio.h>

int main() {
    int n;
    scanf("%d", &n);

    int max;
    scanf("%d", &max);

    for (int i = 2; i <= n; i++) {
        int input;
        scanf("%d", &input);

        max = input > max ? input : max;
    }

    printf("%d", max);

    return 0;
}

```

Ссылка на репозиторий: <https://github.com/IAmProgrammist/programming-and-algorithmization-basics/blob/c/lab4/1.c>

## 2. Задача №2

С клавиатуры вводится последовательность чисел. Признак конца ввода - 0. Найдите максимальное значение среди введенных. Если последовательность была пуста, выведете сообщение 'Последовательность пуста'.

Входные данные	Ожидаемый результат	Пояснение
0	Последовательность пуста	Последовательность не содержащая элементов
1 0	1	Последовательность из одного элемента, который сам по себе является максимумом
-12312 0	-12312	Последовательность из одного отрицательного элемента, который сам по себе является максимумом
1 2 0	2	Максимум обновляется в процессе его поиска, только отрицательные числа
-12 7 0	7	Максимум обновляется в процессе его поиска, отрицательные и положительные числа
-12 -7 0	-7	Максимум обновляется в процессе его поиска, только отрицательные числа
3 2 4 5 4 5 0	5	Максимум обновляется несколько раз, последовательность содержит только положительные числа
-12 -9 -33 -5 -1 0	-1	Максимум обновляется несколько раз, последовательность содержит только отрицательные числа
-12 9 -33 1 11 0	11	Максимум обновляется несколько раз, последовательность содержит отрицательные и положительные числа
199 9 -33 1 11 0	199	Максимум содержится в начале последовательности
3 9 -33 1 11 199 0	199	Максимум содержится в конце последовательности

```
#include <stdio.h>
#include <windows.h>

int main() {
    SetConsoleOutputCP(CP_UTF8);

    int input;
    scanf("%d", &input);

    if (input == 0)
        printf("Последовательность пуста");
    else {
        int maxNumber = input;

        scanf("%d", &input);

        while (input != 0) {
            maxNumber = input > maxNumber ? input : maxNumber;

            scanf("%d", &input);
        }

        printf("%d", maxNumber);
    }

    return 0;
}
```

Ссылка на репозиторий: <https://github.com/IAmProgrammist/programming-and-algorithmization-basics/blob/c/lab4/2.c>

### 3. Задача №3

С клавиатуры вводятся  $n$  ( $n > 0$ ) чисел. Найти индекс первого минимального значения. Нумерация элементов - с нуля.

Входные данные	Ожидаемый результат	Пояснение
1	0	Последовательность из одного элемента, который сам по себе является минимумом
-12312	0	Последовательность из одного отрицательного элемента, который сам по себе является минимумом
2 1	1	Минимум обновляется в процессе его поиска
-7 -12	1	Минимум обновляется в процессе его поиска, последовательность содержит только отрицательные числа
7 -12	1	Минимум обновляется в процессе его поиска, последовательность содержит отрицательные и положительные числа
5 3 4 5 2 5	4	Минимум обновляется несколько раз последовательность содержит только положительные числа
-12 -9 -31 -33 -1	3	Минимум обновляется несколько раз, последовательность содержит только отрицательные числа
100 -33 12 -111 0	3	Минимум обновляется несколько раз, последовательность содержит отрицательные и положительные числа
-199 9 -33 1 11	0	Минимум содержится в начале последовательности
3 9 -33 1 -199	4	Минимум содержится в конце последовательности
12 -5 -5 5 7 -5 15	1	Минимальное число повторяется несколько раз в последовательности

```
#include <stdio.h>

int main() {
    int n;
    scanf("%d", &n);

    int min;
    scanf("%d", &min);

    int firstMinIndex = 0;

    for (int i = 1; i < n; i++) {
        int input;
        scanf("%d", &input);

        if (input < min) {
            min = input;
            firstMinIndex = i;
        }
    }

    printf("%d", firstMinIndex);

    return 0;
}
```

Ссылка на репозиторий: <https://github.com/IAmProgrammist/programming-and-algorithmization-basics/blob/c/lab4/3.c>

#### 4. Задача №4

С клавиатуры вводятся  $n$  ( $n > 0$ ) чисел. Найти индекс последнего максимального значения. Нумерация элементов - с нуля.

Входные данные	Ожидаемый результат	Пояснение
1	0	Последовательность из одного элемента, который сам по себе является максимумом
-12312	0	Последовательность из одного отрицательного элемента, который сам по себе является максимумом
1 2	1	Максимум обновляется в процессе его поиска
-12 -7	1	Максимум обновляется в процессе его поиска, последовательность содержит только отрицательные числа
-12 7	1	Максимум обновляется в процессе его поиска, последовательность содержит отрицательные и положительные числа
1 3 4 4 2 5	5	Максимум обновляется несколько раз
-12 -9 -31 -33 -1	4	Максимум обновляется несколько раз, последовательность содержит только отрицательные числа
-111 -112 111 -111 0	2	Максимум обновляется несколько раз, последовательность содержит отрицательные и положительные числа
199 9 -33 1 11	0	Максимум содержится в начале последовательности
3 9 -33 1 199	4	Максимум содержится в конце последовательности
12 25 25 -5 -7 25 15	5	Максимальное число повторяется несколько раз в последовательности



