



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

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


Отчет

по домашнему заданию № 2

**Название домашнего задания: Обработка одномерных массивов.
Обработка одномерных массивов. Создание псевдомодулей.
Процедурный тип параметров.**

Дисциплина: Алгоритмизация и программирование

Студент гр. ИУ6-14Б

 **28.10.2023**

(Подпись, дата)

А.С. Воеводин

(И.О. Фамилия)

Преподаватель

 **28.10.2023**

(Подпись, дата)

О.А. Веселовская

(И.О. Фамилия)

Цель задания – Обработка массива и его сортировка, работа со строкой, создание псевдомодуля.

Задание – Сортировать одномерный массив символов по алфавиту. Использовать метод обменов. Дано натуральное число n , латинские символы S_1, S_2, \dots, S_n , образующие строку. Группы символов, разделенных одним или несколькими пробелами и не содержащих пробелов внутри себя, назовем словами. Найти количество слов, начинающихся с буквы «А» и оканчивающихся на «Е».

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

Составить подпрограмму-процедуру TRAP для вычисления площади, ограниченной заданной функцией и осью x на заданном интервале по формуле , где $f(x)$ – заданная функция, $[a, b]$ – заданный интервал, n – число отрезков разбиения интервала, h – шаг изменения x , находится по формуле $(b-a)/n$.

Функцию $f(x)$ передать в процедуру через параметр. В основной программе использовать процедуру TRAP для вычисления площади, ограниченной функциями:

1. $x + \cos(x)$, $a = -1$, $b = 4$,
2. $\text{tg}(x+1)/(x+1)$, $a = -0,5$, $b = 0$.

Количество разбиений n задается с клавиатуры в основной программе. Проверить программу при $n=10,20,30$.

Ход работы:

- Написание программы и схемы алгоритма в части 1
- Написание программы и схемы алгоритма в части 2
- Написание программы и схемы алгоритма в части 3
- Вывод

Для начала выполним первую часть задания, а именно – напишем программу, сортирующую методом обменов:

```

char arr[1000];
std::cout << "Enter your string: ";
gets_s(arr, 1000);

for (size_t i = 0; i < strlen(arr); ++i) {
    for (size_t j = i + 1; j < strlen(arr); ++j) {
        if (arr[j] < arr[i]) {
            std::swap(arr[j], arr[i]);
        }
    }
}

std::cout << "String after sort: " << arr << std::endl;

```

Рисунок 1 – Код программы первой части задания

```

Enter your string: fasffSFSFKJfasdfasfsH0IUYHOGfasjfsa
String after sort: FFGHHIJK00SSUYaaaaadfffffffffjssssss

```

Рисунок 2 – Тестовые данные 1

```

Enter your string: qwertyuiopasdfghjklzxcvbnm
String after sort: abcdefghijklmnopqrstuvwxyz

```

Рисунок 3 – Тестовые данные 2

```

Enter your string: qwertyuiopasdfghjklzxcvbnmQWERTYUIOPASDFGHJKLZXCVBNM
String after sort: ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz

```

Рисунок 4 – Тестовые данные 3

Как видно из рисунков выше, программа обрабатывает верно: сортирует строчку в лексикографическом порядке. Изобразим алгоритм в виде схемы:

