Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Вятский государственный университет»

Колледж ВятГУ

**ОТЧЕТ**

**ПО ДОМАШНЕЙ КОНТРОЛЬНОЙ РАБОТЕ №1**

**«ВЫЧИСЛЕНИЕ ЗНАЧЕНИЯ ФУНКЦИИ»**

**ПО ДИСЦИПЛИНЕ**

**«ОСНОВЫ АЛГОРИТМИРОВАНИЯ И ПРОГРАММИРОВАНИЯ»**

Выполнил: студент учебной группы

ИСПк-204-52-00

Резенов Андрей Андреевич

Преподаватель:

Сергеева Елизавета Григорьевна

**Цель работы:** изучить базовую структуру организации программы и основные конструкции языка программирования Pascal.

**Задание:** *(Вариант 15)*

1. Написать программу, вычисляющую значение функции:

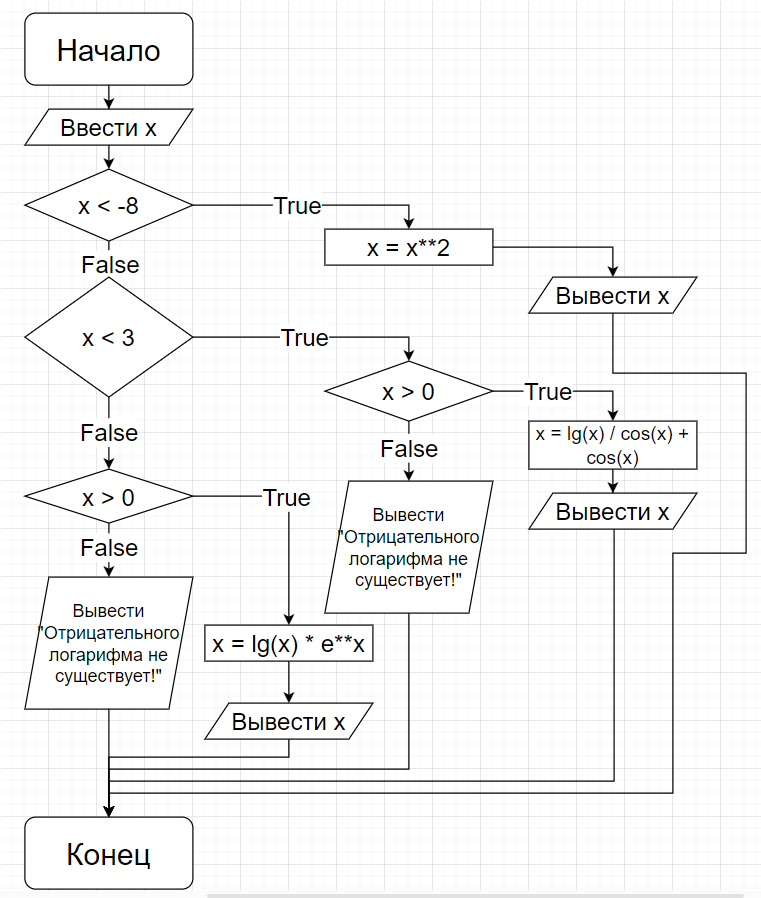
x ^ 2, если x < -8;

lg(x) / cos(x) + cos(x), если -8 <= x < -3;

lg(x) \* e ^ x, если -3 <= x.

Cхема алгоритма:

Рисунок 1



Описание алгоритма:

Вводим x, далее идет проверка «*x* меньше -8». Если проверка выполняется, то х возводится в квадратную степень и выводится, иначе далее идет проверка «*х* меньше 3». При выполнении проверки идет проверка «*х* больше 0». При выполнении проверки производится вычисления в переменной *х* «деление десятичного логарифма *х* на косинуса *х* и сложение с косинусом *х*», после чего выводится *х*, иначе выводится «Отрицательного логарифма не существует!». Если проверка «-8 меньше или равно *х* и *х* меньше 3» не выполняется, то далее идет проверка «*х* больше 0». Если проверка выполняется, то производится вычисление в переменной *х* «умножение десятичного логарифма *x* на число Эйлера в степени *х*» и выводится *х*, иначе выводится «Отрицательного логарифма не существует!».

Код программы:

var

x: real;

begin

Readln(x);

if x < -8 then Writeln(x\*\*2)

else if x < 3 then

begin

if x > 0 then Writeln(Log10(x) / Cos(x) + Cos(x))

else Writeln('Отрицательного логарифма не существует!');

end

else

begin

if x > 0 then Write(Log10(x) \* E\*\*x)

else Writeln('Отрицательного логарифма не существует!');

end;

end.

