

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

Отчёт по лабораторной работе №5

по дисциплине

«Информатика»

«Представление вещественных чисел в формате с плавающей точкой.»

Разработал студент гр. ИВТб-1301-05-00 _____ /Черкасов А. А./
(подпись)

Проверил доцент кафедры ЭВМ _____ /Коржавина А.С./
(подпись)

Киров

2024

Цель работы

Цель работы: закрепить на практике знания форматах представления числовой информации. Написать программы, решающие описанные ниже задачи. Программы должны работать без ошибок на любых наборах входных данных.

Задания

1. Представить число в формате с плавающей точкой в n-разрядной сетке. Формат аналогичен IEEE 754.

Формат Ввода.

В одной строке вещественное число в десятичной системе счисления, разрядность сетки, число разрядов мантииссы.

Формат Вывода.

Строка, отображающая введенное число в формате с плавающей точкой.

Ввод	Вывод
-1.25 16 10	1011110100000000

2. Представить число в формате с плавающей точкой в n-разрядной сетке. Нормализация мантииссы дробная, формат с порядком, последовательность отображения — знак, мантиисса, порядок.

Формат Ввода.

В одной строке вещественное число в десятичной системе счисления, разрядность сетки, число разрядов мантииссы.

Формат Вывода.

Строка, отображающая введенное число в формате с плавающей точкой.

Ввод	Вывод
-1.25 16 10	1101000000000001

3. Представить число в формате с плавающей точкой в n-разрядной сетке. Нормализация мантиссы дробная, формат с характеристикой, последовательность отображения – знак, мантисса, характеристика.

Формат Ввода.

В одной строке вещественное число в десятичной системе счисления, разрядность сетки, число разрядов мантиссы.

Формат Вывода.

Строка, отображающая введенное число в формате с плавающей точкой.

Ввод	Вывод
-1.25 16 10	1101000000010001

Решение

Схема алгоритма представлена на Рисунках 1.1, 1.2 и 1.3.

Код программы приведён в Приложении А1.

