МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФГБОУ ВО АЛТАЙСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Физико-технический факультет

Кафедра информатики и вычислительной техники (ИВТ)

Лабораторная работа №02

**Решение задач с использованием ветвящихся алгоритмов**

Выполнил студент 585 гр.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В.М. Губченко

Проверил: к.т.н,, доцент каф. ВТиЭ

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Ю.Г. Скурыдин

Лабораторная работа защищена

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2018 г.

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Барнаул 2018

1. **Формулировка задачи**

Для заданного числа a найти корень уравнения

где

1. **Постановка задачи**

На вход поступает число a. Программа сравнивает данное число с нулём. Если a>= 0, то программа находит корень уравнения , иначе она находит корень уравнения . На выход поступает действительное число в диапазоне -∞ до +∞

1. **Математическая модель**

Для построения математической модели рассмотрим пример. Если программа получит на вход число 100, она сравнит его с нулём. Т.к. 100 > 0, то далее программа будет искать корень уравнения . Корнем уравнения является число -0.495. Если программа получит на вход число -3, она сравнит его с нулём. -3 < 0, поэтому программа будет искать корень уравнения . Корнем уравнения является 2.302585.

1. **Описание алгоритма**

**Начало алгоритма**

* 1. ввод числа num
  2. проверка условия: num>=0. Если условие выполняется, то продолжить алгоритм, иначе переход к пункту 4.4.1

4.4 присваивание переменной x значение (-1 \* fabs(num-1)) / (2 \* num)

4.5 вывод x

4.4.1 присваивание переменной x значение log(1 + (num \* num))

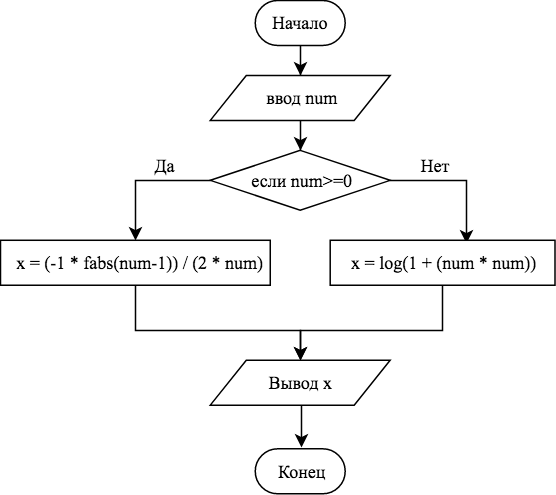
4.5.1 вывод x

**Конец алгоритма**

1. **Проект программы с нахождением корня уравнения**

Проект программы с определением замкнутых программных единиц и необходимых структур данных. В данном случае программа должна состоять из пяти частей: 1 блок ввода, 1 блок проверки условия, 2 блока арифметических вычислений и 1 блок вывода. В качестве структуры данных могут быть введены переменные для хранения исходного числа, переменной x, результата.

1. **Опорный граф (блок-схема) алгоритма**



1. **Текст программы:**

**на языке С++**

**//***Программа, которая находит корень уравнения*

#include <iostream>

#include <cstdlib>

#include <math.h>

using namespace std;

int main (int argc, char const \*argv[]) {

float num, x;

cout << "enter number" << '\n';

cin >> num ;

if (num>=0) {

x = (-1 \* fabs(num-1)) / (2 \* num);

} else {

x = log(1 + (num \* num));

}

cout << "result= " << x ;

return 0;

}

**на языке Pascal**

*{ Программа, которая находит корень уравнения}*

Program lab\_02;

var x, num: real;

Begin

readln(num);

if num >= 0 then

begin

x := (-1 \* abs(num-1)) / (2 \* num);

end

else

begin

x := ln(1 + (num \* num));

end;

writeln(x);

readln;

end.

1. **Проверка работоспособности (тестирование) программы**

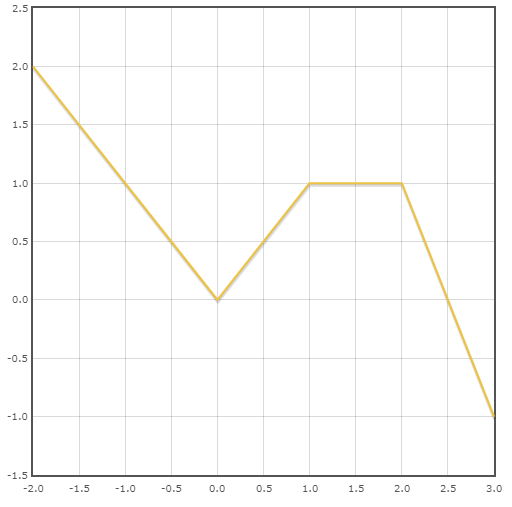
Для проверки работоспособности нужно просто запустить программу.

