**ЛАБОРАТОРНАЯ РАБОТА №2**

**«ПРОГРАММИРОВАНИЕ ЛИНЕЙНЫХ И РАЗВЕТВЛЯЮЩИХСЯ АЛГОРИТМОВ»**

**Цель работы**

Изучение структуры С-программы. Формирование навыков программирования алгоритмов линейной и разветвляющейся структуры на языке С. Исследование особенностей ввода-вывода значений стандартных типов в языках С/С++.

**Порядок выполнения работы**

1. Определить функцию, для решения которой далее будет написана программа:

(Вариант 12)

Данная функция z=f(x) является разветвляющейся и состоит из трёх ветвей. Также в функции содержатся входные параметры a, b, x, значения которых неизвестны и вводятся перед решением.

1. Проанализировать область определения вычисляемой функции. В данном случае при x<=0 x>a, т.к. ln(x) определяется только при x>0. Из этого следует, что при (x<=0) и (x<=a) функция не определена.
2. Разработать структурную схему алгоритма решения задачи.

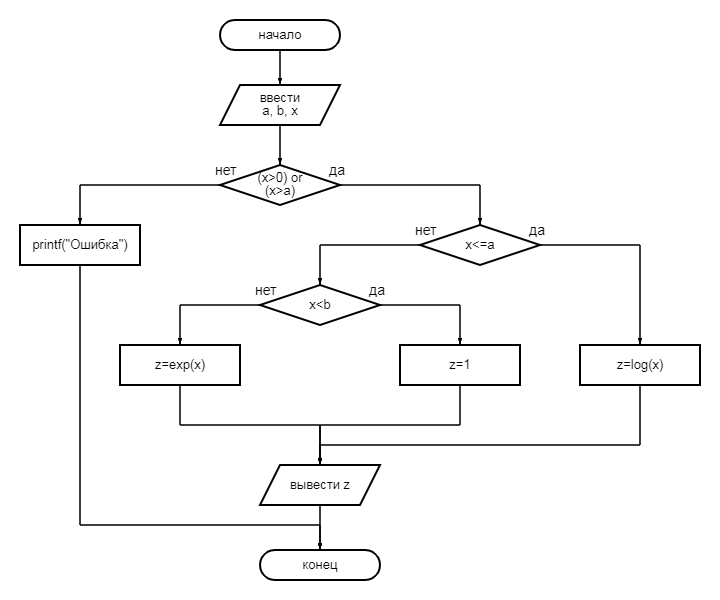


Рисунок 1 – Структурная схема алгоритма

1. Разработать тестовые примеры для данной функции:

* a=1, b=1, x=0: «Ошибка: невозможно вычислить ln(x) при x<=0»
* a=0, b=1, x=0: «Ошибка: невозможно вычислить ln(x) при x<=0»
* a=2, b=2, x=1: z=ln(x)=0
* a=3, b=3, x=2: z=ln(x)=0.693147
* a=1, b=3, x=2: z=1.000000
* a=0, b=1, x=1: z=ex=2.71828
* a=1, b=0, x=2: z=ex=7.38906

1. На основании написанного алгоритма (рисунок 1) написать программу на языке C для вычисления функции z, используя для ввода-вывода данных функции scanf и printf. В результате программа должна вывести на экран полученное значение z:

#include<stdio.h>

#include<math.h>

#include<windows.h>

main()

//вычисление значения функции z=f(x)

{

**float** a,b,x,z;

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

printf("Вычисление функции z=f(x), вариант 12:\n");

//ввод значений параметров a, b и аргумента x

printf("Введите параметр a: ");

scanf("%f",&a);

printf("Введите параметр b: ");

scanf("%f",&b);

printf("Введите переменную x: ");

scanf("%f",&x);

//проверка области определения

**if** ((x>0) || (x>a))

{

//вычисление значения функции z

**if** (x<=a) z=log(x);

**else**

**if** (x<b) z=1;

**else** z=exp(x);

//вывод значения z

printf("f(x)=%f",z);

}

//сообщение об ошибке

**else** printf("Ошибка: невозможно вычислить ln(x) при x<=0");

**return** 0;

}

1. На основании написанного алгоритма (рисунок 1) написать программу на языке C++ для вычисления функции z, используя для потокового ввода-вывода данных объекты cin и cout. В результате программа должна вывести на экран полученное значение z:

#include <iostream>

#include<math.h>

#include<windows.h>

**using namespace** std;

main()

//вычисление значения функции z=f(x)

{

**float** a,b,x,z;

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

cout<<"Вычисление функции z=f(x), вариант 12:\n";

//ввод значений параметров a, b и аргумента x

cout<<"Введите параметр a: ";

cin>>a;

cout<<"Введите параметр b: ";

cin>>b;

cout<<"Введите переменную x: ";

cin>>x;

//проверка области определения

**if** ((x>0) || (x>a))

{

//вычисление значения функции z

**if** (x<=a) z=log(x);

**else**

**if** (x<b) z=1;

**else** z=exp(x);

//вывод значения z

cout<<"f(x)="<<z;

}

//сообщение об ошибке

**else** cout<<"Ошибка: невозможно вычислить ln(x) при x<=0";

**return** 0;

}

1. Выполнить отладку и тестирование написанных программ.

Протестировать поведение функции при вводе значений, не соответствующих области её определения:

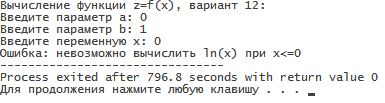


Рисунок 2 – Тестирование области определения

Протестировать, выдаст ли функция верный ответ при значениях, удовлетворяющих её первой ветви.

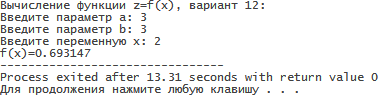


Рисунок 3 – Тестирование первой ветви алгоритма

Протестировать, выдаст ли функция верный ответ при значениях, удовлетворяющих её второй ветви.

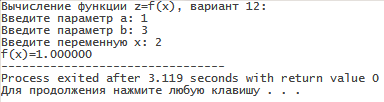


Рисунок 4 – Тестирование второй ветви алгоритма

Протестировать, выдаст ли функция верный ответ при значениях, удовлетворяющих её третьей ветви.

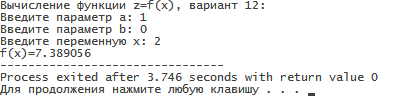


Рисунок 5 – Тестирование третьей ветви алгоритма

По результатам тестирования на написанных ранее тестовых примерах программами были выведены верные ответы (выведенные на рисунках 2-5 результаты аналогичны для обеих написанных программ).

**Вывод**

В ходе работы была изучена структура C-программ. Были сформированы навыки программирования алгоритмов линейной и разветвляющейся структуры на языке, исследованы особенности ввода-вывода значений стандартных типов в языках С/С++.

Результатом работы стали две программы на языке C и C++, находящие значение разветвляющейся функции z=f(x) с входными параметрами a и b. На основании результатам тестирования был сделан вывод, что обе написанные программы полностью справляются со своей задачей.