

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 №4

по дисциплине

«Информатика»

«Форматы представления числовой информации. Представление целых чисел.»

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.

Задания

1. Представление беззнакового числа в n -разрядной сетке.

На вход подаётся целое число в десятичной системе и разрядность сетки. Если число отрицательное или превышает допустимое значение для данной разрядности, выводится ошибка.

Формат ввода.

В одной строке через пробел вводятся число и разрядность сетки.

Формат вывода.

Вывести строку, представляющую число в n -разрядной сетке или сообщение об ошибке.

Ввод	Вывод
1 8	00000001
255 8	11111111
256 8	ошибка

2. Представление числа в прямом коде.

На вход подаётся целое число и разрядность сетки. Если число выходит за пределы представления в данной разрядности, выводится ошибка. Прямой код формируется как для положительных, так и отрицательных чисел.

Формат ввода.

В одной строке через пробел вводятся число и разрядность сетки.

Формат вывода.

Вывести строку, представляющую число в прямом коде или сообщение об ошибке.

Ввод	Вывод
-1 8	10000001
25 8	00011001
-128 8	ошибка

3. Представление числа в дополнительном коде.

На вход подаётся целое число и разрядность сетки. Если число выходит за пределы представления, выводится ошибка.

Формат ввода.

В одной строке через пробел вводятся число и разрядность сетки.

Формат вывода.

Вывести строку, представляющую число в дополнительном коде или сообщение об ошибке.

Ввод	Вывод
-1 8	11111111
25 8	00011001
225 8	ошибка

4. Представление числа в обратном коде.

На вход подаётся целое число и разрядность сетки. Если число выходит за пределы представления, выводится ошибка.

Формат ввода.

В одной строке через пробел вводятся число и разрядность сетки.

Формат вывода.

Вывести строку, представляющую число в обратном коде или сообщение об ошибке.

Ввод	Вывод
-1 8	11111110
25 8	00011001
-168 8	ошибка

5. Расстояние по Хэммингу двух дополнительных кодов.

На вход подаются два числа и разрядность сетки. Преобразуются числа в дополнительные коды и вычисляется количество отличающихся позиций.

Формат ввода.

В одной строке вводятся два числа и разрядность сетки.

Формат вывода.

Вывести расстояние по Хэммингу или сообщение об ошибке.

Ввод	Вывод
-1 25 8	7
-1 225 8	ошибка

Решение

Схема алгоритма представлена на Рисунках 1.1 и 1.2.

Код программы приведён в Приложении А1.

