



**«Московский государственный технический университет  
имени Н.Э. Баумана»  
(национальный исследовательский университет)  
(МГТУ им. Н.Э. Баумана)**

---

Федеральное государственное бюджетное образовательное учреждение высшего образования

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ  
КАФЕДРА КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)


**О т ч е т**

**по домашнему заданию № 2**

**Название домашней работы:** Обработка одномерных массивов. Обработка строк. Создание псевдомодулей. Процедурный тип параметров.

**Дисциплина:** Алгоритмизация и программирование

Студент гр. ИУ6-15Б

  
(Подпись, дата)

В.А. Бирюков  
(И.О. Фамилия)

Преподаватель

  
(Подпись, дата)

О.А. Веселовская  
(И.О. Фамилия)

## Вариант 2

### Часть 1

**Задание:** сортировать одномерный массив вещественных чисел по невозрастанию чисел. Использовать метод обменов.

Напишем код программы на языке C++

```
#include <iostream>
#include <Windows.h>

int main()
{
    SetConsoleOutputCP(WCodePageID: CP_UTF8);
    int n;
    float temp;
    std::cout << "Введите длину массива: " << std::endl;
    std::cin >> n;
    float arr[n];
    std::cout << "Введите элементы массива: " << std::endl;
    for (int i = 0; i < n; i++)
        std::cin >> arr[i];
    std::cout << "Введённый массив: " << std::endl;
    for (int i = 0; i < n; i++)
        std::cout << arr[i] << " ";

    for (int i = 0; i < n; i++) {
        for (int j = 1; j < n; j++) {
            if (arr[j] > arr[j-1]) {
                temp = arr[j];
                arr[j] = arr[j-1];
                arr[j-1] = temp;
            }
        }
    }

    std::cout << std::endl << "Отсортированный массив: " << std::endl;
    for (int i = 0; i < n; i++)
        std::cout << arr[i] << " ";

    return 0;
}
```

Рисунок 1 – код программы

Запустим программу с разными введёнными данными, чтобы проверить правильность работы программы

```
D:\Лабы\АиП\Dz_2\1\stake-build-debug\1.exe
Введите длину массива:
5
Введите элементы массива:
1.3 2.4 3.1 4.6 6.2
Введённый массив:
1.3 2.4 3.1 4.6 6.2
Отсортированный массив:
6.2 4.6 3.1 2.4 1.3
Process finished with exit code 0
```

Рисунок 2 – первый запуск программы

```
D:\Лабы\АиП\Dz_2\1\stake-build-debug\1.exe
Введите длину массива:
10
Введите элементы массива:
10 9 8 7 6 5 4 3 2 1
Введённый массив:
10 9 8 7 6 5 4 3 2 1
Отсортированный массив:
10 9 8 7 6 5 4 3 2 1
Process finished with exit code 0
```