1. **Сравнительный анализ и оценка эффективности работы программ на разных языках программирования**

Ввиду простоты программы она одинаково эффективно выполняется для обоих языков программирования.

1. **Формулировка задачи**

Дано действительное число a. Для функции , графики которых представлены на рисунке. Вычислить .



1. **Постановка задачи**

На вход поступает действительное число x. Программа выполняет арифметические действия зависящие от координаты xНа выход поступает действительное число

1. **Математическая модель**
2. Программа сравнивает данное число сначала с нулём, если x<0, то программа находит корень уравнения , иначе она проверяет, принадлежит ли число x полуинтервалу [0;1). Если принадлежит, то программа ищет корень уравнения . Иначе она проверяет, принадлежит ли число x полуинтервалу [1;2). Если принадлежит, то . Иначе **Описание алгоритма**

**Начало алгоритма**

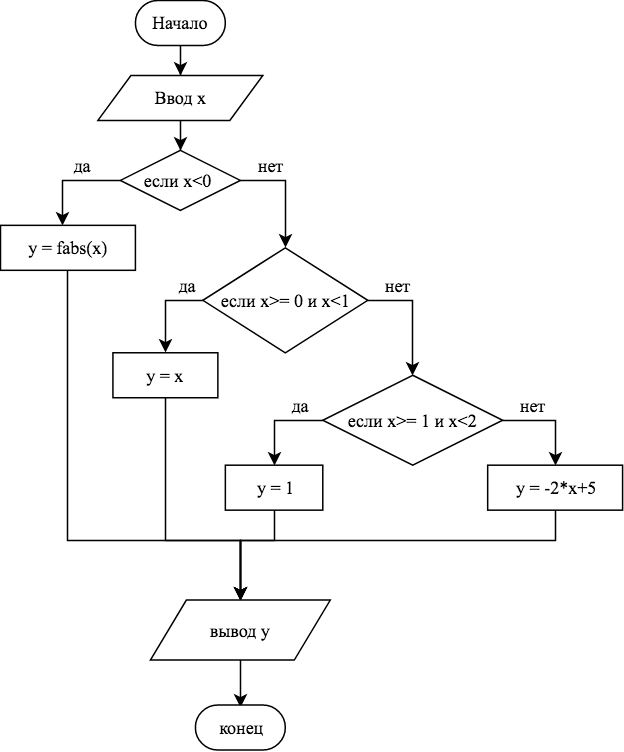
1. Ввод числа x
2. Если x < 0, то присвоение переменной y значения |x|
3. Если x≥0 и x<1, то присвоение переменной y значения x
4. Если x≥1 и x<2, то присвоение переменной y значения 1
5. Если x≥2, то присвоение переменной y значения -2\*x+5
6. Вывод y

**Конец алгоритма**

1. **Проект программы с нахождением корня уравнения**

Проект программы с определением замкнутых программных единиц и необходимых структур данных. В данном случае программа должна состоять из десяти частей: 1 блок ввода, 4 блока проверки условия, 4 блока арифметических вычислений и 1 блок вывода. В качестве структуры данных могут быть введены переменные для хранения исходного числа, переменной y, результата.

1. **Опорный граф (блок-схема) алгоритма**



1. **Текст программы:**

**на языке С++**

#include <iostream>

#include <cstdlib>

#include <math.h>

using namespace std;

int main (int argc, char const \*argv[]) {

float x, y;

cout << "x=";

cin >> x;

if (x<0) { y = fabs(x); }

else if (x>=0 && x<1) { y = x; }

else if (x>=1 && x<2) { y = 1;}

else y = -2\*x+5;

cout << "y=" << y <<'\n';

return 0;

}

**на языке Pascal**

Program lab\_02;

var x, y: real;

Begin

readln(x);

if x < 0 then

begin

y := abs(x);

end

else if (x >= 0) and (x < 1) then

begin

y := x;

end

else if (x >= 1) and (x < 2) then

begin

y := 1;

end

else if x >= 2 then

begin

y := -2\*x+5;

end;

writeln(y);

readln;

end.

1. **Проверка работоспособности (тестирование) программы**

Для проверки работоспособности нужно просто запустить программу.

1. **Сравнительный анализ и оценка эффективности работы программ на разных языках программирования**

Ввиду простоты программы она одинаково эффективно выполняется для обоих языков программирования.