

1. Цель работы

Целью работы является изучение основных управляющих структур программирования и функций

2. Задание

Согласно варианту №3 необходимо:

Вычислить и вывести на экран в виде таблицы значения функции F на интервале X нач, X кон с шагом dx .

Вариант 3

$$F = \begin{cases} ax^2 + bx + c & \text{при } a < 0 \text{ и } c \neq 0 \\ \frac{-a}{x - c} & \text{при } a > 0 \text{ и } c = 0 \\ a(x + c) & \text{в остальных случаях} \end{cases}$$

Рисунок 1 – Задание по варианту

3. Описание созданных функций

Имя: get_num

Назначение: Запрос и проверка числа на корректность

Входные данные:

- -

Выходные данные:

- double x – проверенное число

Побочный эффект: Отсутствует

Тестовые данные:

Вход	Выход
5g	Неверный ввод
5	5

Прототип: double get_num()

Псевдокод

Запросить число

Проверить на корректность

Вернуть число

Блок-схема –

Имя: request_data

Назначение: Запрос данных промежутка

Входные данные:

- -

Выходные данные:

- struct x_int – структура из трёх элементов промежутка

Побочный эффект: Отсутствует

Тестовые данные:

Вход	Выход
-5 5 100	Шаг больше интвервала
-5 5 1	-5 5 1

Прототип: VARS request_data()

Псевдокод

Запросить числа

Добавить в структуру

Вернуть структуру

Блок-схема –

Имя: request_data2

Назначение: Запрос данных коэффициентов

Входные данные:

- -

Выходные данные:

- struct a_b_c – структура из трёх коэффициентов

Побочный эффект: Отсутствует

Тестовые данные:

Вход	Выход
0 g 4	Некоректное число
-5 5 1	-5 5 1

Прототип: VARS request_data2()

Псевдокод

Запросить числа

Добавить в структуру

Вернуть структуру

Блок-схема –

Имя: calc

Назначение: Вычисление функции

Входные данные:

- const double x - Икс для расчёта функции
- struct a_b_c – Структура коэффициентов
- bool flag – Флаг для обозначения деления на нуль

Выходные данные:

- y – Результат

Побочный эффект: Отсутствует

Тестовые данные:

Вход	Выход
0, (-5 5 1), (1 2 0), 0	Деление на нуль
1, (-5 5 1), (1 2 0), 0	-2

Прототип: double calc(const double x, VARS a_b_c, bool & flag)