```
#include <stdio.h>

int main() {
    int n;
    scanf("%d", &n);

    int max;
    scanf("%d", &max);

    int lastMaxIndex = 0;

    for (int i = 1; i < n; i++) {
        int input;
        scanf("%d", &input);

        if (input >= max) {
            max = input;
            lastMaxIndex = i;
        }
    }

    printf("%d", lastMaxIndex);

    return 0;
}
```

Ссылка на репозиторий: <https://github.com/IAmProgrammist/programming-and-algorithmization-basics/blob/c/lab4/4.c>

## 5. Задача №5

С клавиатуры вводятся  $n$  ( $n > 0$ ) чисел. Найти количество минимальных значений.

Входные данные	Ожидаемый результат	Пояснение
1	1	Последовательность из одного элемента, который сам по себе является минимумом
-12312	1	Последовательность из одного отрицательного элемента, который сам по себе является минимумом
2 1	1	Минимум обновляется в процессе его поиска
-12 -70	1	Минимум обновляется в процессе его поиска, последовательность содержит только отрицательные числа
12 -7	1	Минимум обновляется в процессе его поиска, последовательность содержит отрицательные и положительные числа
6 3 4 5 2 5	1	Минимум обновляется несколько раз
-12 -9 -31 -33 -1	1	Минимум обновляется несколько раз, последовательность содержит только отрицательные числа
100 -33 12 -111 0	1	Минимум обновляется несколько раз, последовательность содержит отрицательные и положительные числа
12 -25 -25 -5 -7 -25 15	3	Максимум содержится в начале последовательности

```
#include <stdio.h>

int main() {
    int n;
    scanf("%d", &n);

    int min;
    scanf("%d", &min);

    int minCounter = 1;

    for (int i = 1; i < n; i++) {
        int input;
        scanf("%d", &input);

        if (input < min) {
            min = input;
            minCounter = 1;
        } else if (input == min)
            minCounter++;
    }

    printf("%d", minCounter);

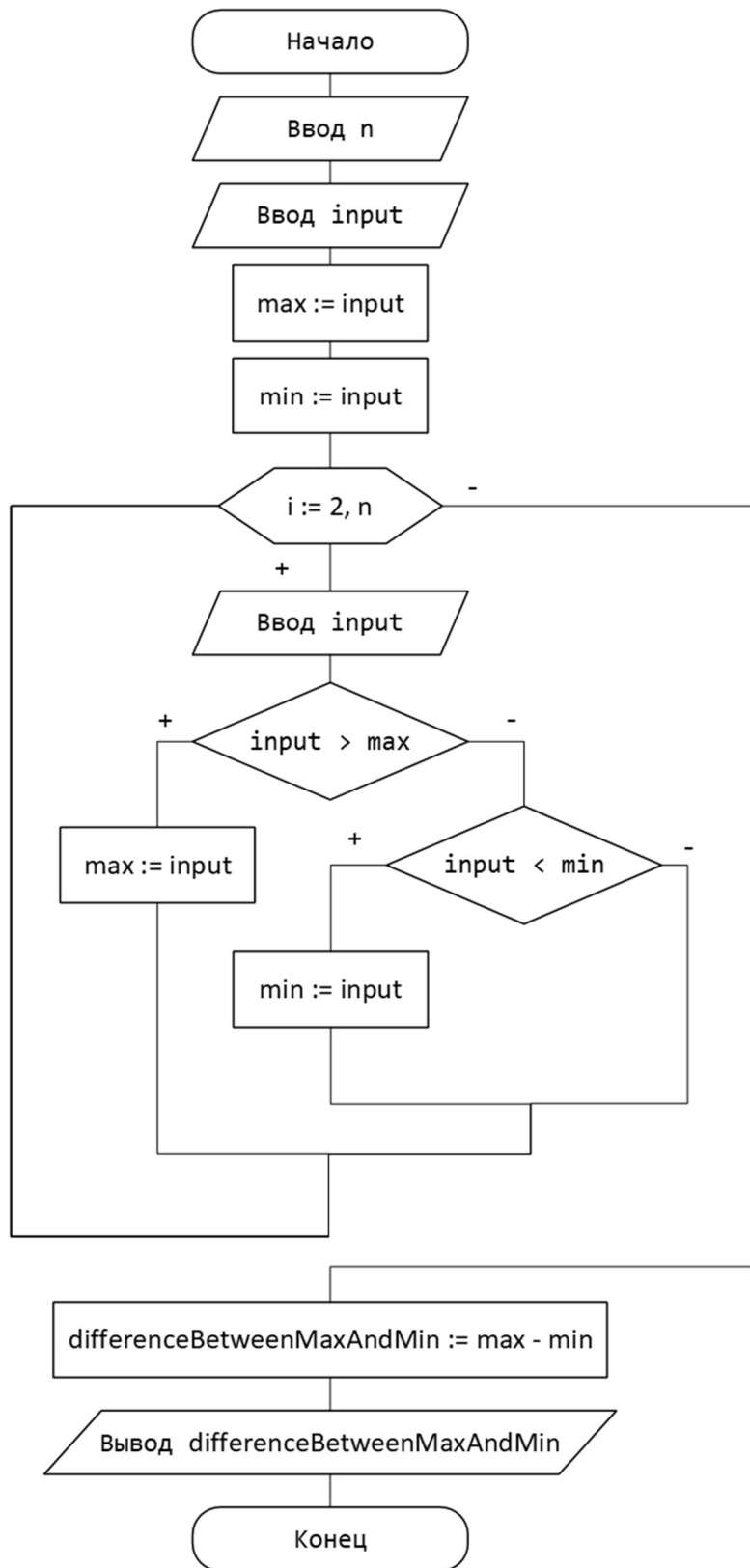
    return 0;
}
```

Ссылка на репозиторий: <https://github.com/IAmProgrammist/programming-and-algorithmization-basics/blob/c/lab4/5.c>

## 6. Задача №6

\* С клавиатуры вводятся  $n$  ( $n > 0$ ) чисел. Найти разность между максимальным и минимальным значением.

Входные данные	Ожидаемый результат	Пояснение
1	0	Последовательность из одного элемента, который сам по себе является минимумом и максимумом
-12312	0	Последовательность из одного отрицательного элемента, который сам по себе является минимумом и максимумом
-1 20	21	Максимум обновляется в процессе его поиска
-1 -20	19	Минимум обновляется в процессе его поиска
5 4 3 4 2 1	4	Минимум обновляется несколько раз
-5 -2 -2 6 7 5 8	13	Максимум обновляется несколько раз
5 4 6 -3 -11 8	19	Максимум и минимум обновляется несколько раз