Рисунок 3 – второй запуск программы

Видим, что программа корректно работает на введённых числах

Составим схему алгоритма этой программы

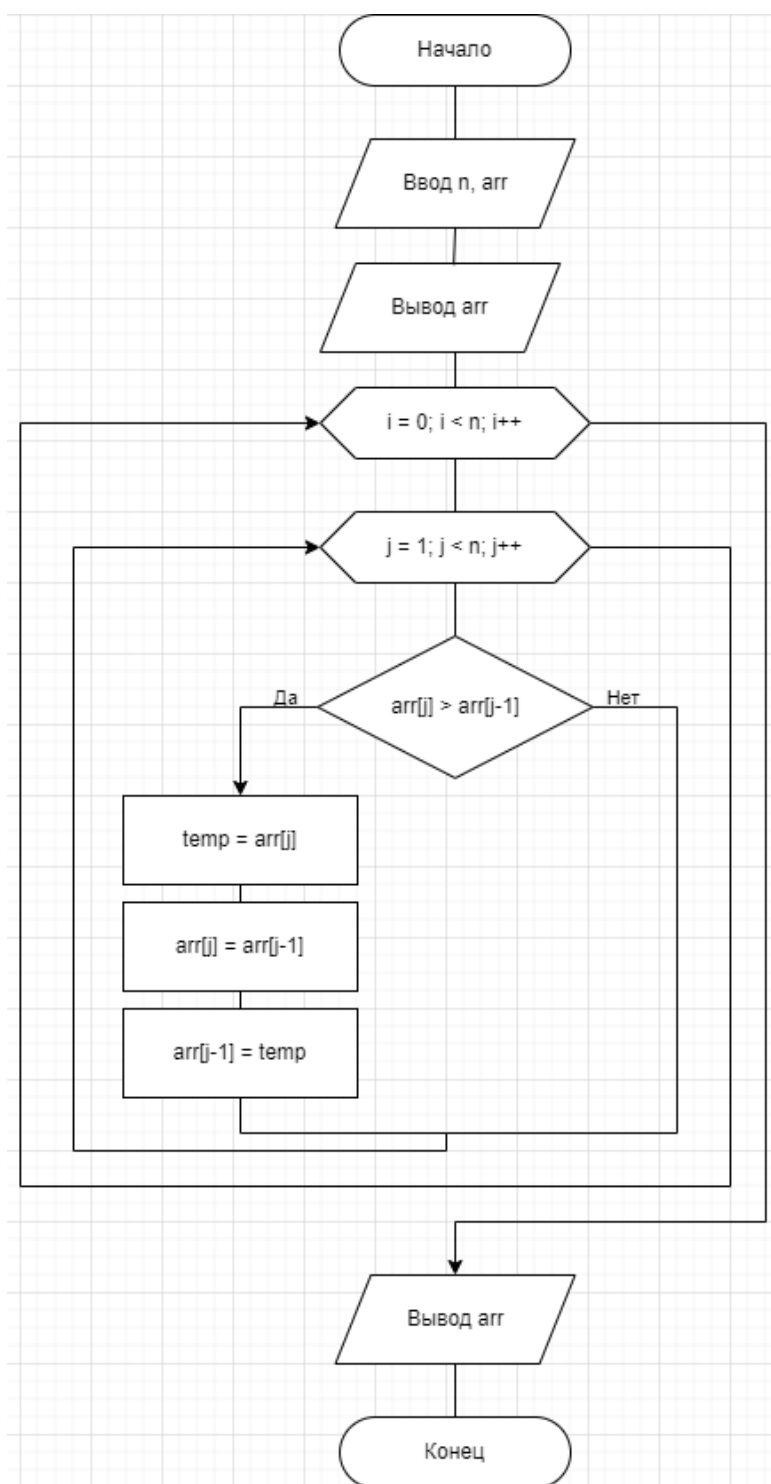


Рисунок 4 – схема алгоритма программы

## Часть 2

**Задание:** Дана непустая последовательность слов из строчных латинских букв (количество слов больше 5): между соседними словами – запятая и пробел, за последним словом – точка. Определить количество слов, которые начинаются на заглавную латинскую букву. Вывести на экран количество слов и сами эти слова.