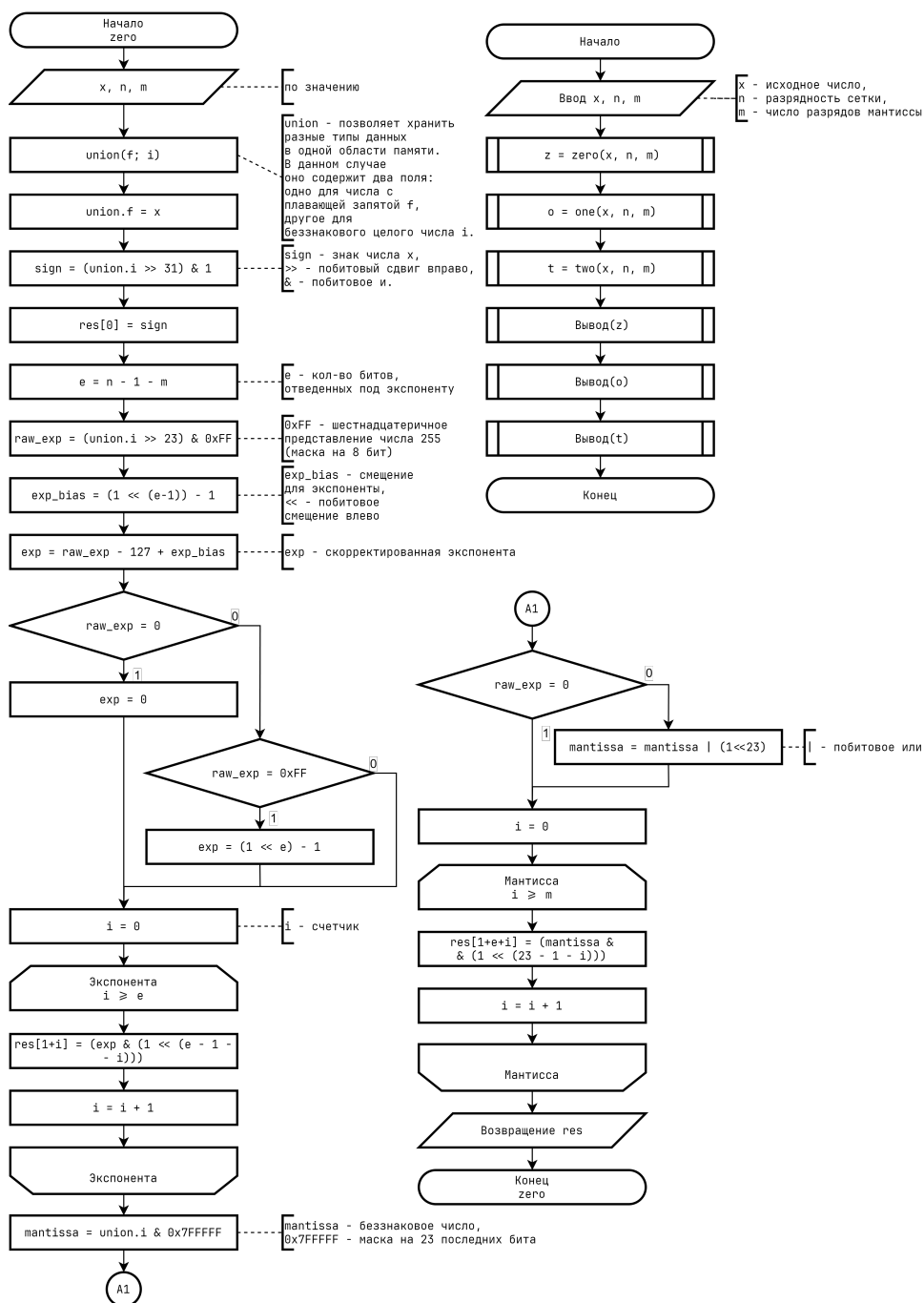


Рисунок 1.1 - Схема алгоритма программы.