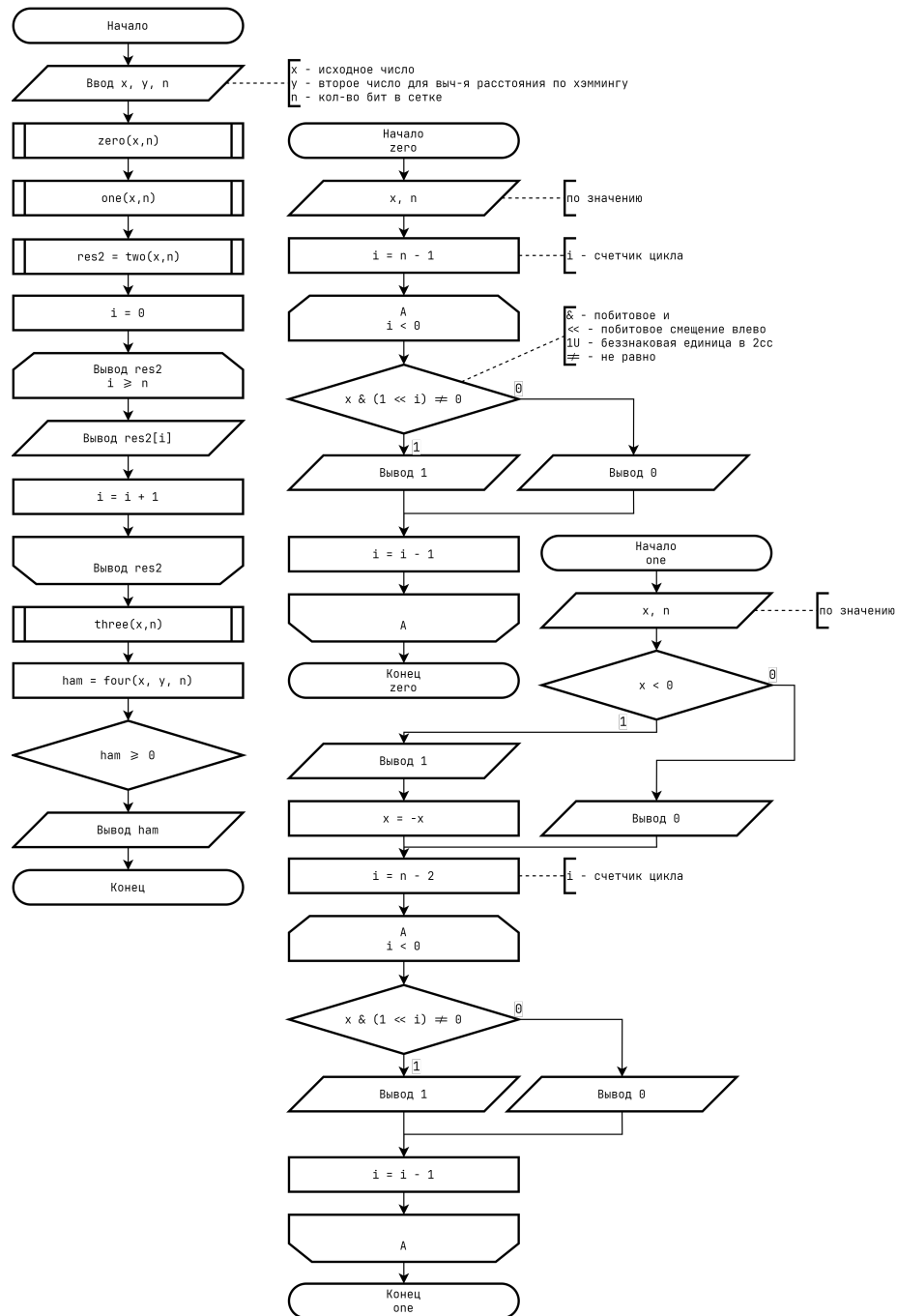


Рисунок 1.1 - Схема алгоритма программы.