Псевдокод

Установить флаг деления на нуль false

Проверить на условия кусочной функции

Расчитать значения для данного случая

*в случае деления на нуль установить флаг true

Блок-схема

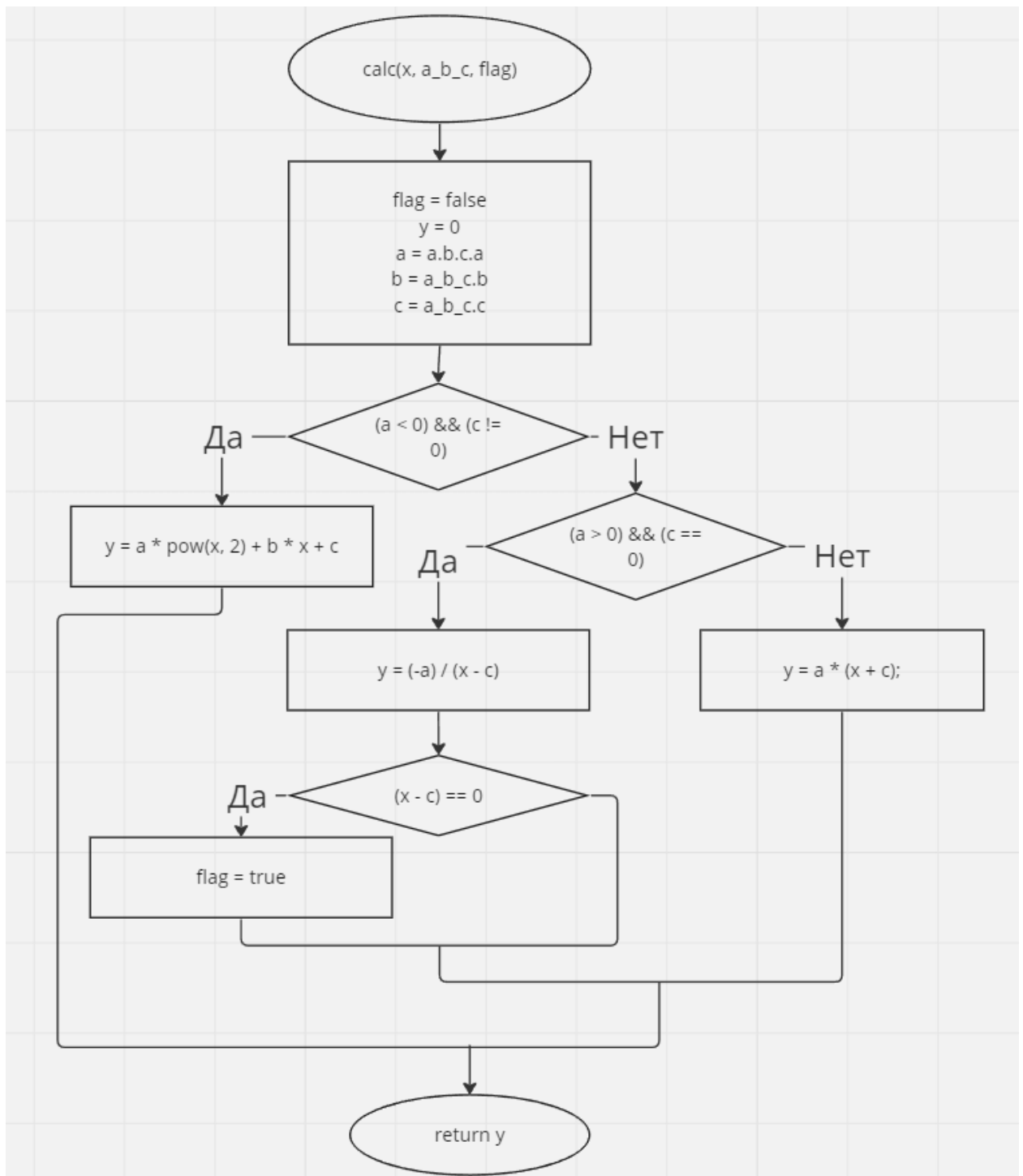


Рисунок 2 – блок схема calc

Имя: print_table

Назначение: Вычисление функции

Входные данные:

- const double x - Икс для расчёта функции
- struct a_b_c – Структура коэффициентов
- bool flag – Флаг для обозначения деления на нуль

Выходные данные:

- y – Результат

Побочный эффект: Отсутствует

Тестовые данные:

Вход	Выход
(-5 5 1) (1 2 0)	Таблица значений на 10 строк
(-5 5 2) (1 2 0)	Таблица значений на 5 строк

Прототип: void print_table(VARS x_int, VARS a_b_c)

