

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ВЯТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Институт математики и информационных систем
Факультет автоматики и вычислительной техники
Кафедра электронных вычислительных машин

Отчет по лабораторной работе №5
по дисциплине
«Программирование»

Выполнил студент гр. ИВТб-1301-05-00	_____ /Макаров С.А./
Руководитель зав. кафедры ЭВМ	_____ /Долженкова М.Л./

Киров 2025

Цель

Цель лабораторной работы: получить базовые сведения о наиболее известных алгоритмах сортировки, изучить принципы работы с текстовыми файлами.

Задание

1. Реализовать сортировку данных с помощью алгоритма подсчётом.
2. Реализовать сортировку данных с помощью поразрядного алгоритма.
3. В обоих случаях необходимо предусмотреть возможность изменения компаратора (реализация компаратора в виде передаваемой в подпрограмму функции).
4. Считывание и вывод данных необходимо производить из текстового файла.
5. Для демонстрации работы программных реализаций самостоятельно подготовить варианты входных данных (при этом объем тестовых файлов должен позволять оценить скорость работы программ).

Решение

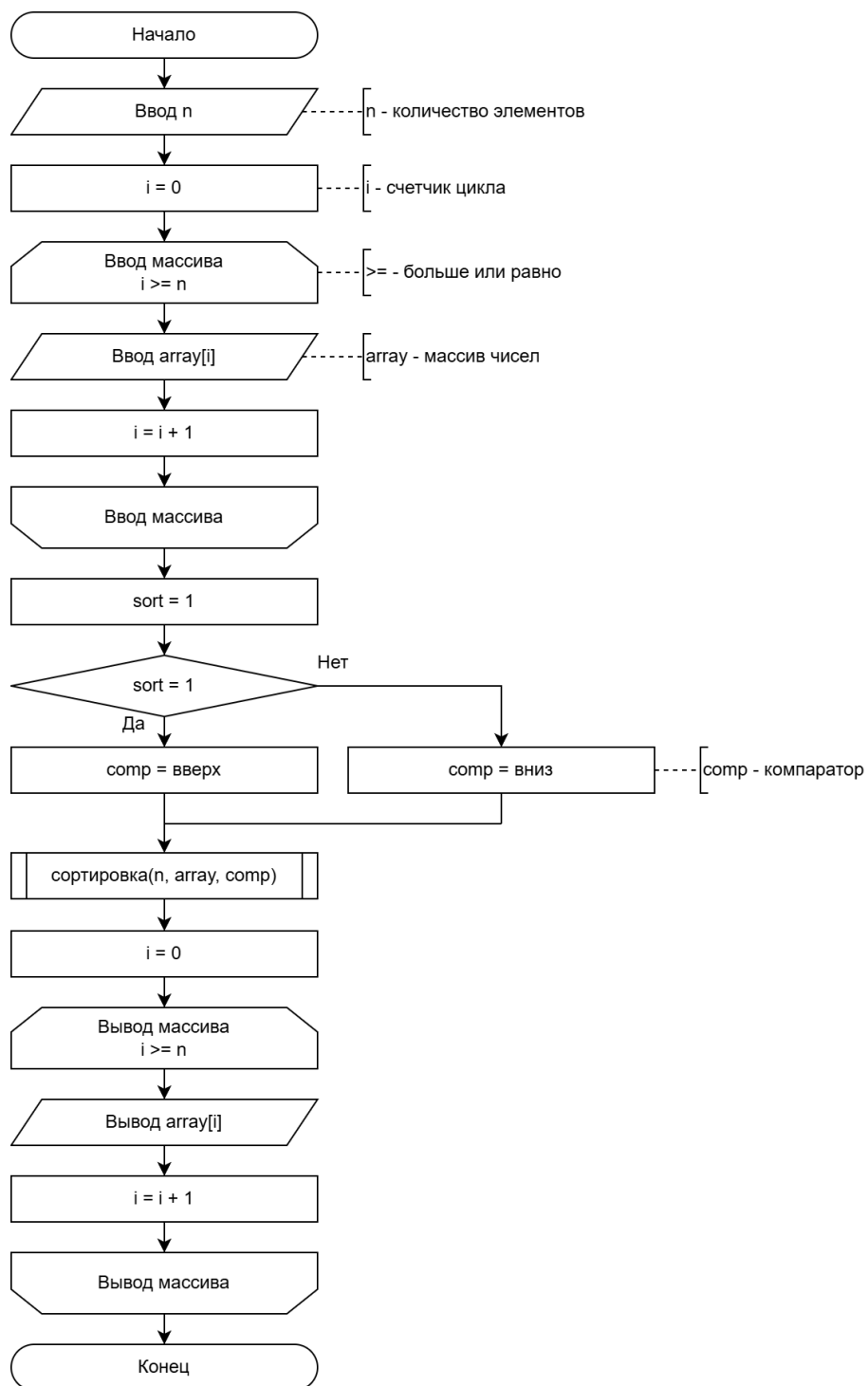


Рисунок 1 – Схема алгоритма сортировки подсчетом

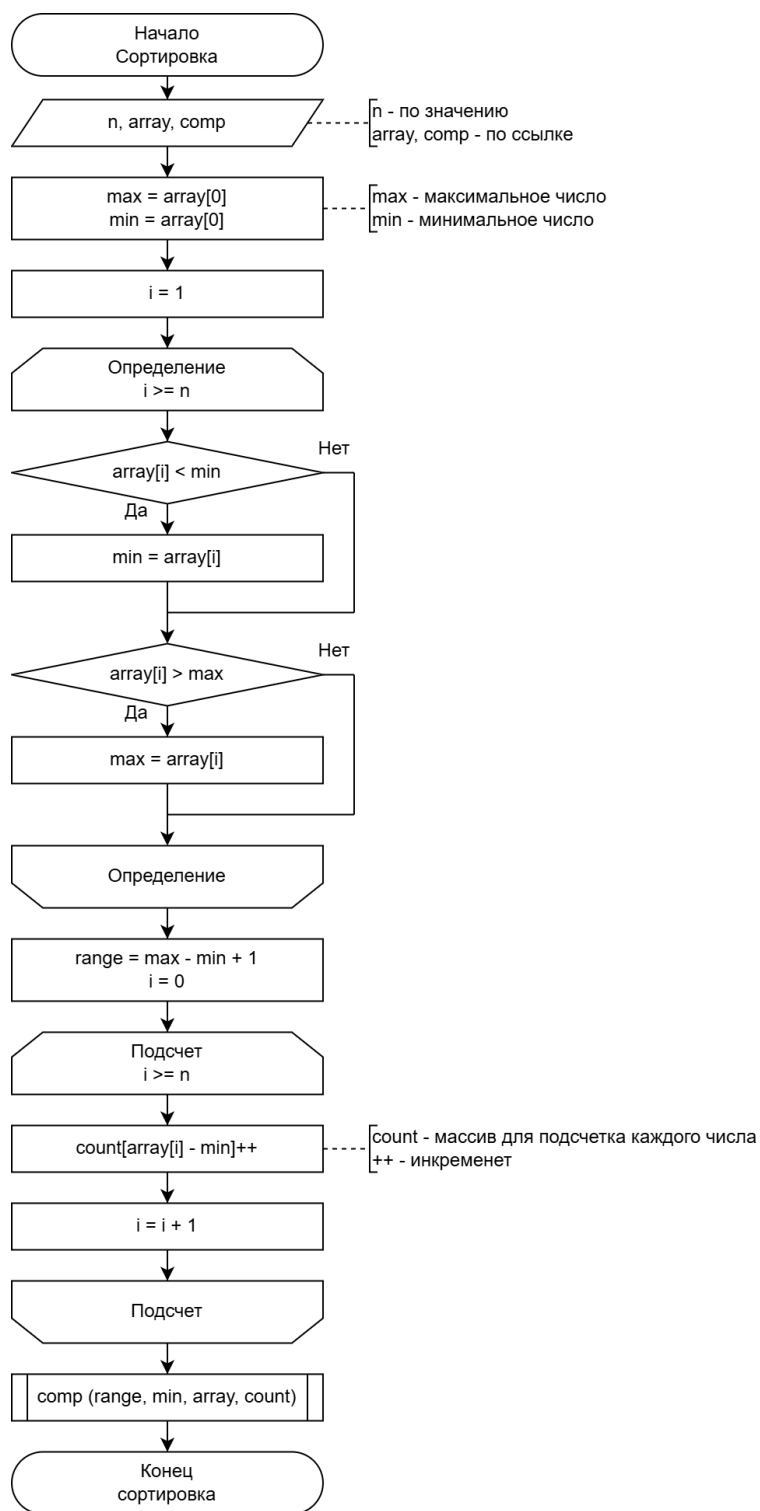


Рисунок 2 – Схема алгоритма подпрограммы «Сортировка»