```
#include <stdio.h>

int main() {
    int n;
    scanf("%d", &n);

    int input;
    scanf("%d", &input);

    int max = input;
    int min = input;

    for (int i = 1; i < n; i++) {
        scanf("%d", &input);

        if (input > max)
            max = input;
        else if (input < min)
            min = input;
    }

    int differenceBetweenMaxAndMin = max - min;

    printf("%d", differenceBetweenMaxAndMin);

    return 0;
}
```

Ссылка на репозиторий: <https://github.com/IAmProgrammist/programming-and-algorithmization-basics/blob/c/lab4/6.c>

## 7. Задача №7

С клавиатуры вводится последовательность. Признак конца ввода - 0. Найти сумму четных чисел.

Входные данные	Ожидаемый результат	Пояснение
0	0	Последовательность не содержащая элементов
1 0	0	Последовательность из одного элемента, который является нечётным
-1 0	0	Последовательность из одного отрицательного элемента, который является нечётным
1 3 7 -1 -7 -3 -1999 0	0	Последовательность из положительных и отрицательных элементов, содержащая только нечётные элементы
6 0	6	Последовательность из одного элемента, который является чётным
-12 0	-12	Последовательность из одного отрицательного элемента, который является чётным
2 6 -8 88 4 -1000 92 0	-816	Последовательность из положительных и отрицательных элементов, содержащая только чётные элементы
-12 9 4 6 77 82 -7 2 1200000 50050005 0	1200082	Последовательность из положительных и отрицательных элементов, содержащая как чётные так и нечётные элементы

```
#include <stdio.h>

int main() {
    int input;
    scanf("%d", &input);

    int evenSum = 0;

    while (input != 0) {
        if (input % 2 == 0)
            evenSum += input;

        scanf("%d", &input);
    }

    printf("%d", evenSum);

    return 0;
}
```

Ссылка на репозиторий: <https://github.com/IAmProgrammist/programming-and-algorithmization-basics/blob/c/lab4/7.c>

## 8. Задача №8

Дано целое число  $n$  ( $n > 0$ ). Найти максимальную цифру в записи этого числа.

Входные данные	Ожидаемый результат	Пояснение
7	7	Число из одной цифры, которое и будет являться максимальной цифрой
180539710	9	Число, состоящее из нескольких цифр
1111111111	1	Число, состоящее из одного типа цифр
11111118	8	Максимальная цифра находится в конце числа
700000	7	Максимальная цифра находится в начале числа
98765423210987654232109876 54232109876542321098765423 21098765423210987654232109 87654232109876542321098765 42321098765423210987654232 10987654232109876542321098 76542321098765423210	9	Очень большое число

```
#include <stdio.h>

#define END_OF_LINE '\n'

int convertCharToDigit(int character) {
    return character - '0';
}

int main() {
    int character = getchar();

    int maxCharacter = 1;

    while (character != END_OF_LINE) {
        maxCharacter = character > maxCharacter ? character : maxCharacter;

        character = getchar();
    }

    int maxDigit = convertCharToDigit(maxCharacter);

    printf("%d", maxDigit);

    return 0;
}
```

Ссылка на репозиторий: <https://github.com/IAmProgrammist/programming-and-algorithmization-basics/blob/c/lab4/8.c>



## 9. Задача №9

Вводится последовательность из натуральных чисел. Признак конца ввода 0. Вывести количество четных и нечетных чисел.

Входные данные	Ожидаемый результат	Пояснение
0	0 0	Последовательность не содержащая элементов
1 0	0 1	Последовательность из одного элемента, который является нечётным
-1 0	0 1	Последовательность из одного отрицательного элемента, который является нечётным
1 3 7 -1 -7 -3 -1999 0	0 7	Последовательность из положительных и отрицательных элементов, содержащая только нечётные элементы
6 0	1 0	Последовательность из одного элемента, который является чётным
-12 0	1 0	Последовательность из одного отрицательного элемента, который является чётным
2 6 -8 88 4 -1000 92 0	7 0	Последовательность из положительных и отрицательных элементов, содержащая только чётные элементы
-12 9 4 6 77 82 -7 2 1200000 50050005 0	6 4	Последовательность из положительных и отрицательных элементов, содержащая как чётные так и нечётные элементы

