1. **Цель работы**

Целью работы является изучение основных управляющих структур программирования и функций

1. **Задание**

Согласно варианту №3 необходимо:  
Вычислить и вывести на экран в виде таблицы значения функции 𝐹 на интервале 𝑋нач, 𝑋кон с шагом 𝑑𝑥.

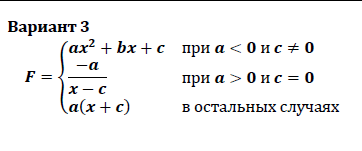


Рисунок 1 – Задание по варианту

1. **Описание созданных функций**

**Имя:** get\_num

**Назначение:** Запрос и проверка числа на корректность

**Входные данные:**

* -

**Выходные данные:**

* double x **–** проверренное 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**

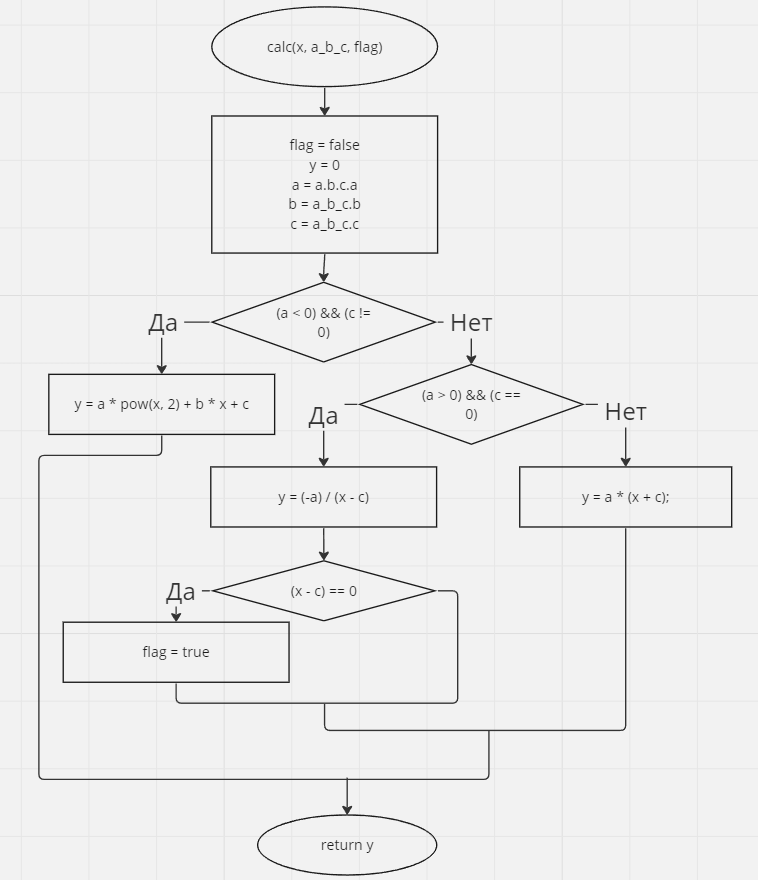
****

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:**

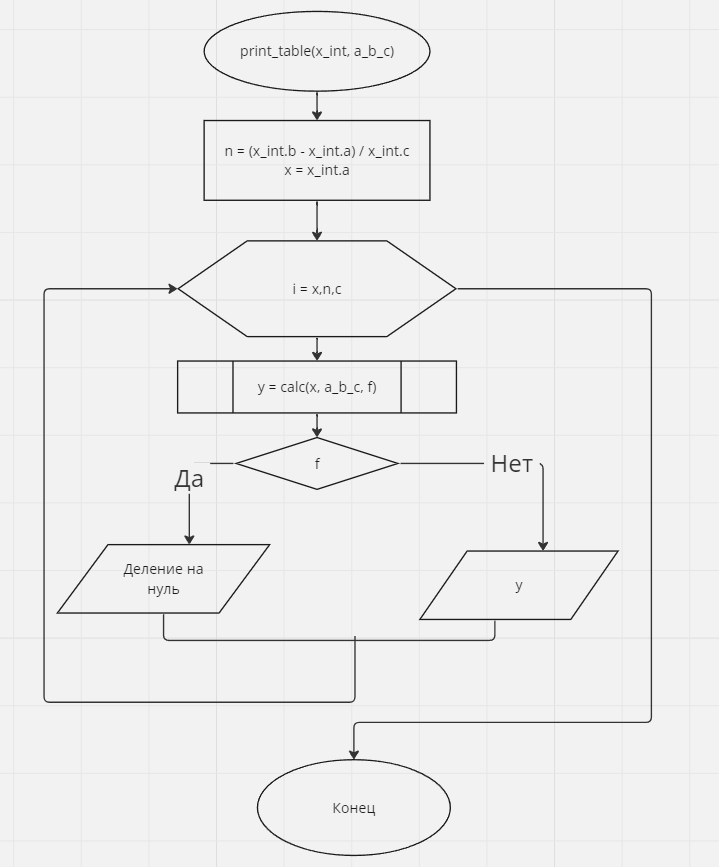
|  |  |
| --- | --- |
| Вход | Выход |
| (-5 5 1) (1 2 0) | Таблица значений на 10 строк |
| (-5 5 2) (1 2 0) | Таблица значений на 5 строк |

**Прототип:** void print\_table(VARS x\_int, VARS a\_b\_c)

**Псевдокод**

Напечатать шапку таблицу  
Запустить цикл на весь интервал  
 Печать строки на каждый x  
Напечатать конец таблицы

**Блок-схема**

****

**Рисунок 3 – блок схема print\_table**

1. **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;

}

VARS request\_data() // Функция запроса промежутка

{

VARS 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;

}

VARS request\_data\_2() // Функция запроса коэффициентов

{

VARS a\_b\_c;

double x;

cout << "Введите коэффициент а: " << endl;

cout << "a = ";

x = get\_num();

a\_b\_c.a = x; // Добавление а 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 нуль"); // Печать строки при делении на нуль

}

else

{

printf("|%15.2f|%15.2f|\n", x, calc(x, a\_b\_c, f)); // Обычная печать

}

}

printf("\\-------------------------------/\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

}

1. **Пример работы программы**

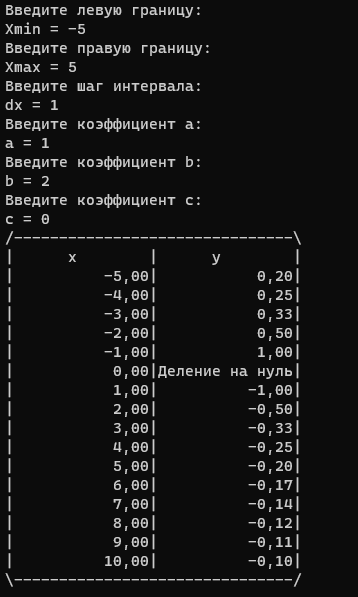


Рисунок 4 – Результат работы программы

Полученные данные совпадают с действительными

1. **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