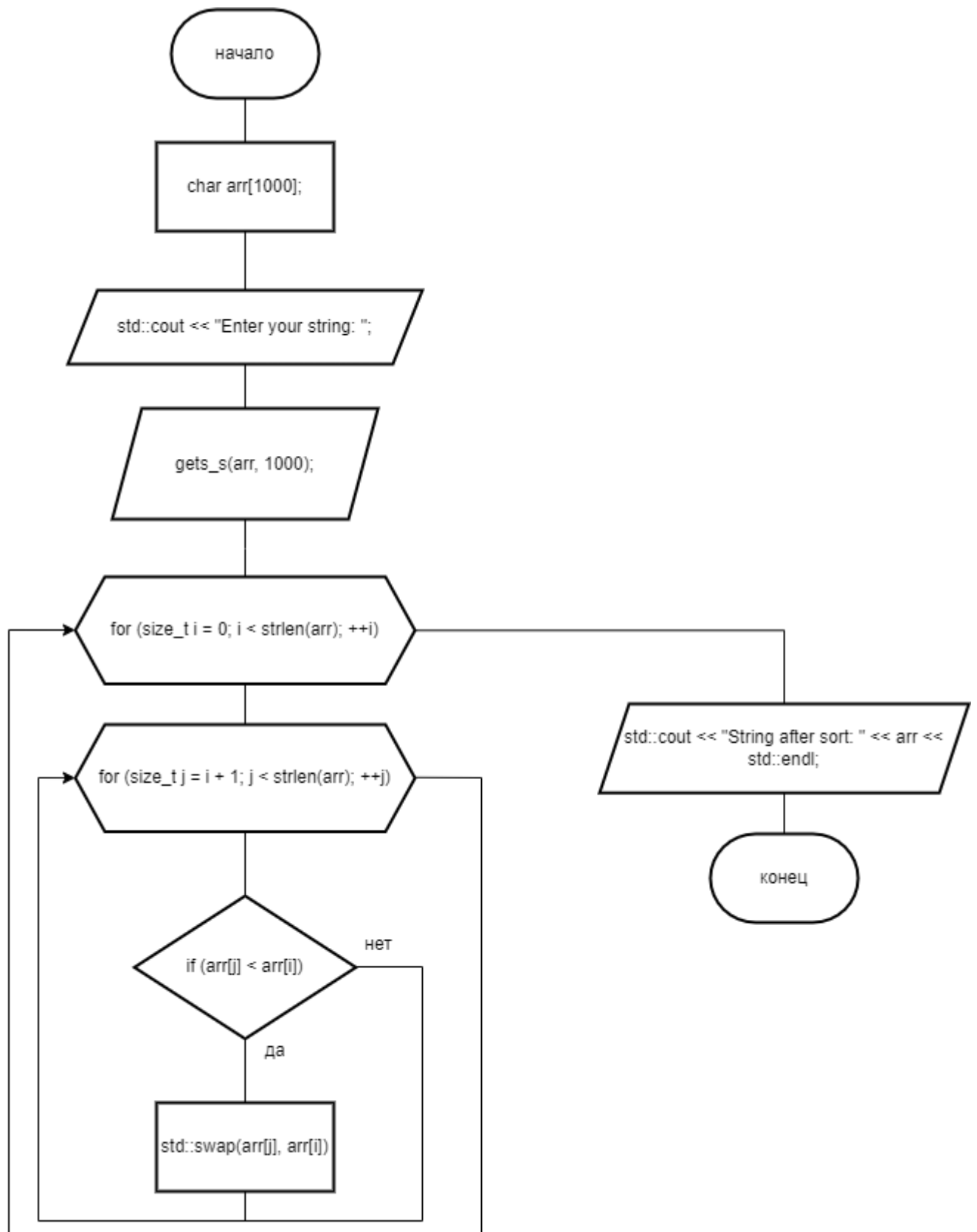


Рисунок 5 – Схема алгоритма первой части задания

Далее напомним программу для второй части задания:

```

int n = 0;
std::cout << "Enter your buffer size: ";
std::cin >> n;
char* arr = new char[n + 1];
gets_s(arr, n + 1);
std::cout << "Enter your string: ";
gets_s(arr, n + 1);

size_t start = 0;
size_t end = 0;

size_t count = 0;
for (size_t i = 0; i < strlen(arr); ++i) {
    if (arr[i] == ' ') {
        if (arr[start] == 'A' && arr[end - 1] == 'E') {
            ++count;
        }
        start = i + 1;
        end = i + 1;
    }
    else {
        ++end;
    }
}

if (end != start && arr[start] == 'A' && arr[end - 1] == 'E') {
    ++count;
}

std::cout << "Count words that starts with A and ends with E is: " << count << std::endl;
delete[] arr;
arr = nullptr;

```

Рисунок 6 – Код программы для второй части задания

Проверим работоспособность программы:

```

Enter your buffer size: 1000
Enter your string: hello world from IU6-14B
Count words that starts with A and ends with E is: 0

```

Рисунок 7 – Тестовые данные 1

```

Enter your buffer size: 1000
Enter your string: asdf dsaf adsfsad fadsf asdkfh sadfh sadfjl
Count words that starts with A and ends with E is: 0

```

Рисунок 8 – Тестовые данные 2

```

Enter your buffer size: 1000
Enter your string: AE SDFASFASDF AE AFSAFDASFADSFEEEEEEAAFDASDFE ADFADSFASFDAFe
Count words that starts with A and ends with E is: 3