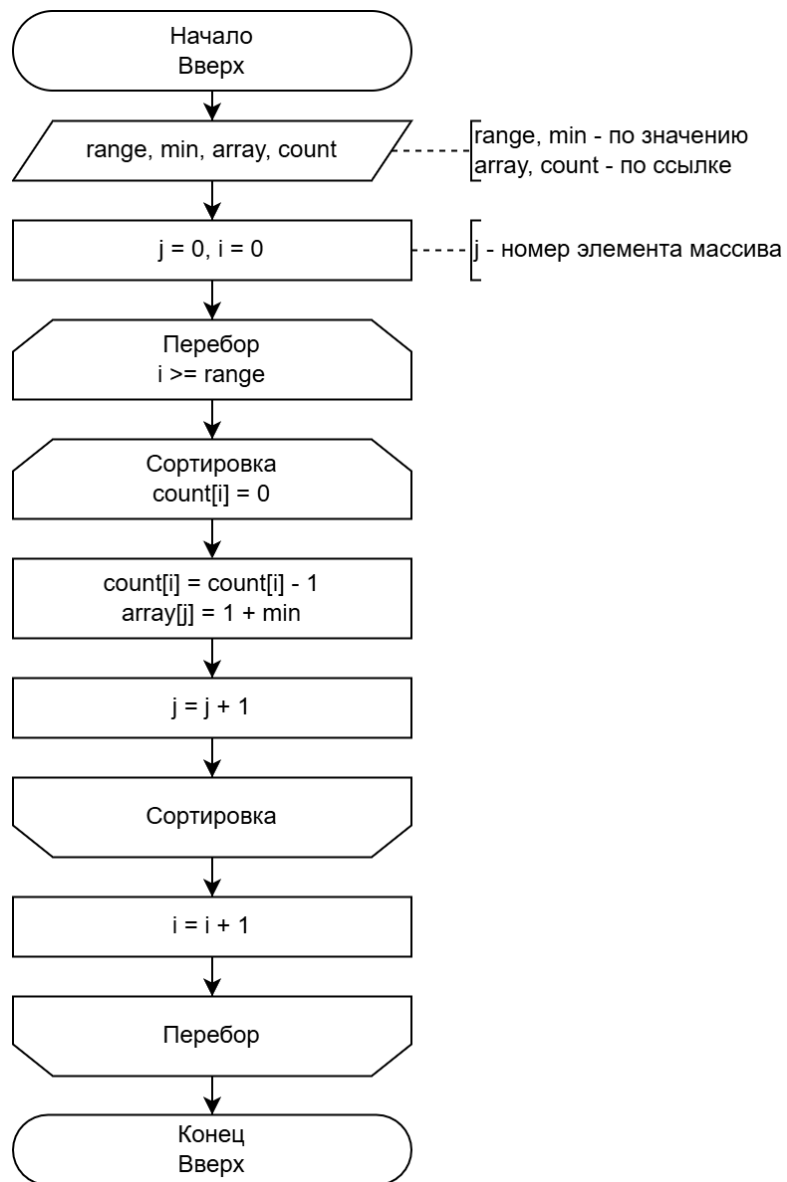


Рисунок 3 – Схема алгоритма подпрограммы «Вверх»

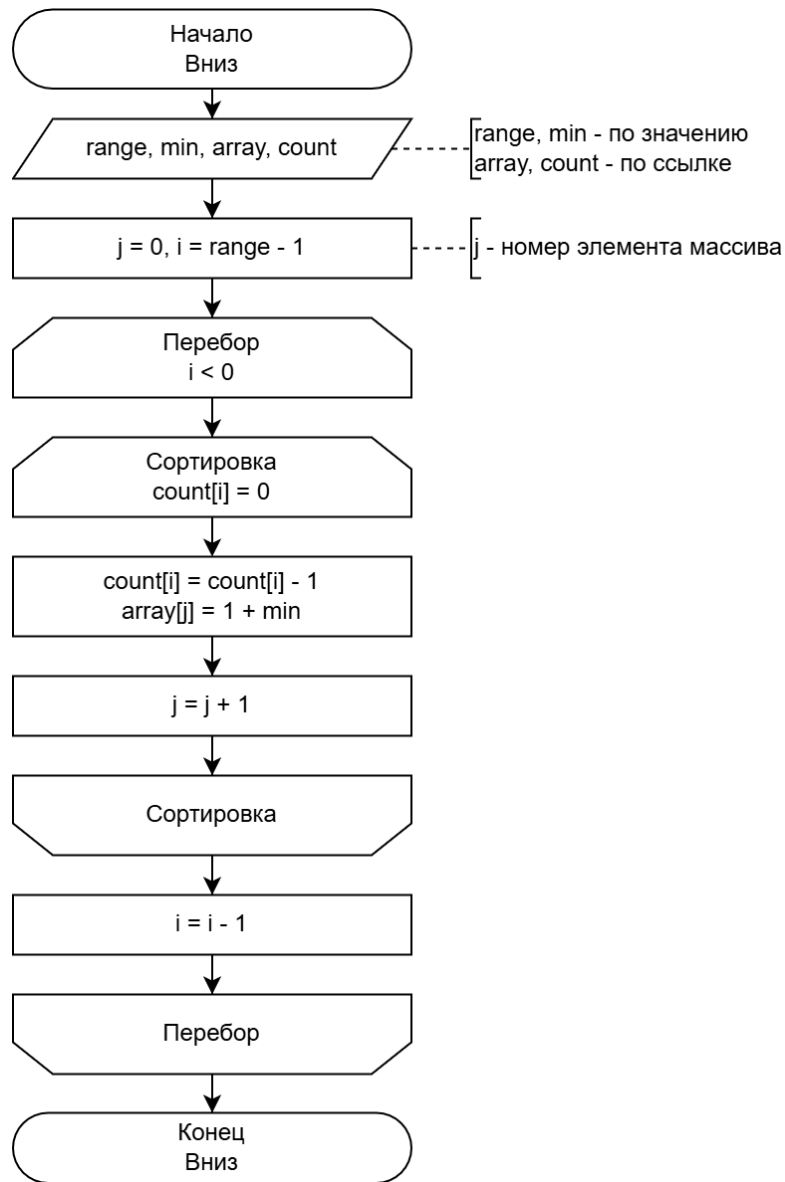


Рисунок 4 – Схема алгоритма подпрограммы «Вниз»

```

#include <stdio.h>
#include <stdlib.h>
void sort(int n, int array[],
void(*comparator)(int range, int min, int array[], int count[])) {
    int min = array[0];
    int max = array[0];
    for (int i = 1; i < n; i++) {
        if (array[i] < min) min = array[i];
        if (array[i] > max) max = array[i];
    }
    int range = max - min + 1;
    int* count = (int*)calloc(range, sizeof(int));