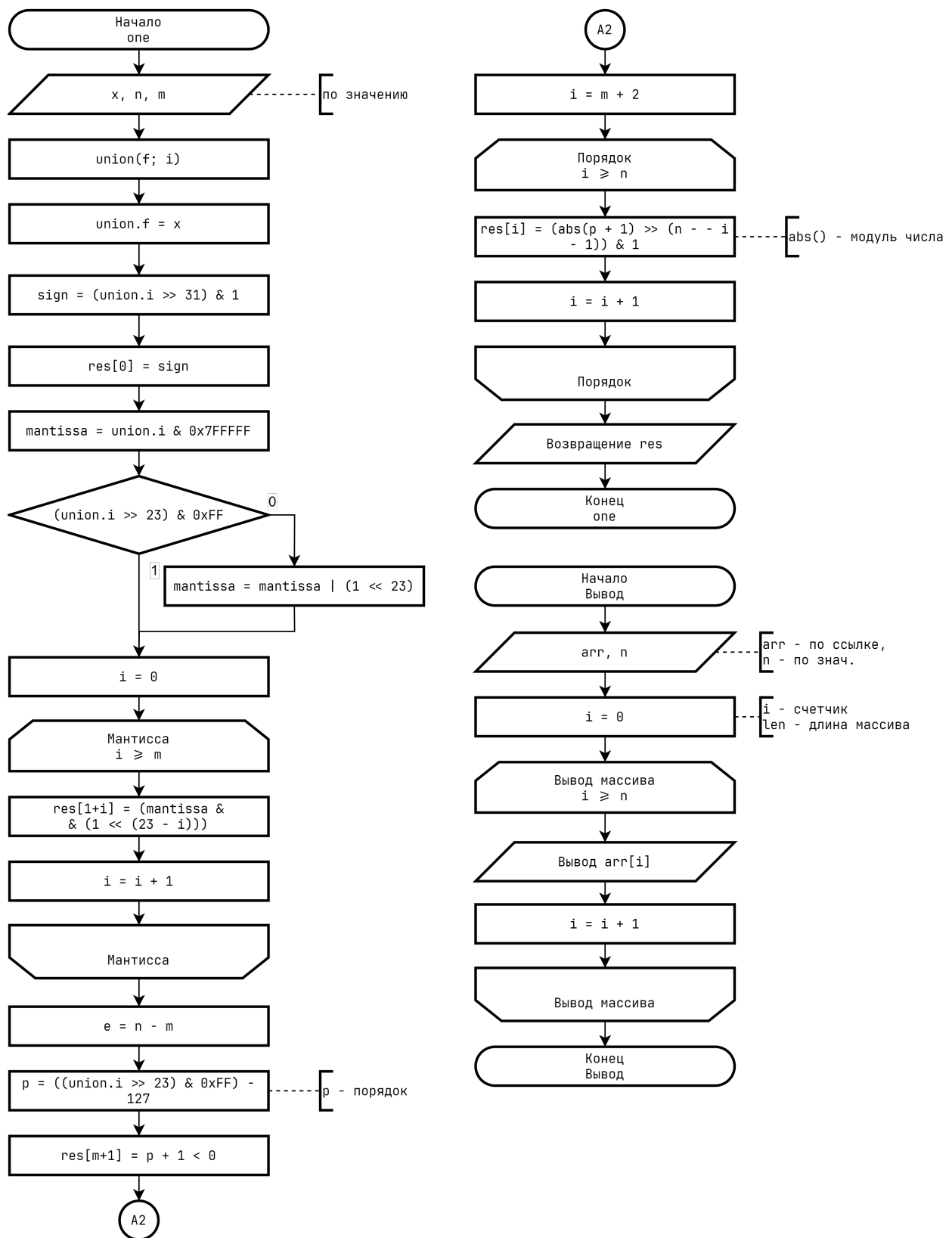


Рисунок 1.2 - Схема алгоритма программы.

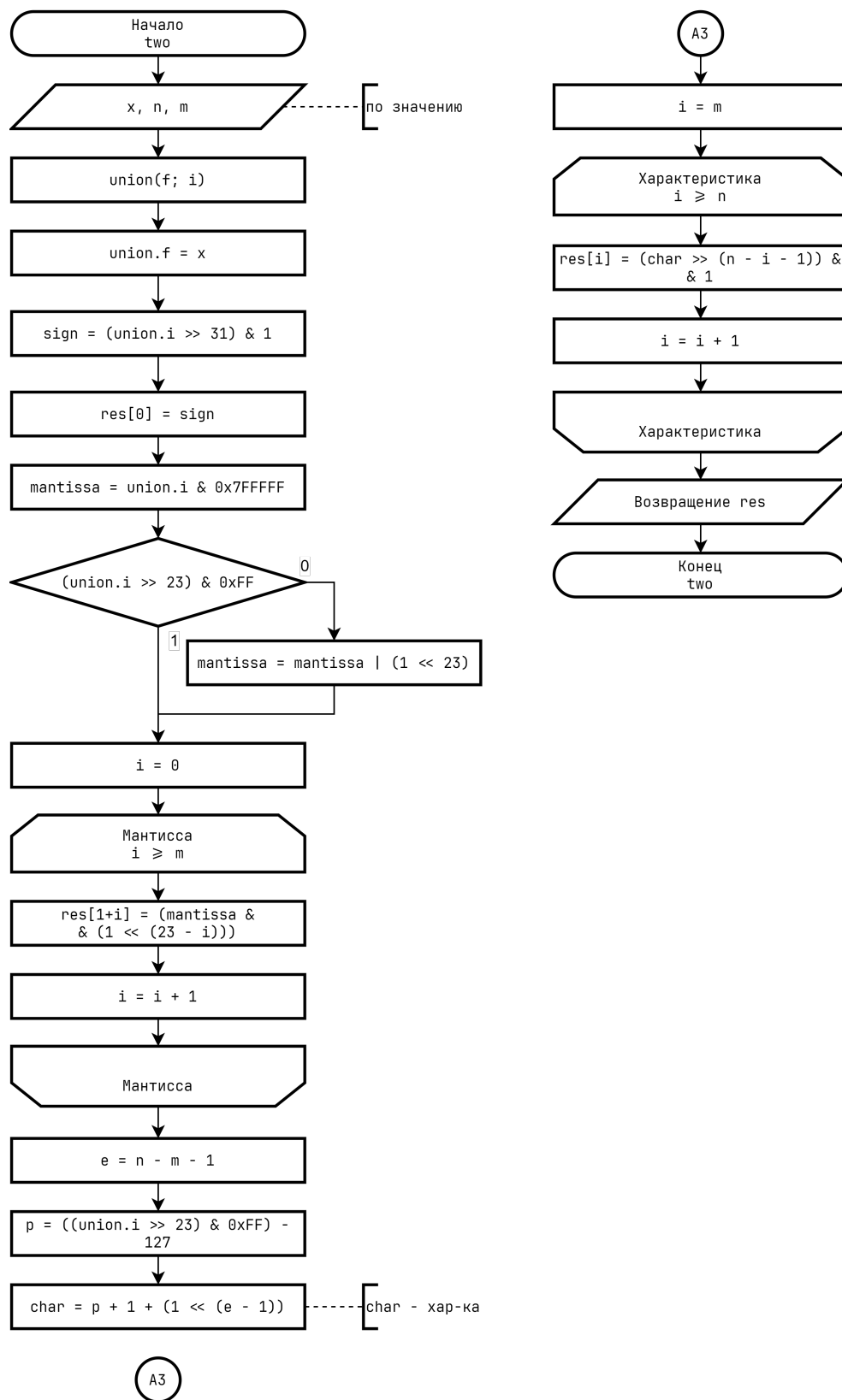


Рисунок 1.3 - Схема алгоритма программы.