```

Рисунок 9 – Тестовые данные 3

Несложно убедиться, что программа выполняется верно. Теперь изобразим этот алгоритм в виде схемы:

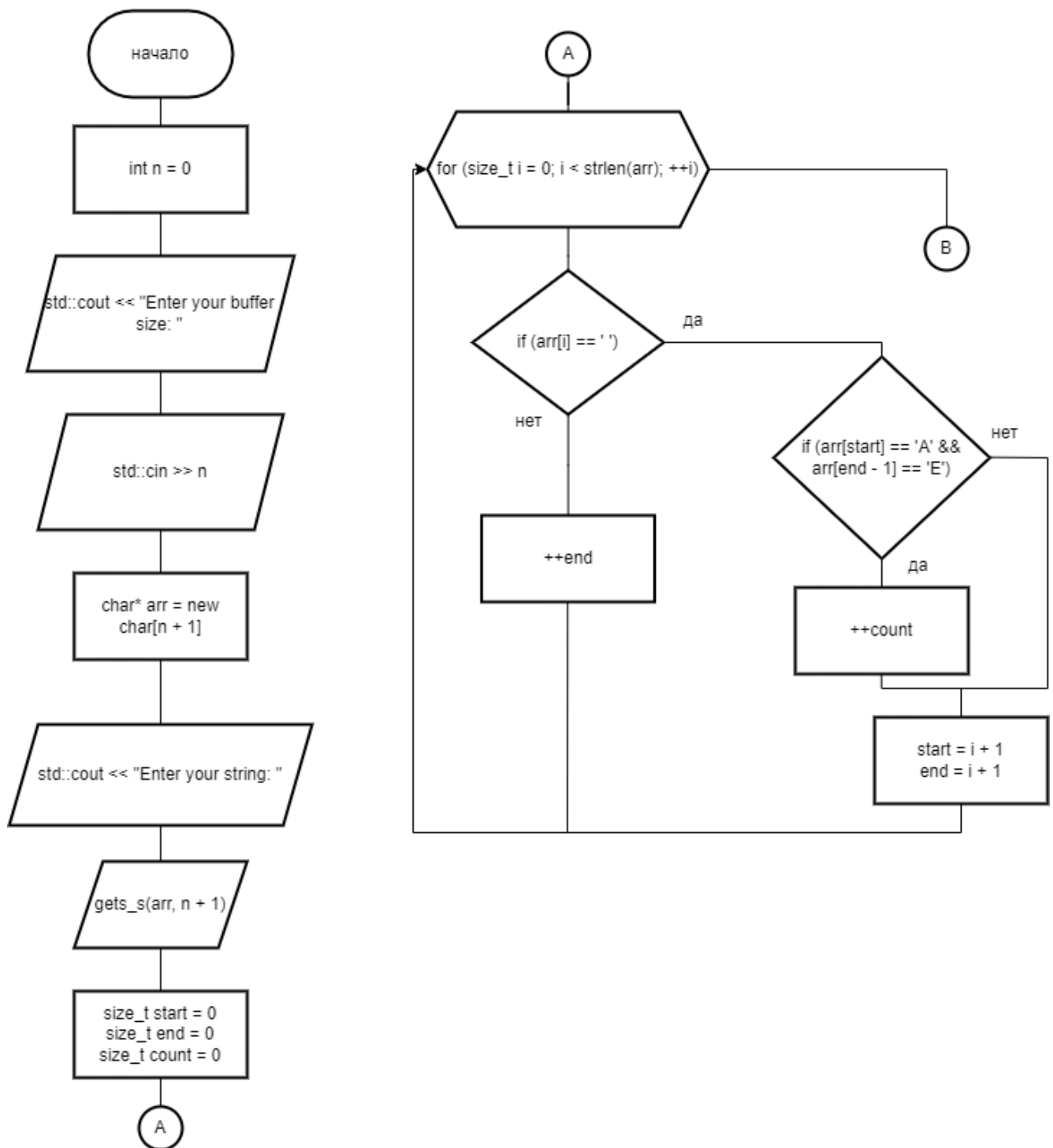


Рисунок 10 – Схема алгоритма часть 1

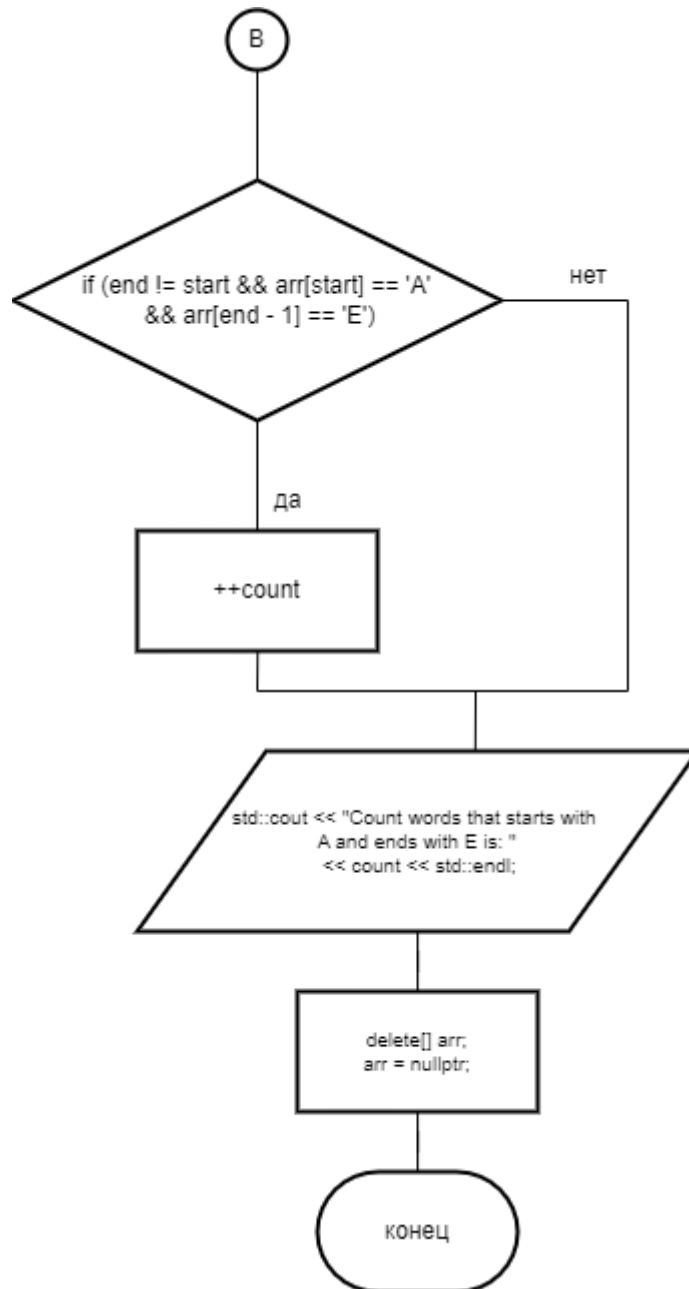


Рисунок 11 – Схема алгоритма часть 2

Далее напишем программу для третьей части задания: создадим заголовочный файл Trap.hpp и файл реализации Trap.cpp:

```
#pragma once

double trap(double a, double b, double n, double (*func)(double x));
double func1(double x);
double func2(double x);
```

Рисунок 12 – Заголовочный файл Trap.hpp

```

#include "Trap.hpp"
#include <cmath>

double trap(double a, double b, double n, double(*func)(double x))
{
    double summ = 0.f;
    double h = (b - a) / n;

    for (size_t i = 2; i < n; ++i) {
        summ += func(a + h * i);
    }

    summ = h * ((func(a) + func(b)) / 2 + summ);

    return summ;
}

double func1(double x) {
    return x + std::cos(x);
}

double func2(double x) {
    return std::tan(x + 1) / (x + 1);
}

```

Рисунок 13 – Файл реализации Trap.cpp


```

double a = 0;
double b = 0;
double n = 0;
short func_num = 0;
double (*func)(double);
std::cout << "Enter a, b, n: ";
std::cin >> a >> b >> n;
std::cout << "Choose your function:\n"
    << "1) x + cos(x)\n"
    << "2) tg(x+1)/(x+1)\n";
std::cin >> func_num;
if (func_num == 1) {
    func = func1;
}
else {
    func = func2;
}

std::cout << trap(a, b, n, func) << std::endl;