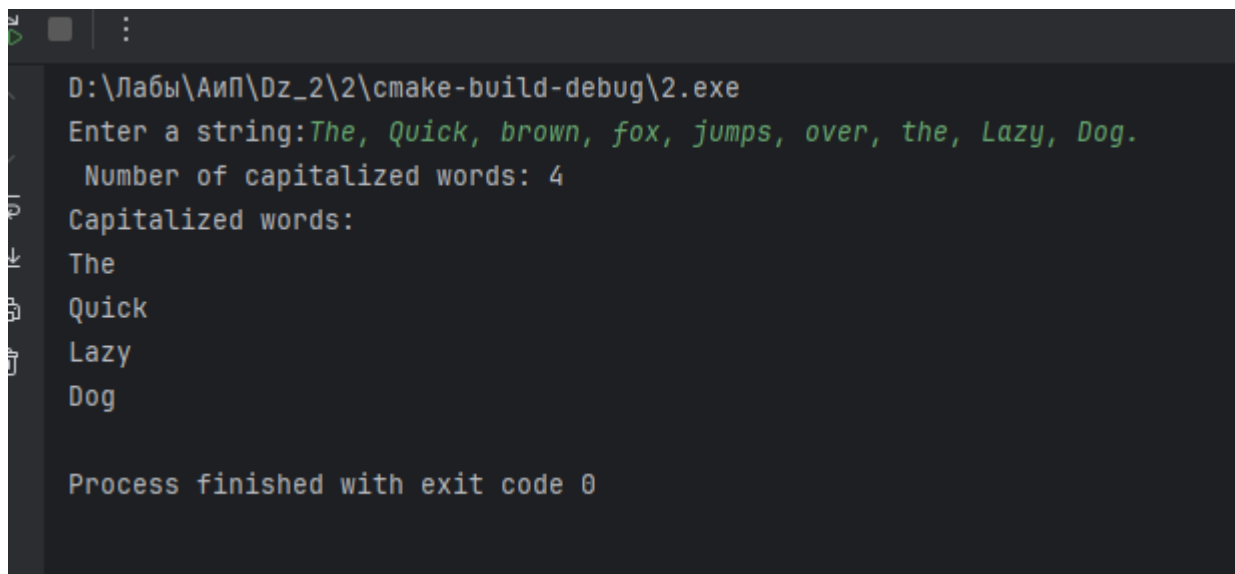
Напишем код программы на языке C++

```
#include <iostream>
#include <string>

int main() {
    std::string str;
    int k = 0;
    std::cout << "Enter a string: ";
    std::getline(&std::cin, &str);
    for (int i = 0; i < str.length(); i++) {
        if (65<=static_cast<int>(str[i]) && static_cast<int>(str[i])<=90) {
            k++;
        }
    }
    std::cout << "Number of capitalized words: " << k << std::endl;
    std::cout << "Capitalized words: " << std::endl;
    int i=0;
    while (i < str.length()) {
        if (65<=static_cast<int>(str[i]) && static_cast<int>(str[i])<=90) {
            while (str[i] != '.' && str[i] != ',') {
                std::cout << str[i];
                i++;
            }
            std::cout << std::endl;
        }
        i++;
    }
    return 0;
}
```

Рисунок 5 – код программы

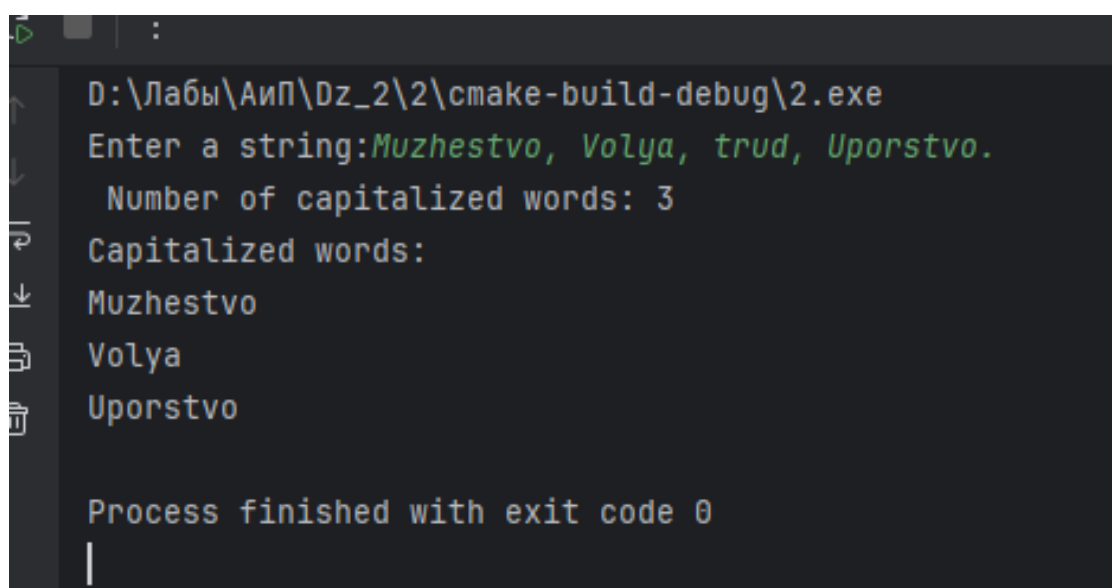
Запустим программу с разными введёнными данными, чтобы убедиться в правильности работы программы



```
D:\Лаб\АиП\Dz_2\2\cmake-build-debug\2.exe
Enter a string:The, Quick, brown, fox, jumps, over, the, Lazy, Dog.
Number of capitalized words: 4
Capitalized words:
The
Quick
Lazy
Dog

Process finished with exit code 0
```