Вывод

В результате работы были разработаны программы для преобразования вещественных чисел в форматы с плавающей точкой по стандарту. Программы корректно обрабатывают входные данные и реализуют различные форматы представления. Тестирование подтвердило их соответствие требованиям.

Приложение A1

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

char* zero(float x, int n, int m) {
    char* res = malloc(n + 1);
    if (!res) return NULL;
    union {
        float f;
        unsigned int i;
    } u;
    u.f = x;
    // Знак
    int sign = (u.i >> 31) & 1;
    res[0] = sign ? '1' : '0';
    int e = n - 1 - m;
    // Извлечение экспоненты
    int raw_exp = (u.i >> 23) & 0xFF; // 8 бит для экспоненты
    int exp_bias = (1 << (e - 1)) - 1;
    int exp = raw_exp - 127 + exp_bias;
    // Обработка переполнения/недополнения экспоненты
    if (raw_exp == 0) { // Денормализованное число
        exp = 0;
    } else if (raw_exp == 0xFF) { // Бесконечность / не число
```

```

        exp = (1 << e) - 1;
    }
    // Экспонента
    for (int i = 0; i < e; i++) {
        res[1 + i] = (exp & (1 << (e - 1 - i))) ? '1' : '0';
    }
    // Мантисса
    unsigned int mantissa = u.i & 0x7FFFFFFF; // Последние 23 бита
    if (raw_exp != 0) { // Добавляем скрытый бит для нормализованных чисел
        mantissa |= (1 << 23);
    }
    for (int i = 0; i < m; i++) {
        res[1 + e + i] = (mantissa & (1 << (23 - 1 - i))) ? '1' : '0';
    }
    res[n] = '\\0';
    return res;
}

char* one(float x, int n, int m) {
    char* res = malloc(n + 1);
    if (!res) return NULL;
    union {
        float f;
        unsigned int i;
    } u;
    u.f = x;
    // Знак
    int sign = (u.i >> 31) & 1;
    res[0] = sign ? '1' : '0';
    // Мантисса
    unsigned int mantissa = u.i & 0x7FFFFFFF;
    if ((u.i >> 23) & 0xFF) {
        mantissa |= (1 << 23);
    }
}

```



```

}
for (int i = 0; i < m; i++) {
    res[1 + i] = (mantissa & (1 << (23 - i))) ? '1' : '0';
}
// Порядок
int e = n - m;
int exp = (u.i >> 23) & 0xFF;
int p = exp - 127;
res[m + 1] = p + 1 < 0 ? '1' : '0';
for (int i = m + 2; i < n; i++) {
    res[i] = (abs(p + 1) >> (n - i - 1)) & 1 ? '1' : '0';
}
res[n] = '\\0';
return res;
}

```

```

char* two(float x, int n, int m) {
    char* res = malloc(n + 1);
    if (!res) return NULL;
    union {
        float f;
        unsigned int i;
    } u;
    u.f = x;
    // Знак
    int sign = (u.i >> 31) & 1;
    res[0] = sign ? '1' : '0';
    // Мантисса
    unsigned int mantissa = u.i & 0x7FFFFFFF;
    if ((u.i >> 23) & 0xFF) {
        mantissa |= (1 << 23);
    }
    for (int i = 0; i < m + 1; i++) {

```

```

        res[1 + i] = (mantissa & (1 << (23 - i))) ? '1' : '0';
    }
    // Xap-ka
    int e = n - m - 1;
    int exp = (u.i >> 23) & 0xFF;
    int p = exp - 127;
    int charc = p + 1 + (1 << (e - 1));
    for (int i = m; i < n; i++) {
        res[i] = (charc >> (n - i - 1)) & 1 ? '1' : '0';
    }
    res[n] = '\\0';
    return res;
}

int main() {
    float x;
    int n, m;
    scanf("%f %d %d", &x, &n, &m);
    printf("0. %s\\n", zero(x, n, m));
    printf("1. %s\\n", one(x, n, m));
    printf("2. %s\\n", two(x, n, m));
    return 0;
}

```