```

```

        for (int i = 0; i < n; i++) count[array[i] - min]++;
        comparator(range, min, array, count);
    }
void comparatorUp(int range, int min, int array[], int count[]) {
    int j = 0;
    for (int i = 0; i < range; i++) {
        while (count[i] > 0) {
            count[i]--;
            array[j++] = i + min;
        }
    }
}
void comparatorDown(int range, int min, int array[], int count[]) {
    int j = 0;
    for (int i = range - 1; i >= 0; i--) {
        while (count[i] > 0) {
            count[i]--;
            array[j++] = i + min;
        }
    }
}
int main() {
    FILE* input = fopen("../input.txt", "r");
    FILE* output = fopen("../output.txt", "w");
    int n = 0;
    fscanf_s(input, "%d", &n);
    int* array = (int*)calloc(n, sizeof(int));
    for (int i = 0; i < n; i++) fscanf_s(input, "%d", &array[i]);
    fclose(input);
    void(*comparator)(int range, int min, int array[], int count[]);
    comparator = comparatorUp;
    sort(n, array, comparator);
    for (int i = 0; i < n; i++) {
        fprintf(output, "%d ", array[i]);
    }
    fclose(output);
    free(array);
    return 0;
}

```

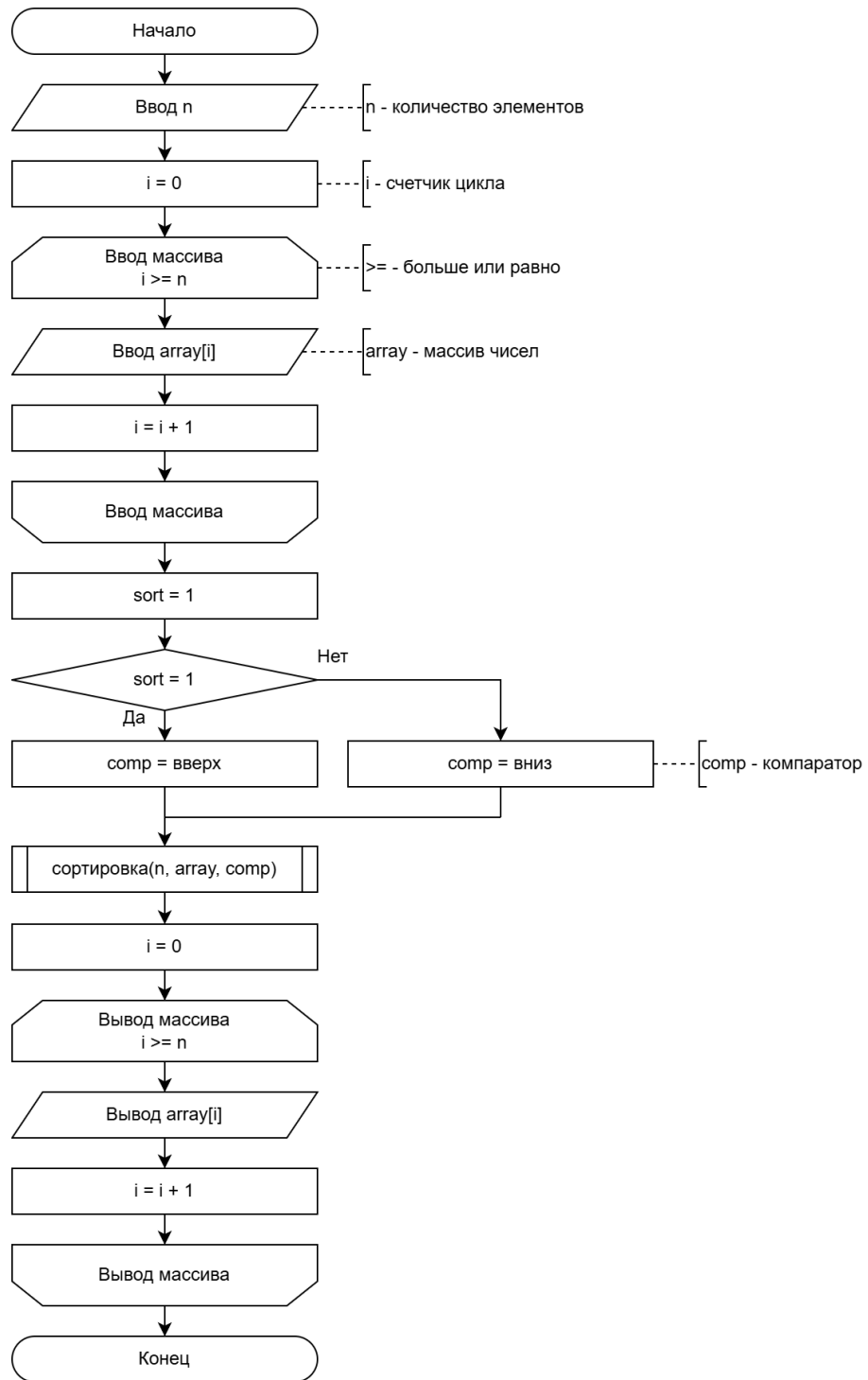


Рисунок 5 – Схема алгоритма поразрядной сортировки

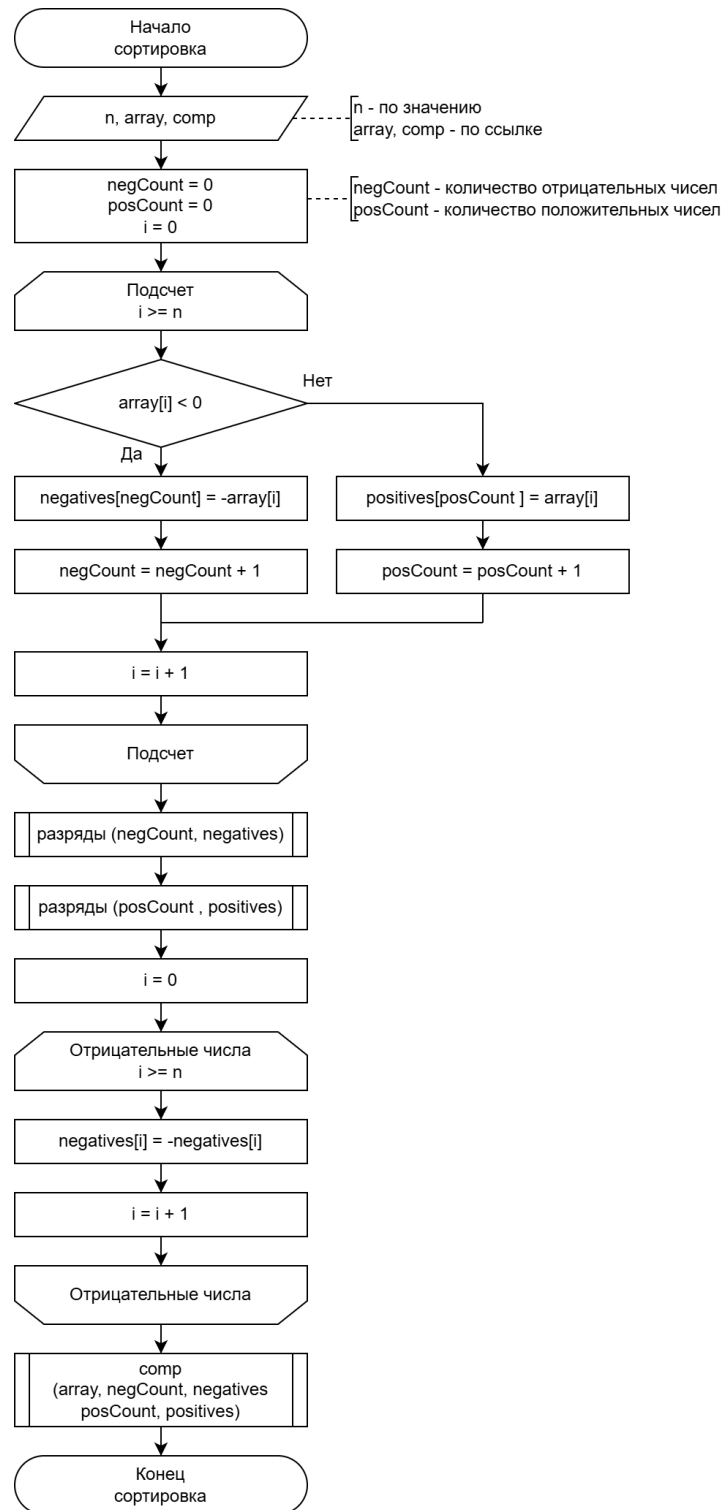


Рисунок 6 – Схема алгоритма подпрограммы «Сортировка»