Рисунок 6 – первый запуск программы



```
D:\Лаб\АиП\Dz_2\2\cmake-build-debug\2.exe
Enter a string:Muzhestvo, Volya, trud, Uporstvo.
Number of capitalized words: 3
Capitalized words:
Muzhestvo
Volya
Uporstvo

Process finished with exit code 0
|
```

Рисунок 7 – второй запуск программы

Видим, что программа корректно работает на введённых строках

Составим схему алгоритма для этой программы

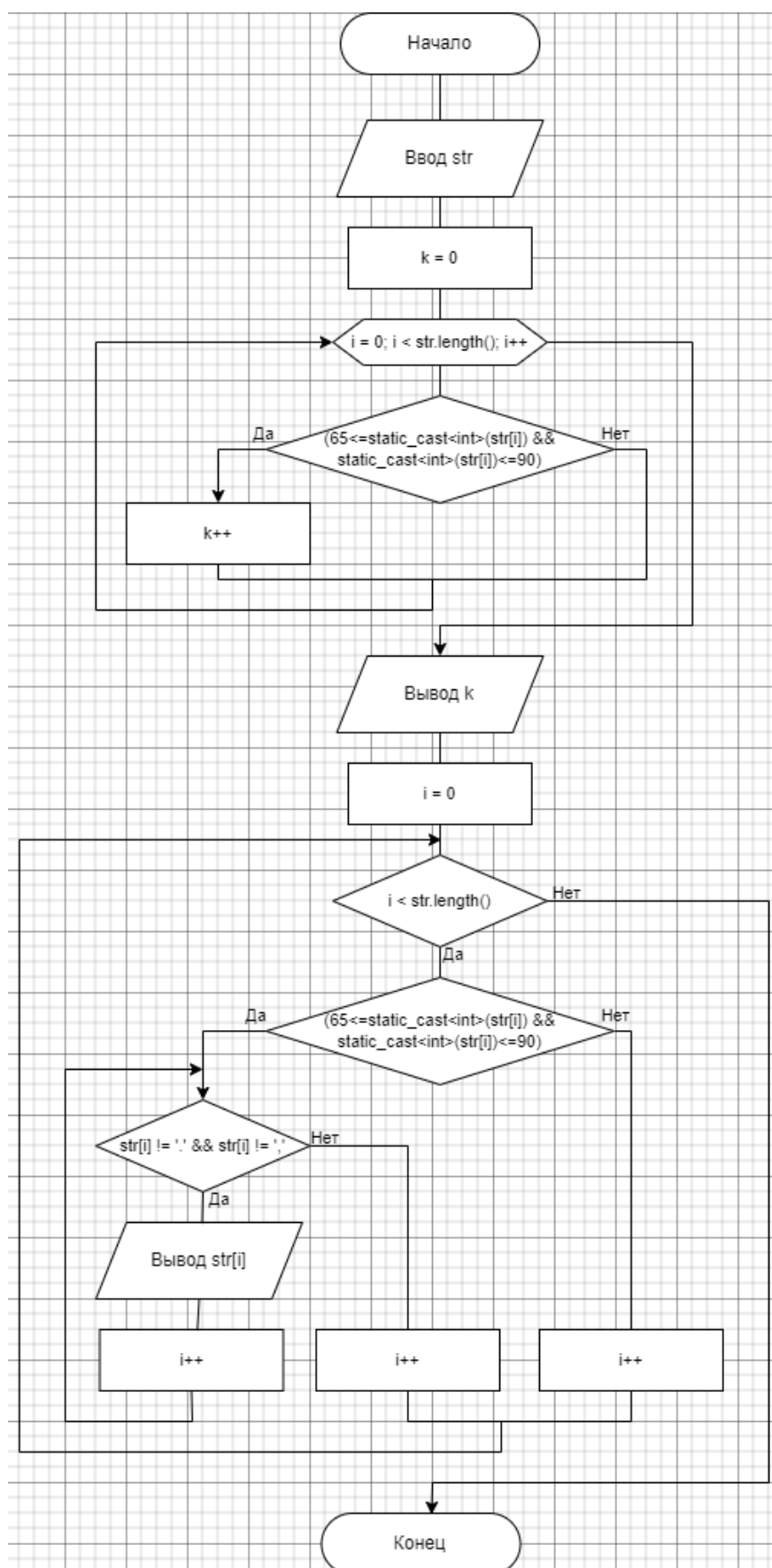


Рисунок 8 – схема алгоритма программы

### Часть 3

**Задача:** Разработать заголовочный файл и файл реализации, содержащие указанную подпрограмму. Написать тестирующую программу..

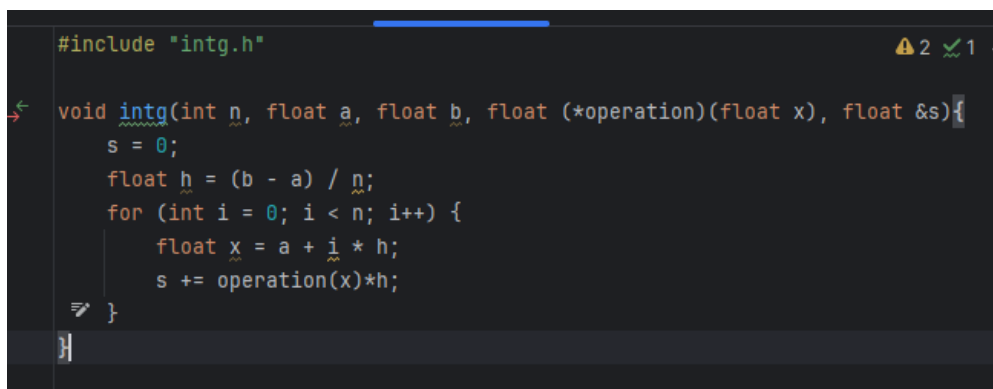
Составить подпрограмму-процедуру INTG вычисления площади, ограниченной заданной функцией и осью x на заданном отрезке по формуле :  $S = \frac{b-a}{n} \sum_{i=1}^n f(x_i)$ , где n – количество отрезков разбиения, a – начало, b – конец заданного отрезка, а f(x) – произвольно заданная функция.

Функцию f(x) передать в процедуру через параметр. В основной программе использовать процедуру INTG для вычисления площади функций:

1.  $\sin(x)/x$  ( a = 0.1 , b = 1 )
2.  $\text{tg}(x + 1)/(x + 1)$  ( a = 1, b = 2)

Количество отрезков n задается с клавиатуры. Проверить программу при n=10,20,30.