```

Рисунок 14 – Основной файл

Введём для теста некоторые значения по условию:

```

Enter a, b, n: -1 4 10
Choose your function:
1) x + cos(x)
2) tg(x+1)/(x+1)
1
7.39411

```

Рисунок 15 – Тестовые данные 1

```

Enter a, b, n: -1 4 20
Choose your function:
1) x + cos(x)
2) tg(x+1)/(x+1)
1
7.5888

```

Рисунок 16 – Тестовые данные 2

```
Enter a, b, n: -1 4 30
Choose your function:
1)  $x + \cos(x)$ 
2)  $\text{tg}(x+1)/(x+1)$ 
1
7.61129
```

Рисунок 17 – Тестовые данные 3

```
Enter a, b, n: -0.5 0 10
Choose your function:
1)  $x + \cos(x)$ 
2)  $\text{tg}(x+1)/(x+1)$ 
2
0.57893
```

Рисунок 18 – Тестовые данные 4

```
Enter a, b, n: -0.5 0 20
Choose your function:
1)  $x + \cos(x)$ 
2)  $\text{tg}(x+1)/(x+1)$ 
2
0.606858
```

Рисунок 19 – Тестовые данные 5

```
Enter a, b, n: -0.5 0 30
Choose your function:
1)  $x + \cos(x)$ 
2)  $\text{tg}(x+1)/(x+1)$ 
2
0.61607
```

Рисунок 20 – Тестовые данные 6

Результат программы можно проверить, вычислив интеграл, но так как количество разбиений довольно маленькое, то погрешность относительно истинного результата довольно велика, но если увеличивать n , то погрешность будет уменьшаться. Изобразим этот алгоритм в виде схемы:

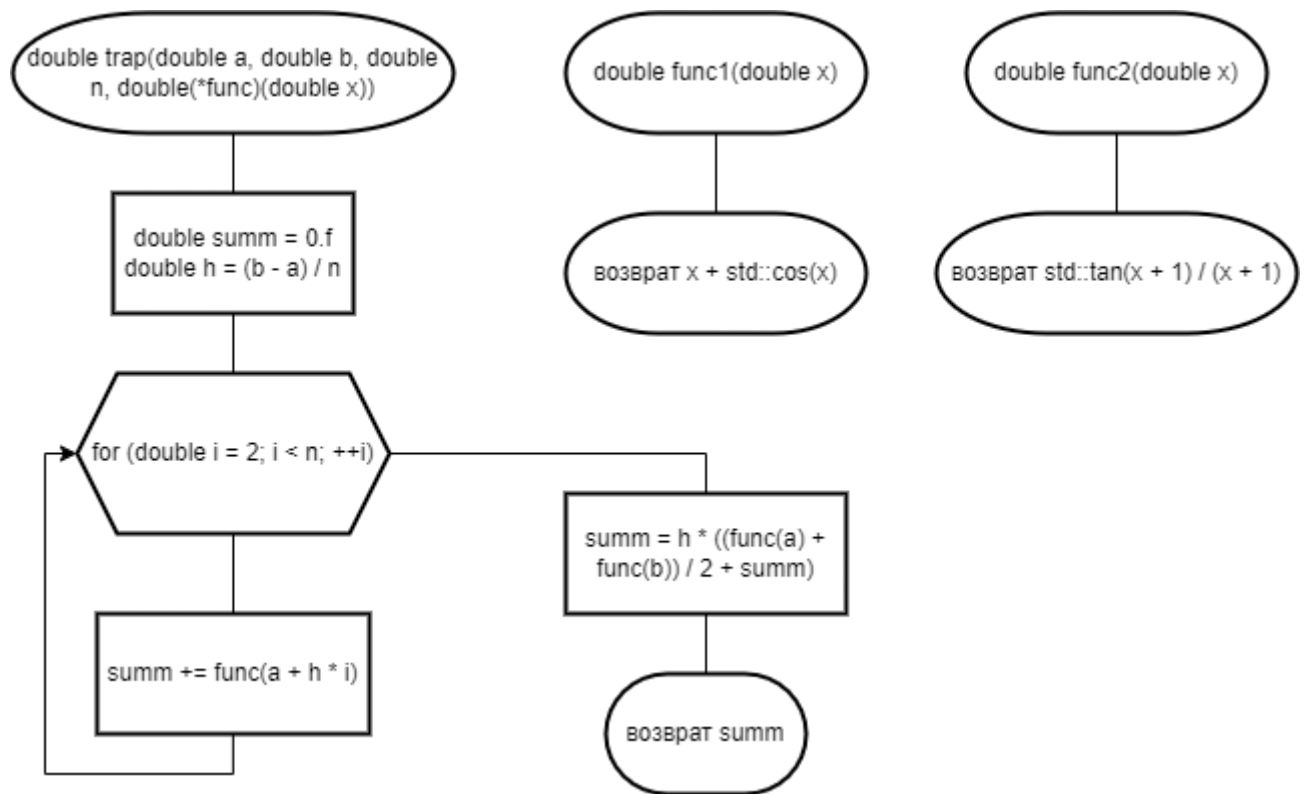


Рисунок 21 – Схема алгоритма используемых функций

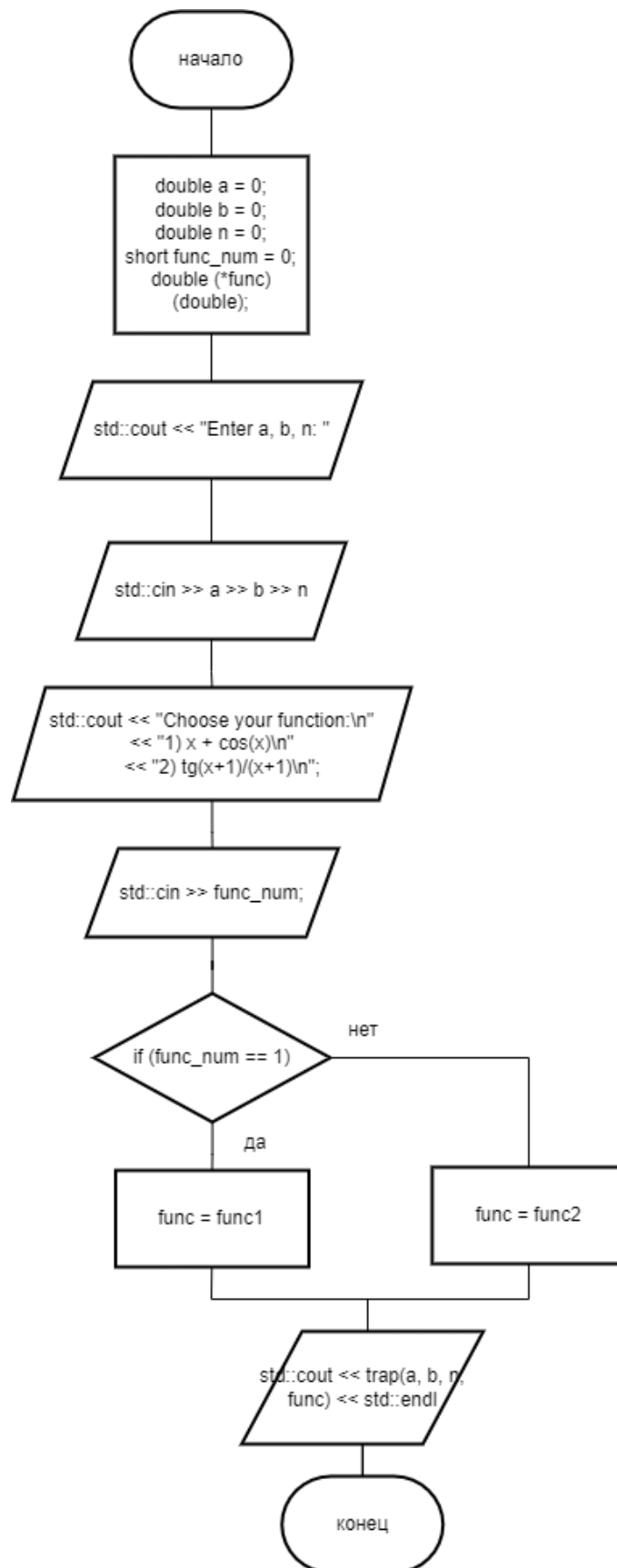


Рисунок 22 – Основная часть программы

Вывод: В ходе домашнего задания я научился обрабатывать массивы и строки, создавать псевдомодули, работать с функциями, которые передаются в параметрах через указатели на функции, использовать процедурный тип программирования.