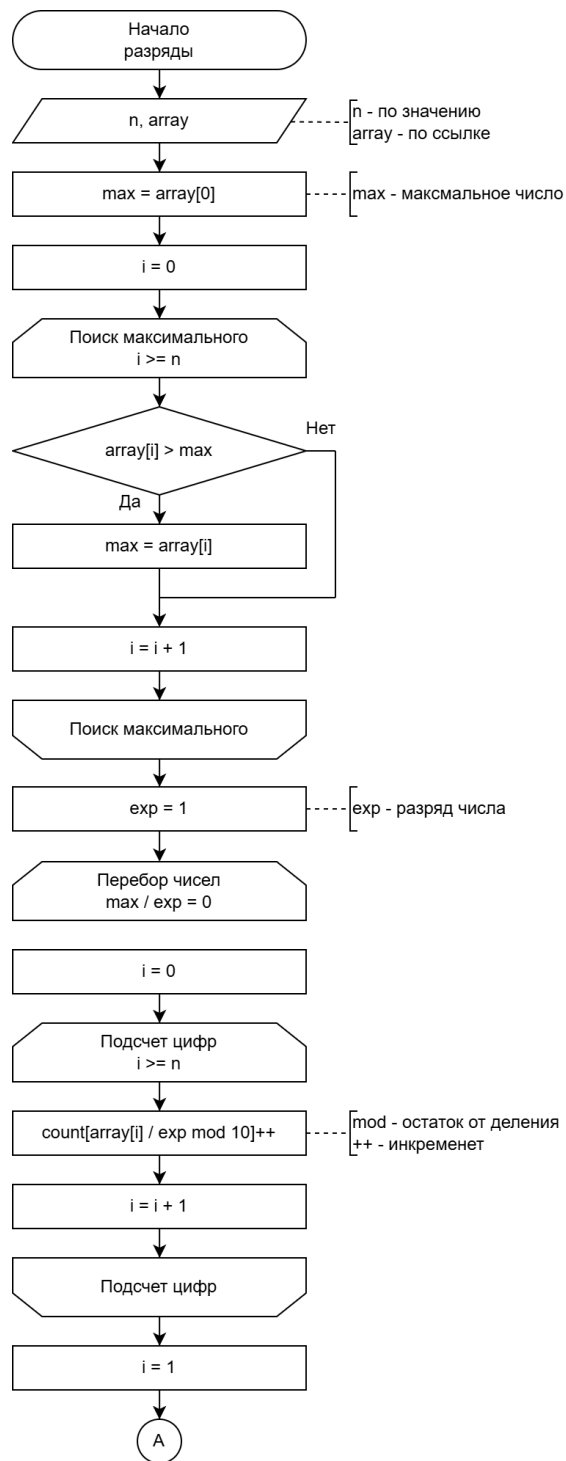


Рисунок 7 – Схема алгоритма подпрограммы «Разряды»

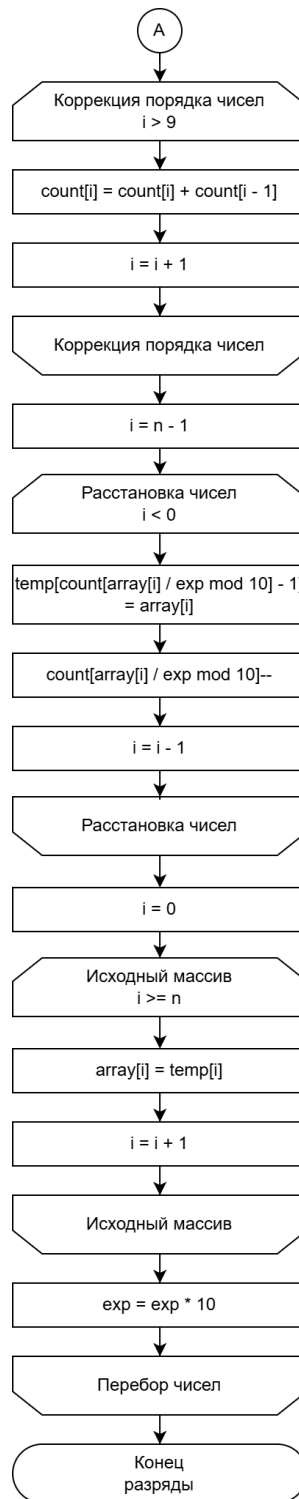


Рисунок 8 – Продолжение схемы алгоритма подпрограммы «Разряды»