Начнём решение с написания файла реализации процедуры INTG

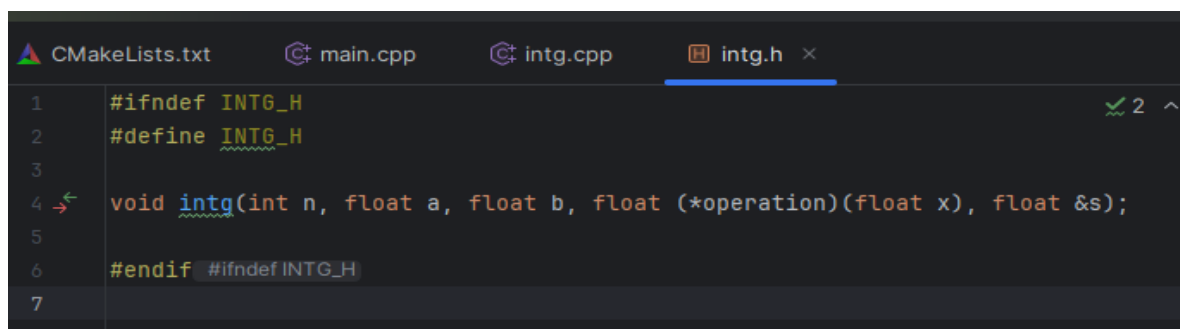


```
#include "intg.h"

void intg(int n, float a, float b, float (*operation)(float x), float &s){
    s = 0;
    float h = (b - a) / n;
    for (int i = 0; i < n; i++) {
        float x = a + i * h;
        s += operation(x)*h;
    }
}
```

Рисунок 9 – код файла реализации подпрограммы

Создадим заголовочный файл для подпрограммы INTG



```
CMakeLists.txt  main.cpp  intg.cpp  intg.h x
1  #ifndef INTG_H
2  #define INTG_H
3
4  void intg(int n, float a, float b, float (*operation)(float x), float &s);
5
6  #endif #ifndef INTG_H
7
```



Рисунок 10 – заголовочный файл intg.h

Теперь напишем код основной программы

```
1  #include <iostream>
2  #include "intg.h"
3  #include <cmath>
4
5  float first_f(float x) {
6      return sin(x)/x;
7  }
8
9  float second_f(float x) {
10     return tan(x+1)/(x+1);
11 }
12
13 int main() {
14     int n;
15     float s, a, b;
16     std::cout << "Enter n: " << std::endl;
17     std::cin >> n;
18
19     s = 0, a = 0.1, b = 1;
20     intg(n,a,b,first_f, [&s]);
21     std::cout << "Integral of sin(x)/x" << std::endl;
22     std::cout << s << std::endl;
23
24     s = 0, a = 1, b = 2;
25     intg(n,a,b,second_f, [&s]);
26     std::cout << "Integral of tg(x+1)/(x+1)" << std::endl;
27     std::cout << s << std::endl;
28
29     return 0;
30 }
31
```

Рисунок 11 – код программы

Запустим программу при  $n=10$ ,  $n=20$  и  $n=30$ , чтобы убедиться в том, что программа работает правильно

```
D:\Лаб\АиП\Dz_2\3\cmake-build-debug\3.exe
Enter n:
10
Integral of sin(x)/x
0.853017
Integral of tg(x+1)/(x+1)
-0.43167

Process finished with exit code 0
```

Рисунок 12 – результат работы программы при  $n = 10$

```
D:\Лаб\АиП\Dz_2\3\cmake-build-debug\3.exe
Enter n:
20
Integral of sin(x)/x
0.849623
Integral of tg(x+1)/(x+1)
-0.403636
Process finished with exit code 0
```

Рисунок 13 – результат работы программы при  $n = 20$

```
D:\Лаб\АиП\Dz_2\3\cmake-build-debug\3.exe
Enter n:
30
Integral of sin(x)/x
0.848472
Integral of tg(x+1)/(x+1)
-0.394572
Process finished with exit code 0
```

Рисунок 14 – результат работы программы при  $n = 30$

В верности полученных результатов можем убедиться, вычислив эти интегралы вручную. Также видим, что при увеличении количества разбиений, уменьшается погрешность.