```
#include <stdio.h>

int main() {
    int input;
    scanf("%d", &input);

    int evenCount = 0;
    int oddCount = 0;

    while (input != 0) {
        if (input % 2 == 0)
            evenCount++;
        else
            oddCount++;

        scanf("%d", &input);
    }

    printf("%d %d", evenCount, oddCount);

    return 0;
}
```

Ссылка на репозиторий: <https://github.com/IAmProgrammist/programming-and-algorithmization-basics/blob/c/lab4/9.c>

## 10. Задача №10

Дано целое число  $n$  ( $n > 0$ ). Найти произведение отличных от нуля цифр данного числа.

Входные данные	Ожидаемый результат	Пояснение
7	7	Число из одной цифры, которое и будет являться ответом
1805347109	30240	Число, состоящее из нескольких цифр, включая ноль
222222	64	Число, состоящее из одного типа цифр
80000000	8	Число из нулей и одной цифры, не равной нулю
11111111111111111111111111111111 11111111111111111111111111111111 11111111111111111111111111111111 11111111111111111111111111111111 11111111111111111111111111111111	1	Очень большое число

11

```
#include <stdio.h>

#define END_OF_LINE '\n'

int convertCharToDigit(int character) {
    return character - '0';
}

int main() {
    int input = getchar();

    int digitsProduct = 1;

    while (input != END_OF_LINE) {
        int digit = convertCharToDigit(input);

        if (digit != 0)
            digitsProduct *= digit;

        input = getchar();
    }

    printf("%d", digitsProduct);

    return 0;
}
```

Ссылка на репозиторий: <https://github.com/IAmProgrammist/programming-and-algorithmization-basics/blob/c/lab4/10.c>

## 11.Задача №11

Дано целое число  $n$  ( $n > 0$ ). Проверить, входит ли в запись числа  $n$  данная цифра  $digit$  ровно  $k$  раз

Входные данные	Ожидаемый результат	Пояснение
$n = 5$ $digit = 5$ $k = 1$	YES	Число из одной цифры, необходимая цифра и является числом, количество цифр равно одному
$n = 3$ $digit = 5$ $k = 1$	NO	Число из одной цифры, необходимая цифра не является числом, количество цифр равно одному
$n = 5$ $digit = 5$ $k = 120$	NO	Необходимое количество цифр превышает общее количество цифр числа
$n = 14545916$ $digit = 5$ $k = 2$	YES	Необходимая цифра встречается ровно требуемое количество раз
$n = 125786125$ $digit = 4$ $k = 1$	NO	Необходимой цифры нет в числе
$n = 19745674694$ $digit = 4$ $k = 2$	NO	Необходимых цифр в числе больше чем необходимое количество на 1
$n = 19745674694$ $digit = 4$ $k = 4$	NO	Необходимых цифр в числе меньше чем необходимое количество на 1
$n = 19745674694$ $digit = 6$ $k = 5$	NO	Необходимых цифр в числе больше чем необходимое количество
$n = 19745674694$ $digit = 9$ $k = 1$	NO	Необходимых цифр в числе меньше чем необходимое количество

```
#include <stdio.h>

int main() {
    int number, digit, requiredDigitCount;
    scanf("%d %d %d", &number, &digit, &requiredDigitCount);

    int actualDigitCount = 0;

    while (number != 0) {
        if (number % 10 == digit)
            actualDigitCount++;

        number /= 10;
    }

    if (actualDigitCount == requiredDigitCount)
        printf("YES");
    else
        printf("NO");

