Федеральное государственное автономное образовательное учреждение высшего образования

«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)

О т ч е т

по домашнему заданию № 2

Название домашнего задания: Обработка массивов и строк. Создание программных модулей.

Дисциплина: Алгоритмизация и программирование.

Студент гр. ИУ6-13Б _____ **С.М Соболев**
(Подпись, дата) (И.О. Фамилия)

Преподаватель _____ **О.А. Веселовская**
(Подпись, дата) (И.О. Фамилия)

Москва, 2025

Часть 1. Обработка одномерных массивов.

Цель: Разработка программы, которая сортирует массив вещественных чисел по невозрастанию.

Задание: Упорядочить массив вещественных чисел R(n) (n<=40) по невозрастанию значений его элементов, используя метод сортировки выбором.

Схема алгоритма программы:

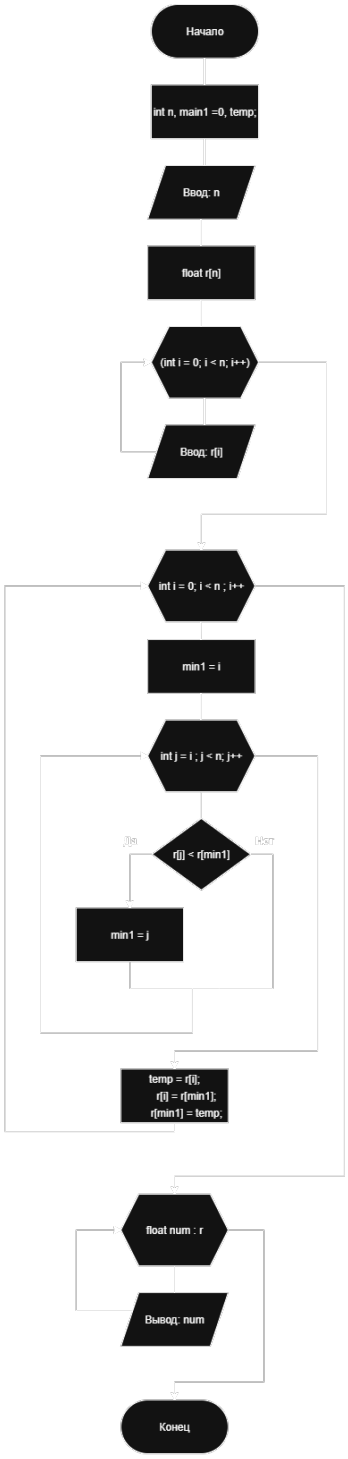

Рисунок 1 – схема алгоритма программы

Код программы:

```
int main()
{
    int n, min1 = 0;
    float temp;
    cout << "Введите n" << endl;
    cin >> n;
    float r[n];
    for (int i = 0; i < n; i++) cin >> r[i];
    for (int i = 0; i < n ; i++){
        min1 = i;
        for (int j = i ; j < n; j++){
            if (r[j] < r[min1]){
                min1 = j;
                }
        }
        temp = r[i];
        r[i] = r[min1];
        r[min1] = temp;
    }
    for (float num : r) cout << num << " ";
    return 0;
}
```

Рисунок 2 – Код программы

Тестирование.

Таблица 1 – Тестирование программы

Входные данные	**Выходные данные**
10 12.345 67.890 34.567 89.012 45.678 90.123 56.789 23.456 78.901 41.234	12.345 23.456 34.567 41.234 45.678 56.789 67.89 78.901 89.012 90.123
15 12.345 67.890 34.567 89.012 45.678 90.123 56.789 23.456 78.901 41.234 15.678 82.345 29.012 76.543 38.901	12.345 15.678 23.456 29.012 34.567 38.901 41.234 45.678 56.789 67.89 76.543 78.901 82.345 89.012 90.123

Вывод: в ходе выполнения задания была разработана и протестирована программа сортировки методом вставки. Задание помогло закрепить знания о методах сортировки.

Часть 2. Обработка строк.

Цель: разработка программы, которая определяет количество слов с четными номерами, которые имеют длину более трех букв.

Задание: Дана непустая последовательность слов из строчных латинских букв (количество слов больше 3): между соседними словами – запятая, за последним словом – точка. Определить количество слов с четными номерами, которые имеют длину более трех букв. Вывести строку и найденные слова. Если таковых нет – вывести сообщение.

