1. Лабораторная работа № 2

«Программирование алгоритмов циклической структуры»

* 1. Цель работы

Получение навыков программирования алгоритмов циклической структуры на языке С. Исследование эффективности применения различных видов циклов в задаче табулирования функции.

## Постановка задачи

Получить навыки работы с циклическими структурами for и while.

* 1. Ход выполнения работы

Вариант выполнения задания – 18.

* + 1. Описание алгоритма решения задачи
       1. Математическое обоснование

Требуется найти область определения и область значений для функции, представленной Формулой 2.1.

Формула 2.1

Для первой ветки определены такие значения: условия для ввода x –

, область значений – .

Для второй ветки: область определения – , область значений – .

Для третьей ветки: область определения –, область значений – .

* + - 1. Определение входных и выходных данных

Пользователь вводит пять вещественных чисел – a, b, xb, xe, dx.

Программа выводит таблицу «Значение аргумента – Значение функции» или ошибку, сообщающую причину невозможности работы программы.

* + - 1. Структурная схема алгоритма

Структурная схема основного алгоритма продемонстрирована на Рисунке 2.1.

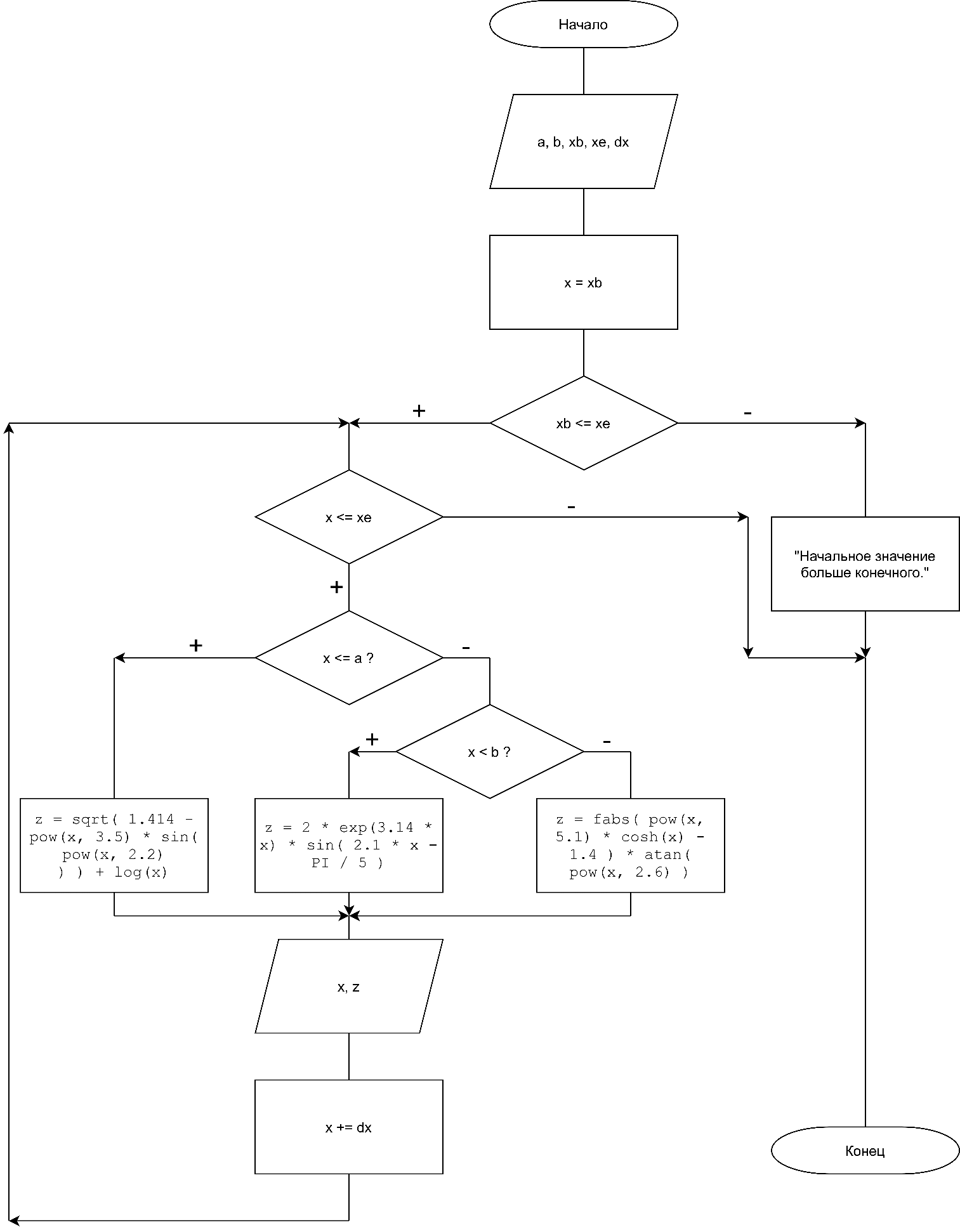


Рисунок 2.1 – Структурная схема основного алгоритма

* + 1. Текст программы (ЯП C, операционная система GNU/Linux Ubuntu)

#include <stdio.h>

#include <math.h>

#define PI 3.14

int main ()

{

float a, b, x, z, xb, xe, dx;

printf(" (!) Внимание (!) \n");

printf("Если значение функции невычисляемо, в ячейке будет отображено число 666.666 .\n");

// Ввод значений

printf("Введите значение a: "); scanf("%f", &a);

printf("Введите значение b: "); scanf("%f", &b);

printf("Введите значение x\_нач: "); scanf("%f", &xb);

printf("Введите значение x\_кон: "); scanf("%f", &xe);

printf("Введите значение dx: "); scanf("%f", &dx);

x = xb;

if (x <= xe)

{

printf("\n+-----------------+----------------------------------+");

printf("\n| Значение X | z ( X ) = ... |");

printf("\n+-----------------+----------------------------------+");

while (x <= xe)

{

// Расчёт значения функции