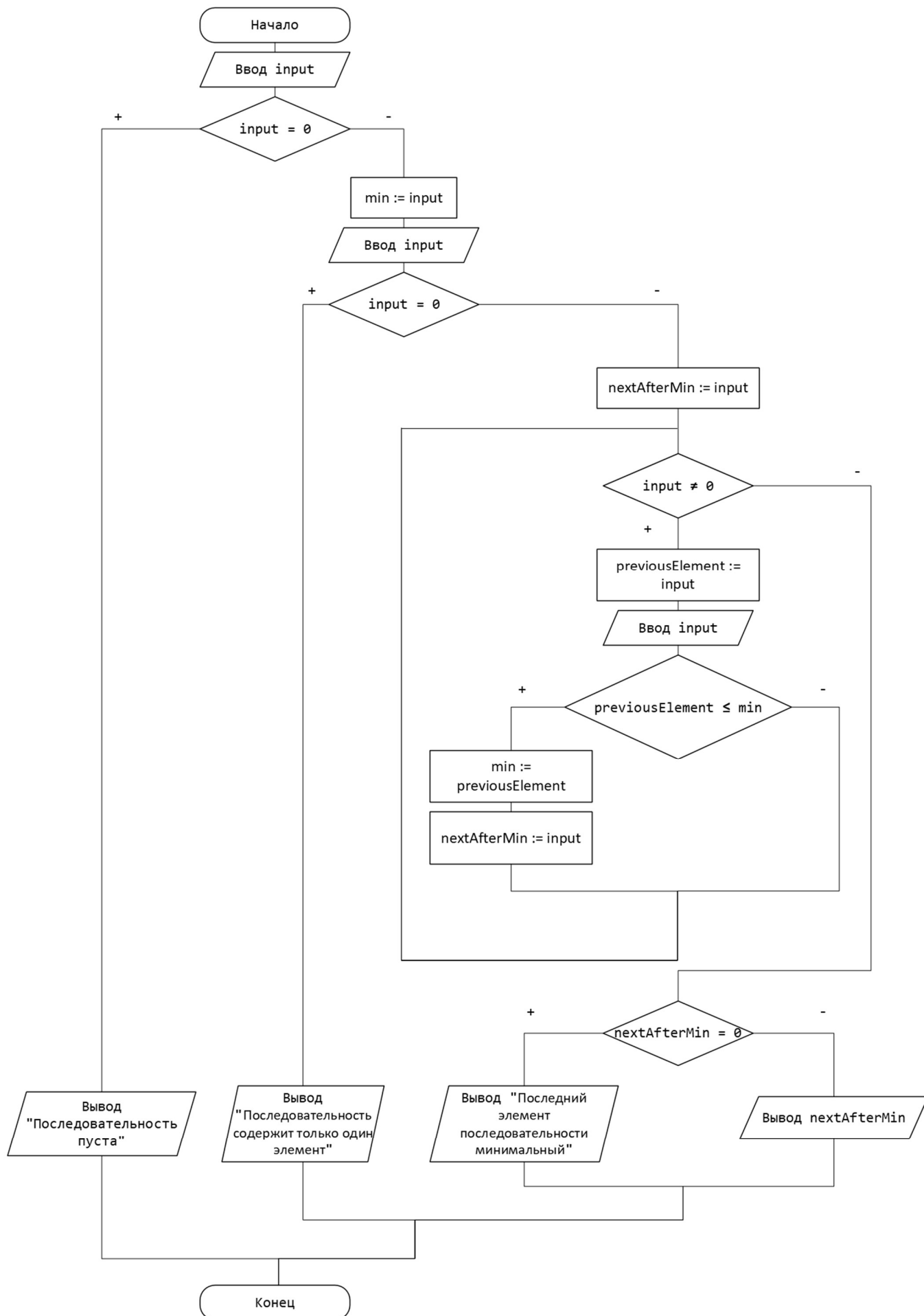
    return 0;
}
```

Ссылка на репозиторий: <https://github.com/IAmProgrammist/programming-and-algorithmization-basics/blob/c/lab4/11.c>

## 12.Задача №12

\* С клавиатуры вводятся целые числа. Признак конца ввода – ноль. Определить число, следующее за последним из введенных минимальных значений.

Входные данные	Ожидаемый результат	Пояснение
0	Последовательность пуста	Последовательность не содержащая элементов
3 0	Последовательность содержит только один элемент	Единственный элемент, сам по себе и являющийся минимумом. Так как он единственный, то чисел после него не существует
1 2 0	2	Последовательность из двух элементов, первый - минимальный. Минимум не обновляется
76 12 8 55 61 -9 122 -99 78 10 81 0	78	Минимум обновляется в процессе его поиска
-1 5 -1 4 -1 3 -1 2 0	2	В последовательности несколько раз встречается минимум
8743 22 9238 -41 9 77 42 42 42 -100 0	Последний элемент последовательности минимальный	Последний элемент последовательности минимальный
-1 4 -1 3 -1 2 -1 0	Последний элемент последовательности минимальный	В последовательности несколько раз встречается минимум, последний элемент минимальный
-1 -1 -9 -12 -6 0	-6	Последовательность состоит только из отрицательных чисел



```
#include <stdio.h>
#include <windows.h>

int main() {
    SetConsoleOutputCP(CP_UTF8);

    int input;
    scanf("%d", &input);

    if (input == 0)
        printf("Последовательность пуста");
    else {
        int min = input;

        scanf("%d", &input);

        if (input == 0)
            printf("Последовательность содержит только один элемент");
        else {
            int nextAfterMin = input;

            int previousElement;
            while (input != 0) {
                previousElement = input;

                scanf("%d", &input);

                if (previousElement <= min) {
                    min = previousElement;
                    nextAfterMin = input;
                }
            }

            if (nextAfterMin == 0)
                printf("Последний элемент последовательности минимальный");
            else
                printf("%d", nextAfterMin);
        }
    }