Псевдокод

Напечатать шапку таблицы

Запустить цикл на весь интервал

 Печать строки на каждый x

Напечатать конец таблицы

Блок-схема

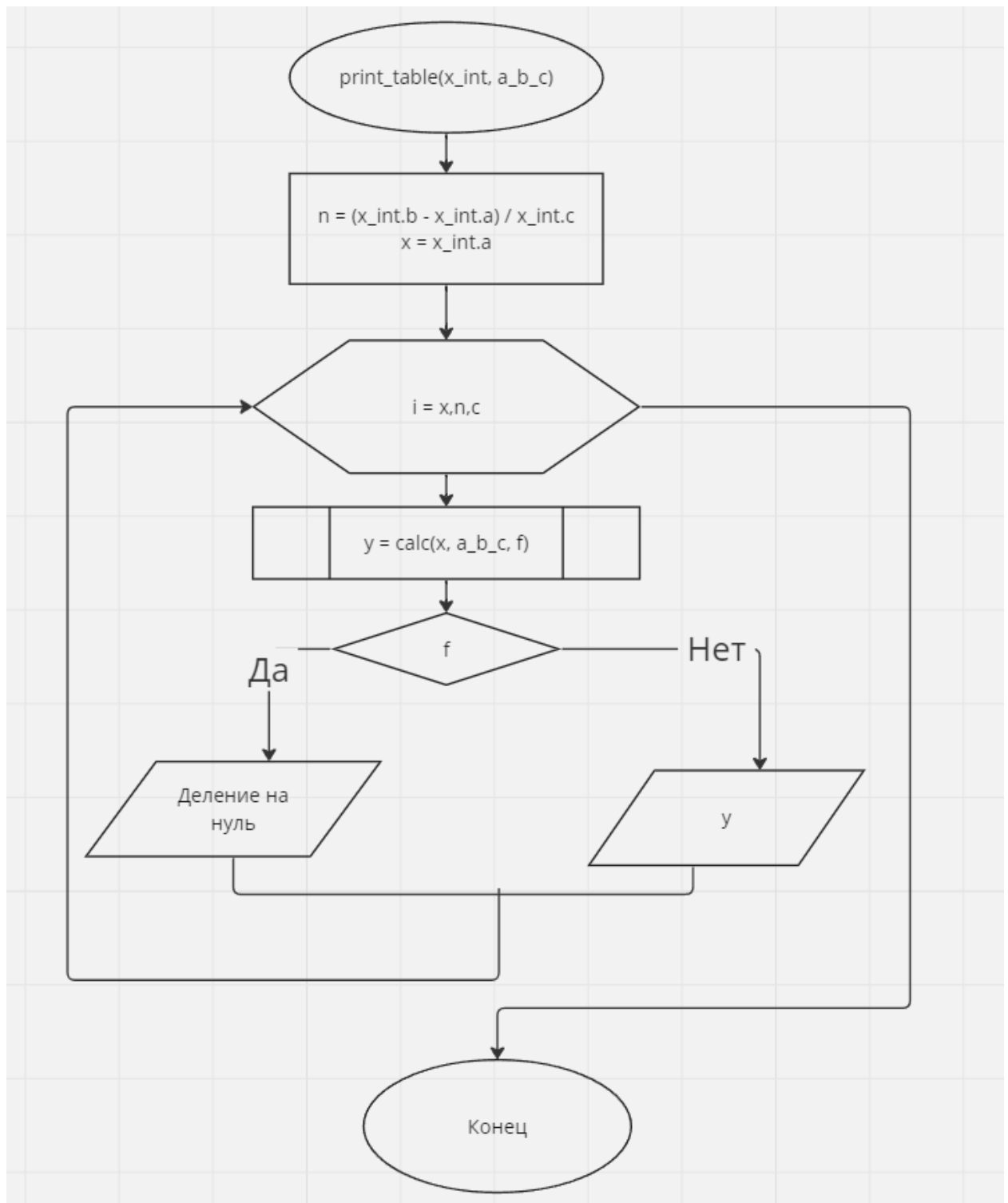


Рисунок 3 – блок схема print_table

4. Текст программы

```

#include<iostream>
#include<cmath>

using namespace std;

struct VARS // Структура для промежутка/коэффициентов
{
    double a;

```

```

        double b;
        double c;
};

double get_num() // Запрос и проверка числа на корректность
{
    double x;

    cin >> x;
    while (cin.fail() || (cin.peek() != '\n')) // Проверка на корректность
    {
        cin.clear(); // Очищение флага ошибки
        cin.ignore(numeric_limits<streamsize>::max(), '\n'); // Очистка буфера
запроса
        cout << "Повторите ввод: ";
        cin >> x;
    }

    return x;
}

VAR request_data() // Функция запроса промежутка
{
    VAR x_int;

    double x;

    cout << "Введите левую границу: " << endl;
    cout << "Xmin = ";
    x = get_num();

    x_int.a = x; // Добавление Xmin в структуру

    cout << "Введите правую границу: " << endl;
    cout << "Xmax = ";
    x = get_num();
    while (x <= x_int.a) // Проверка на ограничения(больше левой границы)
    {
        x = get_num();
    }
    x_int.b = x; // Добавление Xmax в структуру

    cout << "Введите шаг интервала: " << endl;
    cout << "dx = ";
    x = get_num();
    while ((x <= 0) || (x > (x_int.b - x_int.a))) // Проверка на корректность и
ограничения
    {
        x = get_num();
    }
    x_int.c = x; // Добавление dx в структуру

    return x_int;
}

VAR request_data_2() // Функция запроса коэффициентов
{
    VAR a_b_c;

    double x;

    cout << "Введите коэффициент a: " << endl;
    cout << "a = ";

    x = get_num();