if (x <= a)

{

z = ((pow(x, 3.5) \* sin( pow(x, 2.2) ) <= 1.414) && x > 0) ? (sqrt( 1.414 - pow(x, 3.5) \* sin( pow(x, 2.2) ) ) + log(x)) : 666.666;

}

else if (x < b)

{

z = 2 \* exp(3.14 \* x) \* sin( 2.1 \* x - PI / 5 );

}

else

{

z = (x >= 0) ? ( fabs( pow(x, 5.1) \* cosh(x) - 1.4 ) \* atan( pow(x, 2.6) ) ) : 666.666;

}

// Вывод значения

printf("\n| %15f | %32f |", x, z);

x += dx;

}

printf("\n+-----------------+----------------------------------+");

} else {

printf("Начальное значение больше конечного.");

}

printf("\nПрограмма завершила свою работу....\n");

return 0;

}

* + 1. Текст программы (ЯП C++, операционная система GNU/Linux Ubuntu)

#include <iostream>

#include <iomanip>

#include <cmath>

#define PI 3.14

using namespace std;

int main ()

{

long double a, b, x, z, xb, xe, dx;

cout << " (!) Внимание (!) " << endl

<< "Если значение функции невычисляемо, в ячейке будет отображено число 666.666 ." << endl;

cout << "Введите значение a: "; cin >> a;

cout << "Введите значение b: "; cin >> b;

cout << "Введите значение x\_нач: "; cin >> xb;

cout << "Введите значение x\_кон: "; cin >> xe;

cout << "Введите значение dx: "; cin >> dx;

x = xb;

if (x <= xe)

{

cout << endl << "+-----------------+----------------------------------+"

<< endl << "| Значение X | z ( X ) = ... |"

<< endl << "+-----------------+----------------------------------+";

while (x <= xe)

{

if (x <= a)

{

z = ((pow(x, 3.5) \* sin( pow(x, 2.2) ) <= 1.414) && x > 0) ? (sqrt( 1.414 - pow(x, 3.5) \* sin( pow(x, 2.2) ) ) + log(x)) : 666.666;

}

else if (x < b)

{

z = 2 \* exp(3.14 \* x) \* sin( 2.1 \* x - PI / 5 );

}

else

{

z = (x >= 0) ? ( fabs( pow(x, 5.1) \* cosh(x) - 1.4 ) \* atan( pow(x, 2.6) ) ) : 666.666;

}

cout << endl << "| " << setw(15) << x << " | " << setw(32) << z << " |";

x += dx;

}

printf("\n+-----------------+----------------------------------+");

} else {

printf("Начальное значение больше конечного.");

}

printf("\nПрограмма завершила свою работу....\n");

return 0;

}

* + 1. Описание тестовых примеров

Тестовые примеры представлены на Рисунках 2.2, 2.3 и 2.4.