Составим схемы алгоритмов для этой программы и её подпрограмм

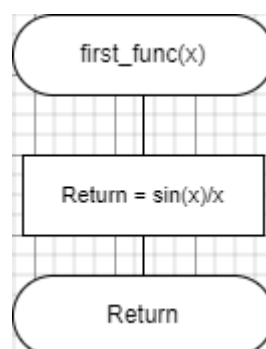


Рисунок 15 – схема алгоритма для функции, вычисляющей  $\sin(x)/x$

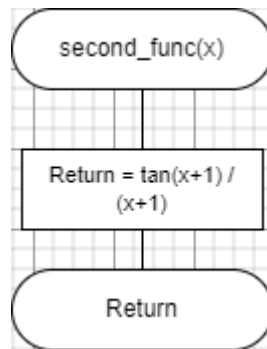


Рисунок 16 – схема алгоритма для функции, вычисляющей  $\text{tg}(x+1)/(x+1)$

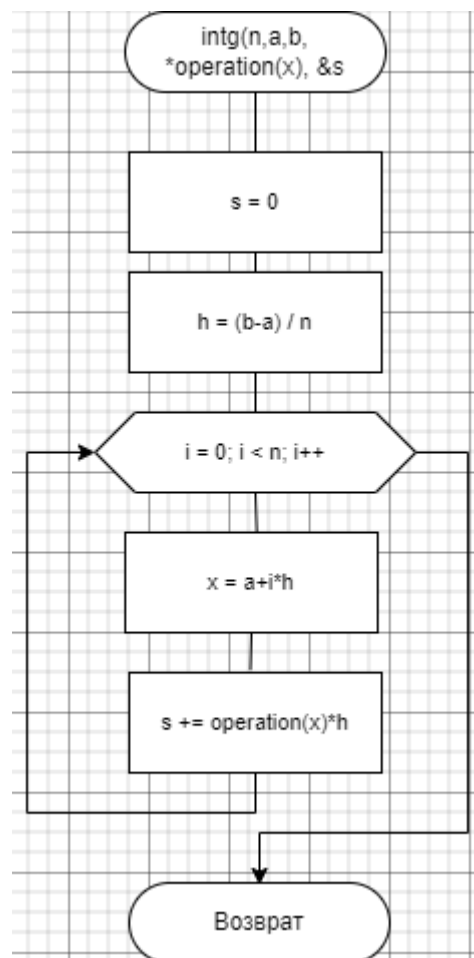


Рисунок 17 – схема алгоритма процедуры intg

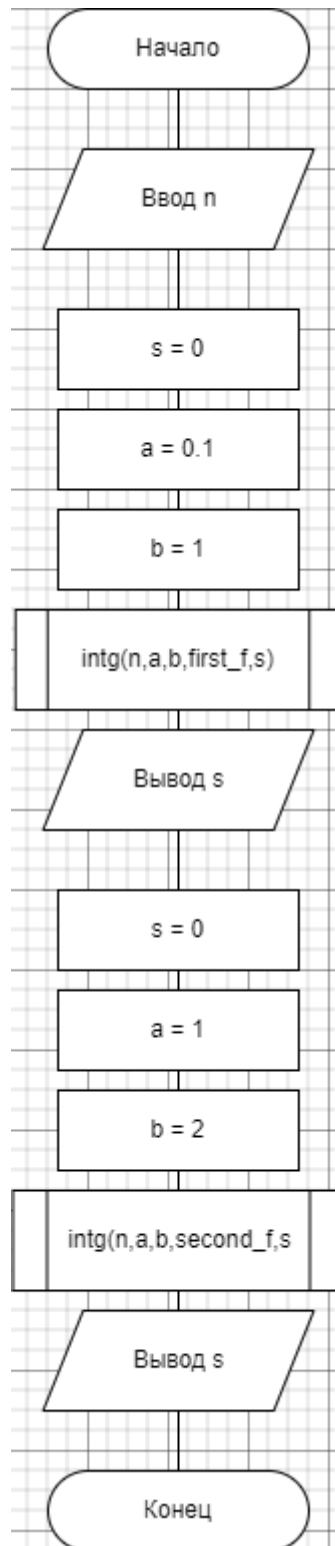


Рисунок 18 – схема алгоритма основной программы

**Вывод:** в ходе выполнения домашнего задания я получил базовые навыки работы с массивами и строками, научился создавать псевдомодули и работать с процедурным типом параметров