    return 0;
}
```

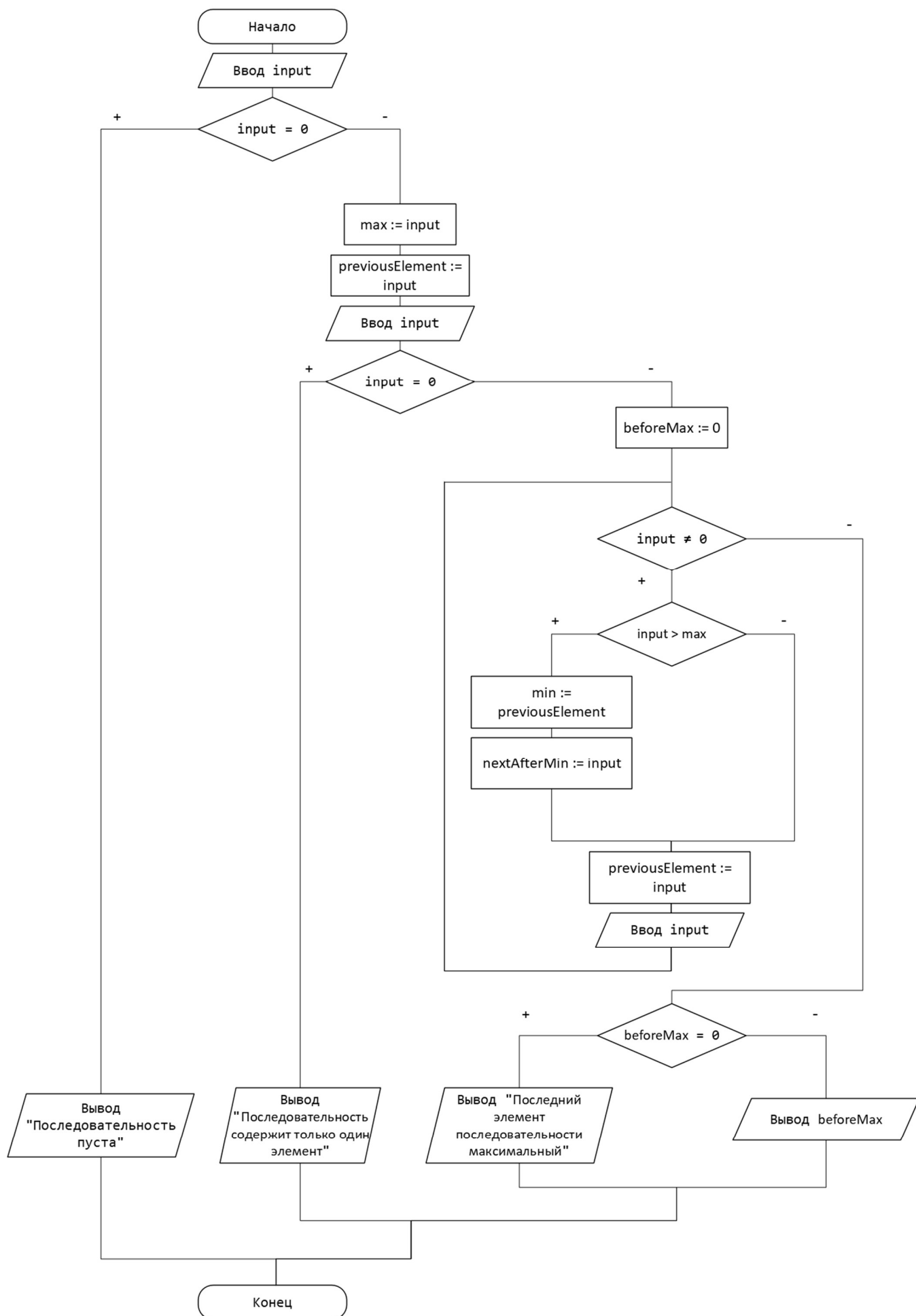
Ссылка на репозиторий: <https://github.com/IAmProgrammist/programming-and-algorithmization-basics/blob/c/lab4/12.c>

### 13.Задача №13

\* С клавиатуры вводятся целые числа. Признак конца ввода – ноль. Определить число, предшествующее первому из введенных максимальных значений.

Входные данные	Ожидаемый результат	Пояснение
0	Последовательность пуста	Последовательность не содержащая элементов
3 0	Последовательность содержит только один элемент	Единственный элемент, сам по себе и являющийся максимумом. Так как он единственный, то чисел после него не существует
1 2 0	2	Последовательность из двух элементов, второй - максимальный. Максимум не обновляется
76 12 8 55 61 -9 122 -99 78 10 81 0	-9	Максимум обновляется в процессе его поиска
4 1 2 5 4 5 3 5 2 5 1 0	2	В последовательности несколько раз встречается одинаковый максимум
8743 22 22 22 3 25 5 42 42 42 -100 0	Первый элемент последовательности максимальный	Первый элемент последовательности максимальный
9 6 9 1 9 9 9 3 2 1 9 7 0	Первый элемент последовательности максимальный	В последовательности несколько раз встречается максимум, первый элемент минимальный
-12 -13 -9 -12 -6 0	-12	Последовательность состоит только из отрицательных чисел





```

#include <stdio.h>
#include <windows.h>

int main() {
    SetConsoleOutputCP(CP_UTF8);

    int input;
    scanf("%d", &input);

    if (input == 0)
        printf("Последовательность пуста");
    else {
        int max = input;
        int previousElement = input;

        scanf("%d", &input);

        if (input == 0)
            printf("Последовательность содержит только один элемент");
        else {
            int beforeMax = 0;

            while (input != 0) {
                if (input > max) {
                    max = input;
                    beforeMax = previousElement;
                }

                previousElement = input;

                scanf("%d", &input);
            }

            if (beforeMax == 0)
                printf("Первый элемент последовательности максимальный");
            else
                printf("%d", beforeMax);
        }
    }

    return 0;
}

```

Ссылка на репозиторий: <https://github.com/IAmProgrammist/programming-and-algorithmization-basics/blob/c/lab4/13.c>

## 14. Задача №14

С клавиатуры вводятся символы. Признак конца ввода – символ перехода на новую строку '\n' (Клавиша *Enter*). Определить количество букв

Входные данные	Ожидаемый результат	Пояснение
	0	Пустая строка
qoeufslfihrwfepoqrfj	20	Строка из гласных и согласных букв в нижнем регистре
aoieuy	6	Все гласные буквы
aaaioiaiao	9	Строка содержит только гласные буквы
qwrtpsdfghjklzxcvbnm	20	Все согласные буквы
,./?<>~`!@#\$\$%^&*()_+- =[{ }]*- +\ 0987654321	0	Специальные символы, цифры
the 42 quick, brown 9921 fox. 0012586 jumps over 12 35 123 12345 - the lazy dog?		Строка, содержащая латинские буквы и спецсимволы