Результат выполнения программы:

Рисунок 2



Рисунок 3

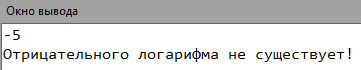
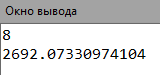
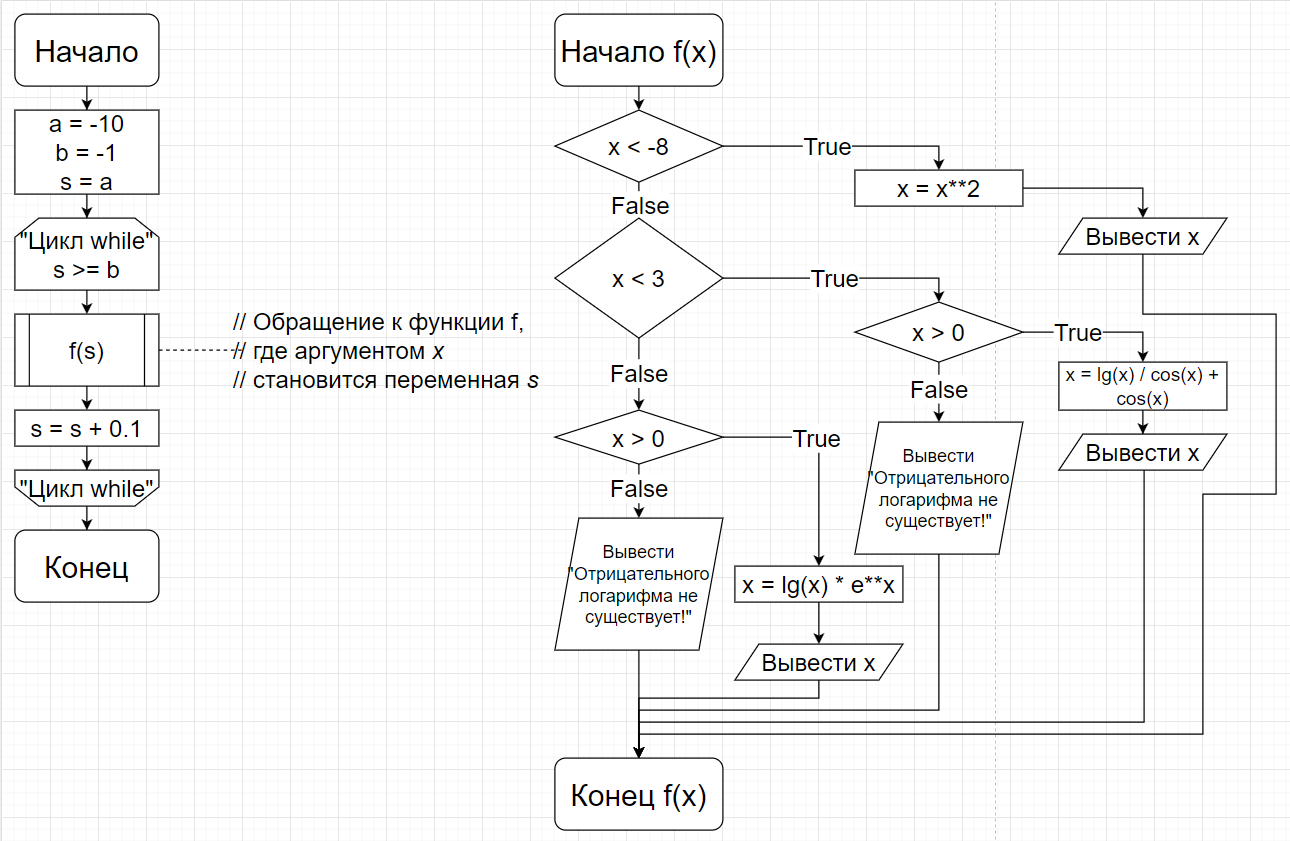


Рисунок 4



1. Вычислить значение функции на интервале [-10; -1] с шагом 0,1.

Схема алгоритма:



Описание алгоритма:

Переменным *a*, *b* и *s* присваиваются значения -10, -1 и *a* соответственно. Далее идет цикл с предусловием *s >= b*. Пока *s < b*, идет обращение к функции f с аргументом *x*, который принимает переменную *s* и выводит результат, потом переменная *s* увеличивается на 0.1.