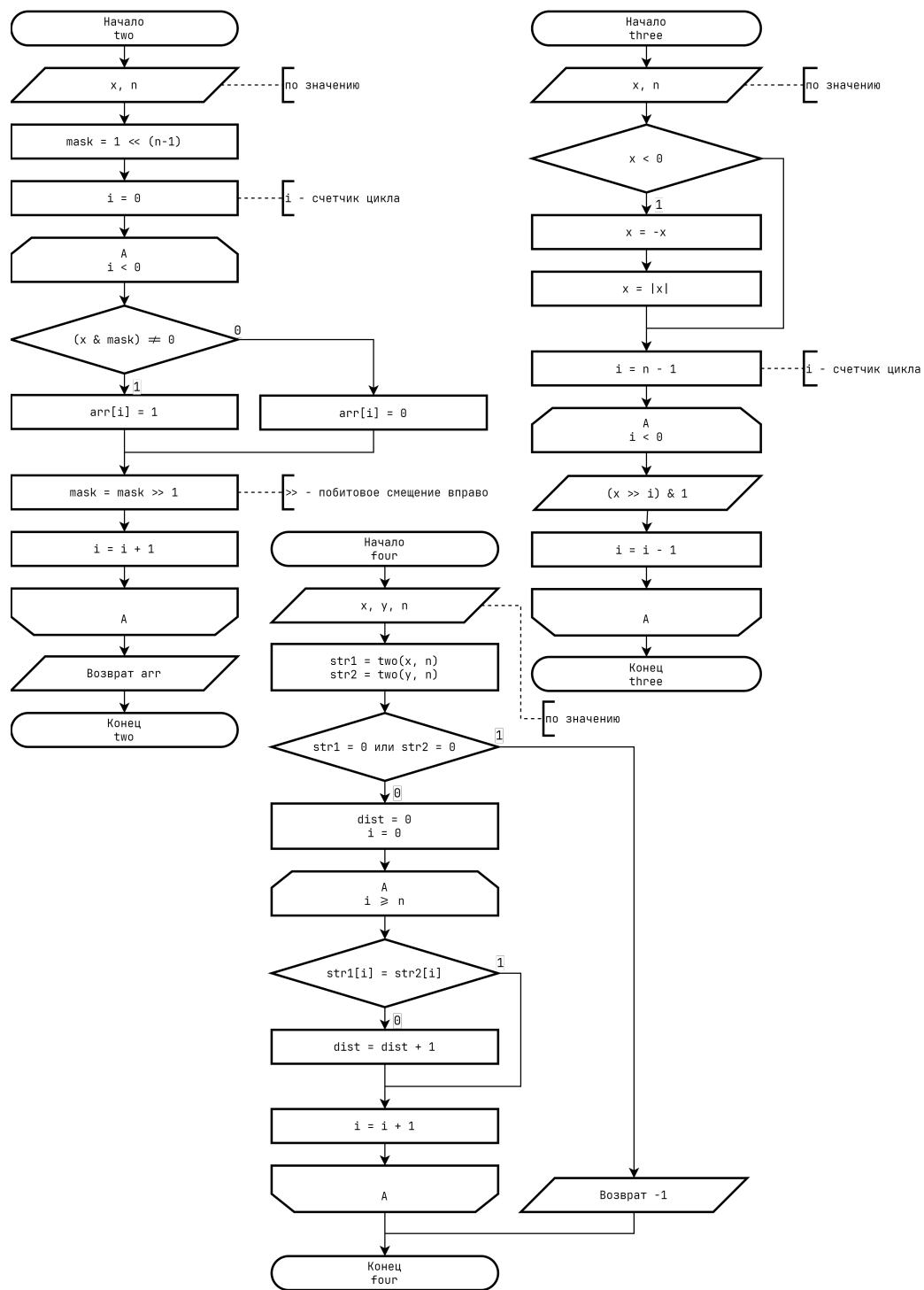


Рисунок 1.2 - Схема алгоритма программы.

Вывод

В ходе работы удалось достичь цели — закрепить знания о представлении чисел. Были разработаны и проверены программы, которые работают правильно. Работа помогла лучше понять тему и улучшить навыки программирования.

Приложение A1

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

void print_error(const char* message, int x, int n) {
    printf("ошибка: %s (x = %d, n = %d)\n", message, x, n);
}

void zero(unsigned int x, int n) {
    printf("0. ");
    if (x >= (1 << n)) {
        print_error("число не входит в диапазон представления для сетки", x, n);
        return;
    }
    for (int i = n - 1; i >= 0; i--) {
        putchar((x & (1 << i)) != 0 ? '1' : '0');
    }
    putchar('\n');
}

void one(int x, int n) {
    printf("1. ");
    if (x <= -(1 << (n - 1)) || x >= (1 << (n - 1))) {
        print_error("число не входит в диапазон представления\
в прямом коде для сетки", x, n);
        return;
    }
}
```

```

}
if (x < 0) {
    putchar('1');
    x = -x;
} else {
    putchar('0');
}
for (int i = n - 2; i >= 0; i--) {
    putchar((x & (1 << i)) != 0 ? '1' : '0');
}
printf(" (прямой код)\n");
}

char* two(int x, int n) {
    char *arr = (char*)malloc(sizeof(char) * (n + 1));
    if (x < -(1 << (n - 1)) || x >= (1 << (n - 1))) {
        print_error("число выходит из диапазона представления\
в дополнительном коде для сетки", x, n);
        free(arr);
        return NULL;
    }
    int mask = 1 << (n - 1);
    for (int i = 0; i < n; i++) {
        arr[i] = (x & mask) != 0 ? '1' : '0';
        mask >>= 1;
    }
    arr[n] = '\0';
    return arr;
}

void three(int x, int n) {
    printf("3. ");
    if (x > (1 << (n - 1)) - 1 || x < -((1 << (n - 1)) - 1)) {

```



```

        print_error("число не входит в диапазон представления для сетки", x, n);
        return;
    }
    if (x < 0) {
        x = abs(x);
        x = ~x;
    }
    for (int i = n - 1; i >= 0; i--) {
        putchar(((x >> i) & 1) ? '1' : '0');
    }
    printf(" (обратный код)\n");
}

```

```

int four(int x, int y, int n) {
    char* str1 = two(x, n);
    char* str2 = two(y, n);
    if (str1 == NULL || str2 == NULL){
        return -1;
    }
    int dist = 0;
    for (int i = 0; i < n; i++) {
        if (str1[i] != str2[i]) {
            dist++;
        }
    }
    free(str1);
    free(str2);
    return dist;
}

```

```

int main() {
    int x, y, n;
    scanf("%d %d %d", &x, &y, &n);

```

```

zero(x, n);
one(x, n);
char* two_result = two(x, n);
if (two_result != NULL) {
    printf("2. %s (дополнительный код)\n", two_result);
    free(two_result);
}
three(x, n);
int ham_dist = four(x, y, n);
if (ham_dist >= 0) {
    printf("4. %d (расстояние по Хэммингу)\n", ham_dist);
}
return 0;
}

```