На Рисунке 2.2 продемонстрирована работа программы, написанной на языке C++. Были введены числа: 2, 3, -1, 3, 0.4. В результате программа вывела таблицу.

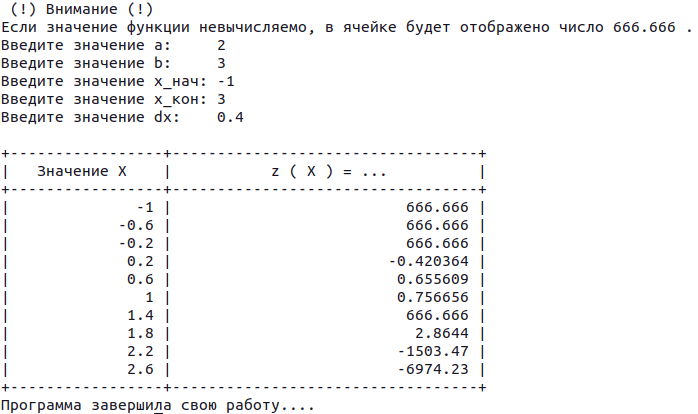


Рисунок 2.2 – Результат работы программы на С++

На Рисунке 2.3 продемонстрирована работа программы, написанной на языке C. Были введены числа: 3.5, 4, 3, 4.2, 0.1. В результате программа вывела таблицу.

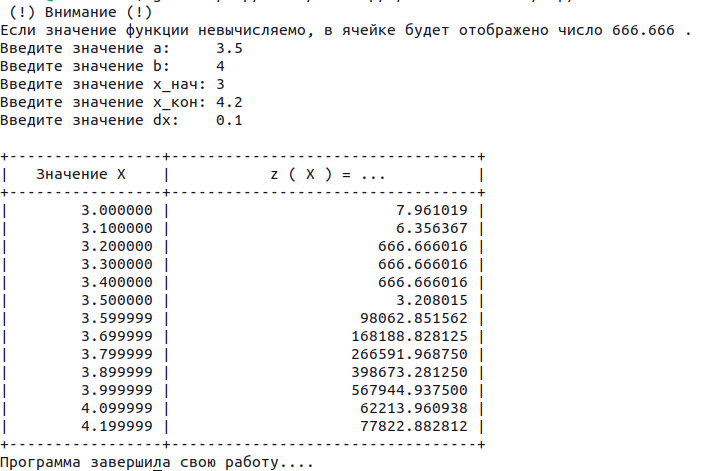


Рисунок 2.3 – Результат работы программы на языке C

Если пользователь введёт неверные данные, а именно если начальное значение X больше конечного, то программа выдаст ошибку. Этот случай показан на Рисунке 2.4.

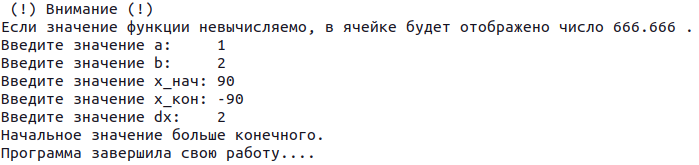


Рисунок 2.4 – Сообщение об ошибке входных данных

## Вывод

В ходе выполнения данной лабораторной работы были изучены и реализованы на практике навыки создания циклических структур for и while. Также были закреплены навыки работы с условными структурами, с математическими функциями, закреплены способы ввода/вывода данных на языках программирования C/C++. Полученные навыки в будущем позволят создавать функции обработки массивов, чтения данных из файла и тому подобное.