```
#include <stdio.h>
#include <stdbool.h>

#define END_OF_LINE '\n'

bool isLetter(int character) {
    return character >= 'a' && character <= 'z';
}

int main() {
    int count = 0;

    int input;
    while ((input = getchar()) != END_OF_LINE)
        if (isLetter(input))
            count++;

    printf("%d", count);

    return 0;
}
```

Ссылка на репозиторий: <https://github.com/IAmProgrammist/programming-and-algorithmization-basics/blob/c/lab4/14.c>

## 15. Задача №15

С клавиатуры вводятся символы. Признак конца ввода - символ перехода на новую строку '\n'. Определить количество согласных букв.

Входные данные	Ожидаемый результат	Пояснение
	0	Пустая строка
qoeufslfihrwfepoqrfj	14	Строка из гласных и согласных букв в нижнем регистре
aoieuy	0	Все возможные буквы, что не должны считаться согласными
gplkh	5	Строка содержит только согласные буквы
qwrtpsdfghjklzxcvbnm	20	Все возможные буквы, что должны считаться согласными
,./?<>~`!@#\$\$%^&*()_+- =][{*}- +\ 0987654321	0	Специальные символы, цифры
the 42 quick, brown 9921 fox. 0012586 jumps over 12 23 123 12345 - the lazy dog?		Строка, содержащая латинские буквы и спецсимволы

```
#include <stdio.h>
#include <stdbool.h>

#define END_OF_LINE '\n'

bool isLetter(int character) {
    return character >= 'a' && character <= 'z';
}

bool isVowel(int character) {
    return character == 'a' ||
           character == 'e' ||
           character == 'i' ||
           character == 'o' ||
           character == 'u' ||
           character == 'y';
}

bool isConsonant(int character) {
    return isLetter(character) && !isVowel(character);
}

int main() {
    int count = 0;

    int input;
    while ((input = getchar()) != END_OF_LINE)
        if (isConsonant(input))
            count++;

    printf("%d", count);

    return 0;
}
```

Ссылка на репозиторий: <https://github.com/IAmProgrammist/programming-and-algorithmization-basics/blob/c/lab4/15.c>

## 16. Задача №16

С клавиатуры вводятся целые числа. Признак конца ввода – ноль. Определить, является ли вводимая последовательность упорядоченной по невозрастанию или по неубыванию или все элементы равны или последовательность не принадлежит ни к какой из групп.

Входные данные	Ожидаемый результат	Пояснение
0	Последовательность пуста	Последовательность не содержащая элементов
5 0	Тип последовательности определить невозможно	Последовательность из одного элемента
-11 2 0	Возрастающая последовательность	Возрастающая последовательность из двух элементов
99 -2 0	Убывающая последовательность	Убывающая последовательность из двух элементов
42 42 0	Последовательность из одинаковых элементов	Последовательность из двух одинаковых элементов
-12 -3 78 99 100 7712 0	Возрастающая последовательность	Возрастающая последовательность из нескольких элементов
123 89 12 -1 -35 -36 -128 0	Убывающая последовательность	Убывающая последовательность из нескольких элементов
12 12 12 12 12 12 12 0	Последовательность из одинаковых элементов	Последовательность из нескольких одинаковых элементов
-12 -12 -12 -12 -12 -12 -12 0	Последовательность из одинаковых элементов	Последовательность из нескольких одинаковых отрицательных элементов
12 -8 9123 2 2 9 76 -912 -923 2222 0	Тип последовательности определить невозможно	Последовательность, тип которой определить невозможно
1 3 5 8 8 9 10 0	Тип последовательности определить невозможно	Возрастающая последовательность, кроме одного элемента в середине (равен предыдущему)
1 3 5 8 -8 9 10 0	Тип последовательности определить невозможно	Возрастающая последовательность, кроме одного элемента в середине (меньше предыдущего)
1 3 5 8 9 10 10 0	Тип последовательности определить невозможно	Возрастающая последовательность, кроме одного элемента в конце (равен предыдущему)
1 3 5 8 9 10 -10 0	Тип последовательности определить невозможно	Возрастающая последовательность, кроме одного элемента в конце (меньше предыдущего)
3 3 5 8 9 10 0	Тип последовательности определить невозможно	Возрастающая последовательность, кроме одного элемента в начале (равен следующему)
7 3 5 8 9 10 0	Тип последовательности определить невозможно	Возрастающая последовательность, кроме одного элемента в начале (больше следующего)

87 56 4 3 3 -12 0	Тип последовательности определить невозможно	Убывающая последовательность, кроме одного элемента в середине (равен предыдущему)
87 56 4 3 42 -12 0	Тип последовательности определить невозможно	Убывающая последовательность, кроме одного элемента в середине (больше предыдущего)
87 56 4 3 -1 -1 0	Тип последовательности определить невозможно	Убывающая последовательность, кроме одного элемента в конце (равен предыдущему)
87 56 4 3 -1 12 0	Тип последовательности определить невозможно	Убывающая последовательность, кроме одного элемента в конце (больше предыдущего)
56 56 4 3 -1 -12 0	Тип последовательности определить невозможно	Убывающая последовательность, кроме одного элемента в начале (равен следующему)
55 56 4 3 -1 -12 0	Тип последовательности определить невозможно	Последовательность из одинаковых элементов, кроме одного элемента в начале (меньше следующего)
42 42 42 142 42 42 0	Тип последовательности определить невозможно	Последовательность из одинаковых элементов, кроме одного элемента в середине (больше предыдущего)
42 42 42 -842 42 42 0	Тип последовательности определить невозможно	Последовательность из одинаковых элементов, кроме одного элемента в середине (меньше предыдущего)
42 42 42 42 42 412 0	Тип последовательности определить невозможно	Последовательность из одинаковых элементов, кроме одного элемента в конце (равен предыдущему)
42 42 42 42 42 -942 0	Тип последовательности определить невозможно	Последовательность из одинаковых элементов, кроме одного элемента в конце (больше предыдущего)
442 42 42 42 42 42 0	Тип последовательности определить невозможно	Последовательность из одинаковых элементов, кроме одного элемента в начале (равен следующему)
-9 42 42 42 42 42 0	Тип последовательности определить невозможно	Последовательность из одинаковых элементов, кроме одного элемента в начале (меньше следующего)