```

```

a_b_c.a = x; // Добавление a в структуру

cout << "Введите коэффициент b: " << endl;
cout << "b = ";

x = get_num();
a_b_c.b = x; // Добавление b в структуру

cout << "Введите коэффициент c: " << endl;
cout << "c = ";

x = get_num();
a_b_c.c = x; // Добавление c в структуру

return a_b_c;
}

double calc(const double x, VARS a_b_c, bool & flag) // Рассчёт функции
{
    flag = false;
    double y = 0;

    double a = a_b_c.a;
    double b = a_b_c.b;
    double c = a_b_c.c;

    if ((a < 0) && (c != 0)) // Проверка трёх случаев по заданию
    {
        y = a * pow(x, 2) + b * x + c;
    }
    else if ((a > 0) && (c == 0))
    {
        y = (-a) / (x - c);
        if ((x - c) == 0)
        {
            flag = true;
        }
    }
    else
    {
        y = a * (x + c);
    }

    return y;
}

void print_table(VARS x_int, VARS a_b_c) // Функция принта таблицы вычислений на
промежутке
{
    double n = (x_int.b - x_int.a) / x_int.c;
    double x = x_int.a;
    bool f;

    printf("/-----\\n");
    printf("|      x      |      y      |\\n");

    for (x; x <= n; x+= x_int.c)
    {
        calc(x, a_b_c, f);
        if (f) // Печать строк таблицы для каждого x
        {
            printf("|%15.2f|%15s|\\n", x, "Деление на ноль"); // Печать
строки при делении на ноль
        }
        else

```



```

        {
            printf("|%15.2f|%15.2f|\n", x, calc(x, a_b_c, f)); // Обычная
печать
        }
    }

    printf("\n-----/\n");
}

int main()
{
    setlocale(LC_ALL, "rus");

    VARS x_int = request_data(); // Запрос промежутка
    VARS a_b_c = request_data_2(); // Запрос коэффициентов

    print_table(x_int, a_b_c); // Принт таблицы вычислений
}

```

5. Пример работы программы

```

Введите левую границу:
Xmin = -5
Введите правую границу:
Xmax = 5
Введите шаг интервала:
dx = 1
Введите коэффициент а:
a = 1
Введите коэффициент b:
b = 2
Введите коэффициент с:
c = 0
/-----\
|      x      |      y      |
|      -5,00  |      0,20  |
|      -4,00  |      0,25  |
|      -3,00  |      0,33  |
|      -2,00  |      0,50  |
|      -1,00  |      1,00  |
|       0,00  | Деление на нуль |
|       1,00  |      -1,00 |
|       2,00  |      -0,50 |
|       3,00  |      -0,33 |
|       4,00  |      -0,25 |
|       5,00  |      -0,20 |
|       6,00  |      -0,17 |
|       7,00  |      -0,14 |
|       8,00  |      -0,12 |
|       9,00  |      -0,11 |
|      10,00  |      -0,10 |
\-----/

```

Рисунок 4 – Результат работы программы

Полученные данные совпадают с действительными

6. Анализ результатов и выводы

В процессе лабораторной работы были изучены основные управляющие структуры программирования и функций

Достоинства программы:

- Проверка данных на корректность
- Возможность использования подпрограмм в других разработках
- Отлов деления на нуль

Недостатки:

- Не было обнаружено