Код программы:

var

a, b: integer;

s: real;

function f(x: real): real;

begin

if x < -8 then Write(x\*\*2, ' ')

else if x < 3 then

begin

if x > 0 then Write(Log10(x) / Cos(x) + Cos(x), ' ')

else Write('Отрицательного логарифма не существует! ');

end

else

begin

if x > 0 then Write(Log10(x) \* E\*\*x, ' ')

else Write('Отрицательного логарифма не существует! ');

end;

end;

begin

a := -10;

b := -1;

s := a;

while s < b do

begin

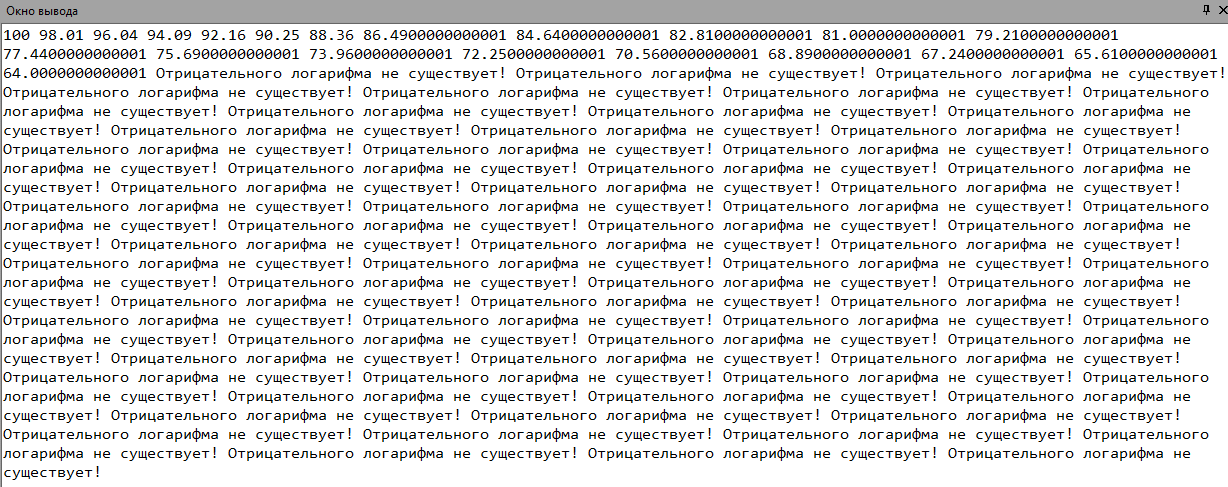
f(s);

s += 0.1;

end;

end.

Результат выполнения программы:



**Ответы на вопросы:**

1. Что такое алгоритм? — это чёткая последовательность действий, выполнение которой даёт какой-то заранее известный результат.
2. Перечислите виды алгоритмов. — линейные, ветвящиеся (условные), циклические.
3. Дайте определение условному алгоритму. — действия в этом алгоритме выполняются, если верны некоторые условия для их выполнения.
4. Виды условных алгоритмов. Какие использовали для решения задачи? — условные алгоритмы бывают полные (есть ветки True и False) и неполные (есть только True). Использовались только полные условные алгоритмы.
5. Дайте определение циклическому алгоритму. — действия в этом алгоритме повторяются в теле цикла, пока выполняются некоторые условия.
6. Виды циклических алгоритмов. Какие использовали для решения задачи? — цикл с предусловием, цикл с постусловием, цикл с параметром. Использовался только цикл с предусловием.

**Вывод:** подводя итоги, можно сказать, что работа оказалась более трудоёмкой, чем ожидалось на первый взгляд, но очень интересной. Пришлось несколько раз переписывать код для первой задачи, оптимизировать её для практики, для этой же задачи начать проектировать схему, предварительно изучив оформление по ГОСТу, разобраться в структуре и на основе практических занятий и написанному коду завершить создание. Вторая же задача показалась очень простой, но только если рассматривать её как задачу, никак не связанную с прошлой. Но оказалось задача предусматривала использование кода с первой задачи, что сильно затруднило выполнение работы. Пришлось изучить синтаксис и алгоритм создания функций в Паскале, а также оформление этих функций и их связей с основным телом в виде схемы. Но в итоге все затруднения были преодолены и задание было выполнено успешно. Мы смогли изучить базовую структуру организации программы и основные конструкции языка программирования Pascal, а также составление базовых схем и алгоритмов.