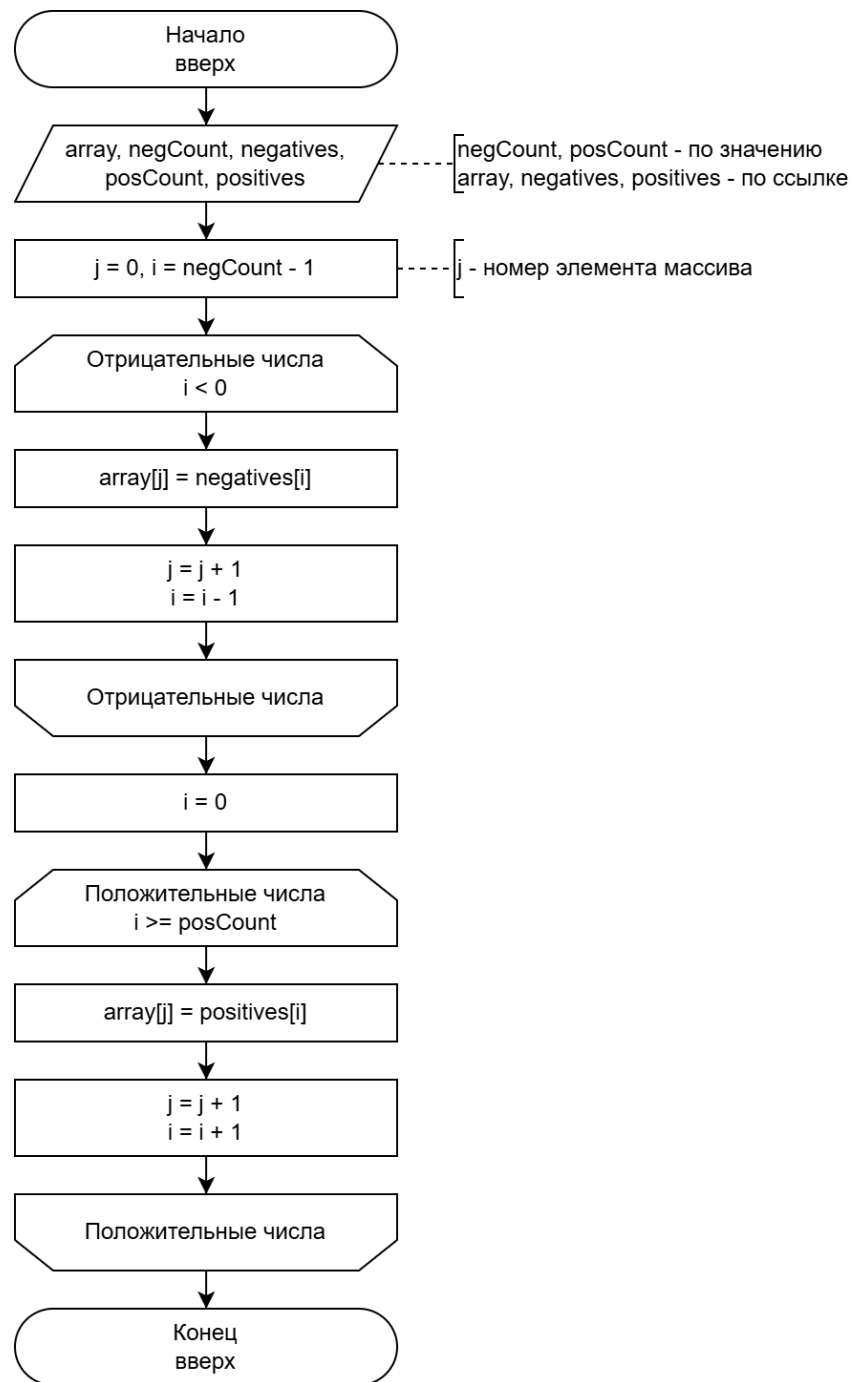


Рисунок 9 – Схема алгоритма подпрограммы «Вверх»