```
#include <stdio.h>
#include <windows.h>
#include <stdbool.h>

#define DECREASING_SEQUENCE (-1)
#define INCREASING_SEQUENCE 1
#define EQUAL_SEQUENCE 0

int defineSequenceType(int numberA, int numberB) {
    if (numberA > numberB) return INCREASING_SEQUENCE;
    if (numberA < numberB) return DECREASING_SEQUENCE;
    return EQUAL_SEQUENCE;
}
```

```

int main() {
    SetConsoleOutputCP(CP_UTF8);

    int input;
    scanf("%d", &input);

    if (input == 0)
        printf("Последовательность пуста");
    else {
        int previousNumber = input;

        scanf("%d", &input);

        if (input == 0)
            printf("Тип последовательности определить невозможно");
        else {
            int requiredSequenceType = defineSequenceType(input, previousNumber);
            bool isSequenceTypeComplied = true;
            previousNumber = input;

            scanf("%d", &input);

            while (input != 0 && isSequenceTypeComplied) {
                int currentSequenceType = defineSequenceType(input, previousNumber);

                isSequenceTypeComplied = currentSequenceType ==
requiredSequenceType;
                previousNumber = input;

                scanf("%d", &input);
            }

            if (isSequenceTypeComplied)
                switch (requiredSequenceType) {
                    case DECREASING_SEQUENCE:
                        printf("Убывающая последовательность");
                        break;
                    case EQUAL_SEQUENCE:
                        printf("Последовательность из одинаковых элементов");
                        break;
                    case INCREASING_SEQUENCE:
                        printf("Возрастающая последовательность");
                        break;
                }
            else
                printf("Тип последовательности определить невозможно");
        }
    }

    return 0;
}

```

Ссылка на репозиторий: <https://github.com/IAmProgrammist/programming-and-algorithmization-basics/blob/c/lab4/16.c>



## 17. Задача №17

С клавиатуры вводятся символы. Признак конца ввода – точка. Определить сумму введенных цифр.

Входные данные	Ожидаемый результат	Пояснение
.	0	Пустая последовательность символов
ADS98yhr978y 2 c34637r283fr29y 3123uyh8- =9`=9=984847f 4586uf0wc800890 89=----- .	230	Последовательность из случайных символов
123456789.	45	Символы, которые должны расцениваться как цифры
aWUIDhkjaF h ifawIAFW faw*/-*/+.=--=/. .	0	Строка без цифр

```
#include <stdio.h>
#include <stdbool.h>

#define END_OF_INPUT '.'

bool isDigit(int character) {
    return character >= '0' && character <= '9';
}

int convertCharToDigit(int character) {
    return character - '0';
}

int main() {
    int digitsSum = 0;

    int input;
    while ((input = getchar()) != END_OF_INPUT)
        if (isDigit(input))
            digitsSum += convertCharToDigit(input);

    printf("%d", digitsSum);

    return 0;
}
```

Ссылка на репозиторий: <https://github.com/IAmProgrammist/programming-and-algorithmization-basics/blob/c/lab4/17.c>

## 18. Задача №18

\*\* С клавиатуры вводятся символы (пробелы и цифры). Признак конца ввода - точка. Определить сумму введенных чисел.

Входные данные	Ожидаемый результат	Пояснение
.	0	Строка без чисел
.	0	Строка без чисел, только пробелы
5.	5	Строка из одного однозначного числа
5.	5	Строка из одного однозначного числа, содержит пробелы до числа
5 .	5	Строка из одного однозначного числа, содержит пробелы после числа
5 . 5	5	Строка из одного однозначного числа, содержит пробелы до и после числа
55123.	55123	Строка из одного многозначного числа
55123.	55123	Строка из одного многозначного числа, содержит пробелы до числа
55123 .	55123	Строка из одного многозначного числа, содержит пробелы после числа
55123 . 55123	55123	Строка из одного многозначного числа, содержит пробелы после числа
12 87 99 70 12 10 90 142.	522	Строка из нескольких многозначных чисел
12 87 99 70 112 10 90 42 .	522	Строка из нескольких многозначных чисел, между числами количество пробелов >= 1

```
#include <stdio.h>
#include <stdbool.h>

#define END_OF_INPUT '.'

bool isDigit(int character) {
    return character >= '0' && character <= '9';
}

int convertCharToDigit(int character) {
    return character - '0';
}

int main() {
    int numbersSum = 0;
    int tempNum = 0;

    int input;
    do {
        input = getchar();

        if (isDigit(input))
            tempNum = tempNum * 10 + convertCharToDigit(input);
        else {
            numbersSum += tempNum;
            tempNum = 0;
        }
    } while (input != END_OF_INPUT);

    printf("%d", numbersSum);

    return 0;
}
```

Ссылка на репозиторий: <https://github.com/IAmProgrammist/programming-and-algorithmization-basics/blob/c/lab4/18.c>

**Код-ревью:**

- 1) В описании задач отсутствует условие
- 2) Упростить решение задачи №9
- 3) 14 задача содержит баг (программа считает гласные буквы, а не все буквы)
- 4) Форматирование кода: отступы после и перед вводом с консоли

### **Работа над ошибками (код-ревью):**

- 1) Отчёт доработан
- 2) Решение упрощено
- 3) Баг устранён
- 4) Необходимые отступы добавлены, лишние – удалены.

**Вывод:** в ходе работы получены навыки написания циклических алгоритмов, получены навыки проведения ручного тестирования.