Схема алгоритма программы:

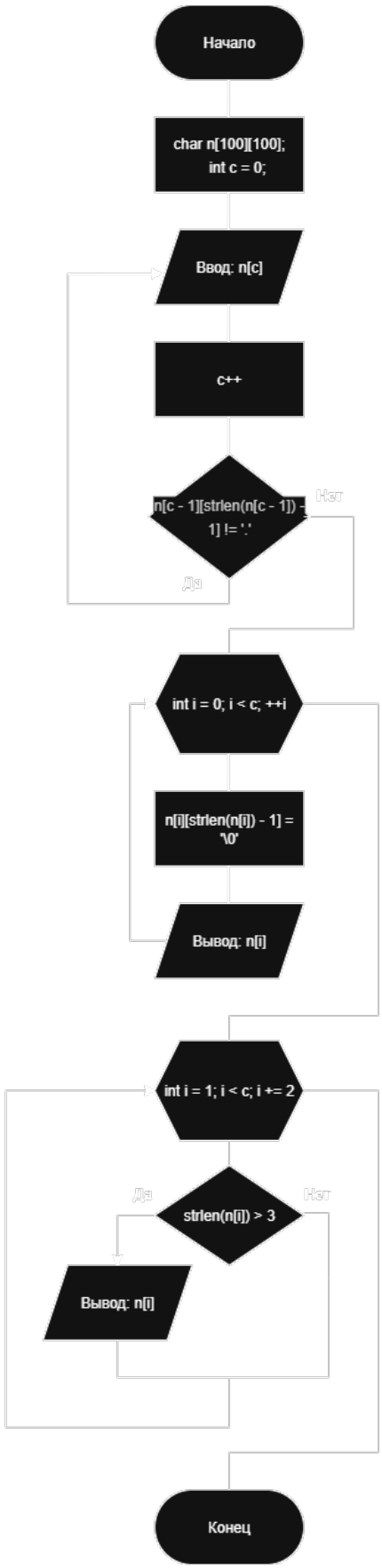

Рисунок 3 – Схема алгоритма программы

Код программы:

```
#include <stdio.h>
#include <iostream>
#include <string.h>
using namespace std;
int main(void)
{
    // здесь продолжайте программу
    puts("Вводите слова через запятую:");
    char n[100][100];
    int c = 0;
    do {
        cin >> n[c];
        c++;
    } while (n[c - 1][strlen(n[c - 1]) - 1] != '.');
    puts("Список введеных слов:");
    for (int i = 0; i < c; ++i){
        n[i][strlen(n[i]) - 1] = '\0';
        cout << n[i] << ' ';
    }
    puts("\nНайденные слова:");
    for (int i = 1; i < c; i += 2){
        if (strlen(n[i]) > 3)
        cout << n[i]<< ' ';
    }
    return 0;
}
```

Рисунок 4 – Код программы

Таблица 2 – Тестирование программы

Входные данные	Выходные данные
apple, banana, computer, dragon, elephant, fantasy, guitar.	List words: apple banana computer dragon elephant fantasy guitar Find words: banana dragon fantasy
it, on, cat, dog, book, tree, phone.	List words: it on cat dog book tree phone Find words: tree

Вывод: в результате выполнения задания была создана и протестирована программа, которая ищет слова на четных позициях, имеющие длину более 3 букв.

Часть 3. Создание псевдомодулей. Процедурный тип параметров.

Цель: создать программу для проверки принадлежности точки плоскости с координатами (x, y) данной кривой y=f(x).

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

Составить подпрограмму-процедуру NEIBR проверки принадлежности точки плоскости с координатами (x, y) данной кривой y=f(x).

Функцию f(x) передать в процедуру через параметр. В основной программе использовать процедуру NEIBR для проверки принадлежности десяти различных точек кривым y=ln(x) и y=x^2-4.

Схема алгоритма программы:

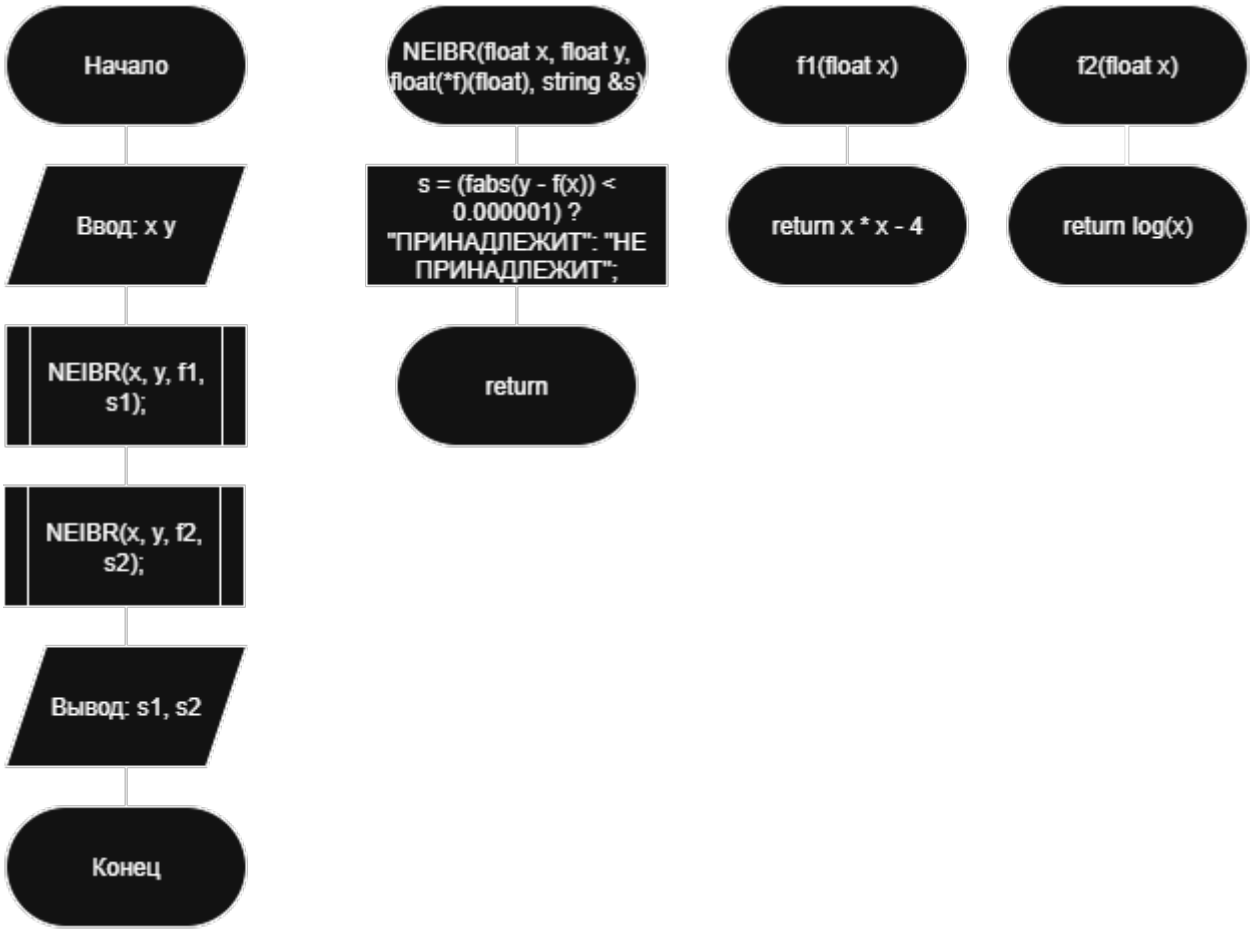

Рисунок 5 – Схема алгоритма программы

Код программы:

```cpp
#include <iostream>
#include "NEIBR.h"
#include <math.h>
#include <string>
using namespace std;

int main()
{
    float x, y;
    string s1 = "", s2 = "";
    cout << "Введите x y:" << endl;
    cin >> x >> y;
    NEIBR(x, y, f1, s1);
    NEIBR(x, y, f2, s2);
    cout << "Точка " << x << " " << y << " " << s1 << " кривой x^2 - 4" << endl;
    cout << "Точка " << x << " " << y << " " << s2  << " кривой ln(x)";
    return 0;
}
```

Рисунок 6 – Код файла main.cpp

```cpp
#include "NEIBR.h"
#include <math.h>
#include <string>
using namespace std;
void NEIBR(float x, float y, float(*f)(float), string &s){
    s = (fabs(y - f(x)) < 0.000001) ? "ПРИНАДЛЕЖИТ": "НЕ ПРИНАДЛЕЖИТ";
}

float f1(float x){
    return x * x - 4;
}

float f2(float x){
    return log(x);
}
```

Рисунок 7 – Код файла реализации

```cpp
#ifndef NEIBR_H
#define NEIBR_H
#include <string>

void NEIBR(float x, float y, float (*f)(float), std::string &s);
float f1(float x);
float f2(float x);

#endif // NEIBR_H
```

Рисунок 8 – Код заголовочного файла

Таблица 3 – Тестирование программы

x y	Вывод
1 0	Точка 1 0 НЕ ПРИНАДЛЕЖИТ кривой x^2 - 4 Точка 1 0 ПРИНАДЛЕЖИТ кривой ln(x)
2 0	Точка 2 0 ПРИНАДЛЕЖИТ кривой x^2 - 4 Точка 2 0 НЕ ПРИНАДЛЕЖИТ кривой ln(x)
5 5	Точка 5 5 НЕ ПРИНАДЛЕЖИТ кривой x^2 - 4 Точка 5 5 НЕ ПРИНАДЛЕЖИТ кривой ln(x)

Вывод: был разработана программа, заголовочный файл и файл реализации, которые содержать указанную подпрограмму. Данная программа успешно вычисляет принадлежность точки к заданной кривой.