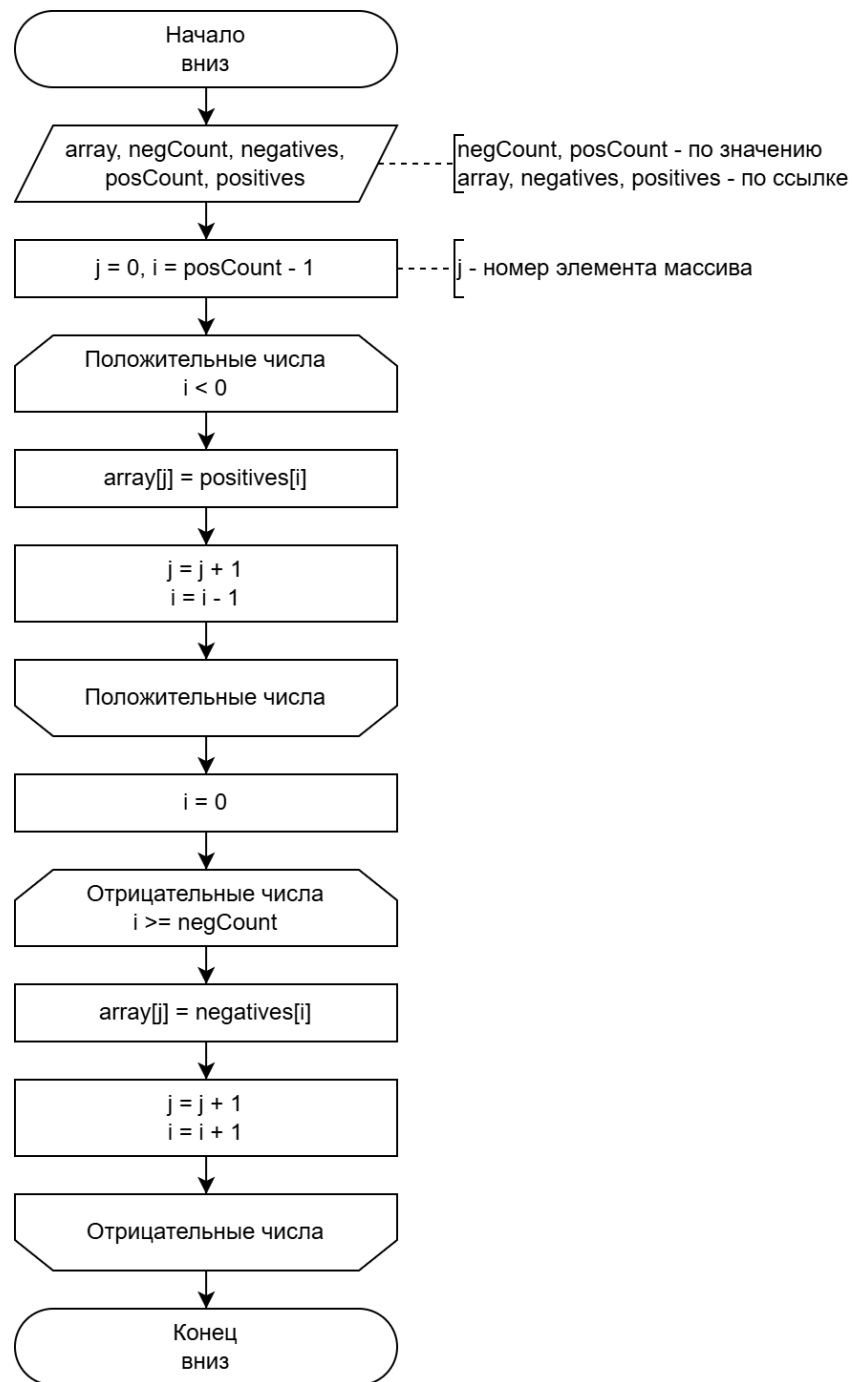


Рисунок 10 – Схема алгоритма подпрограммы «Вниз»

```

#include <stdio.h>
#include <stdlib.h>
void radixSort(int n, int array[]) {
    int max = array[0];
    for (int i = 1; i < n; i++) {
        if (array[i] > max) max = array[i];
    }
}

```

```

for (int exp = 1; max / exp > 0; exp *= 10) {
    int* temp = (int*)calloc(n, sizeof(int));
    int count[10] = { 0 };
    for (int i = 0; i < n; i++) {
        count[array[i] / exp % 10]++;
    }
    for (int i = 1; i < 10; i++) {
        count[i] += count[i - 1];
    }
    for (int i = n - 1; i >= 0; i--) {
        temp[count[array[i] / exp % 10] - 1] = array[i];
        count[array[i] / exp % 10]--;
    }
    for (int i = 0; i < n; i++) {
        array[i] = temp[i];
    }
}

void sort(int n, int array[],
void(*comparator)(int array[],
    int negCount, int negatives[], int posCount, int positives[])) {
    int* negatives = (int*)calloc(n, sizeof(int));
    int* positives = (int*)calloc(n, sizeof(int));
    int negCount = 0, posCount = 0;
    for (int i = 0; i < n; i++) {
        if (array[i] < 0) {
            negatives[negCount++] = -array[i];
        } else {
            positives[posCount++] = array[i];
        }
    }
    radixSort(negCount, negatives);
    radixSort(posCount, positives);
    for (int i = 0; i < negCount; i++) {
        negatives[i] = -negatives[i];
    }
    comparator(array, negatives, negCount, positives, posCount);
    free(negatives);
    free(positives);
}

```

```

void comparatorUp(int array[], int negCount, int negatives[],
    int posCount, int positives[]) {
    int j = 0;
    for (int i = negCount - 1; i >= 0; i--) {
        array[j++] = negatives[i];
    }
    for (int i = 0; i < posCount; i++) {
        array[j++] = positives[i];
    }
}

void comparatorDown(int array[], int negCount, int negatives[],
    int posCount, int positives[]) {
    int j = 0;
    for (int i = posCount - 1; i >= 0; i--) {
        array[j++] = positives[i];
    }
    for (int i = 0; i < negCount; i++) {
        array[j++] = negatives[i];
    }
}

int main() {
    FILE* input = fopen("../input.txt", "r");
    FILE* output = fopen("../output.txt", "w");
    int n = 0;
    fscanf_s(input, "%d", &n);
    int* array = (int*)calloc(n, sizeof(int));
    for (int i = 0; i < n; i++) fscanf_s(input, "%d", &array[i]);
    fclose(input);
    void(*comparator)(int array[], int negCount,
        int negatives[], int posCount, int positives[]);
    comparator = comparatorUp;
    sort(n, array, comparator);
    for (int i = 0; i < n; i++) {
        fprintf(output, "%d ", array[i]);
    }
    fclose(output);
    free(array);
    return 0;
}

```

Вывод

В ходе выполнения лабораторной работы были изучены алгоритмы сортировки подсчётом и поразрядной сортировки по младшим разрядам. Также были изучены принципы работы с текстовыми файлами путём решения предложенных задач.