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 функций»**

**ПО ДИСЦИПЛИНЕ «Основы алгоритмизации и программирования»**

Выполнила: студентка учебной группы

ИСПк-204-52-00

Сергеева Ксения Геннадьевна

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2024

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

1. Написать программу, вычисляющую значение функции (Вариант 19):
2. ln(x) / lg(x) - cos(x) / e ^ x, если x < -10; Вводим x, который меньше -10, считаем y, выводим y.

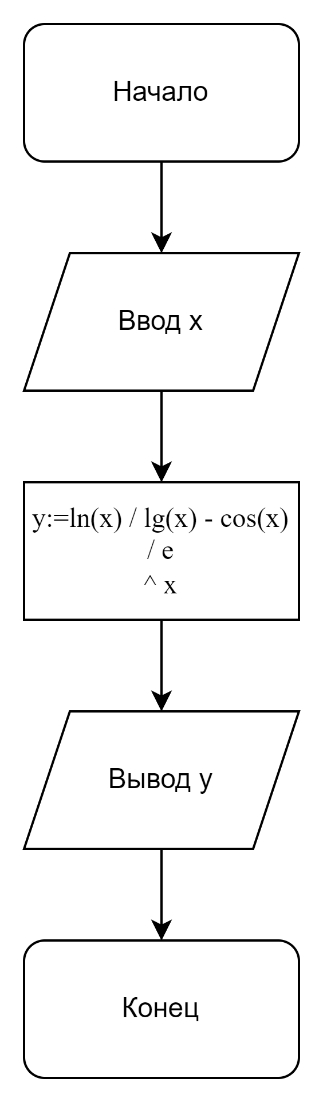


Рисунок 1. Схема алгоритма для задачи 1. 1.

Код:

var x,y: real;

begin

writeln('Введите x,при x<-10');

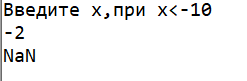
readln(x);

y:=(ln(x)/(ln(x)/ln(10)))-(cos(x)/exp(x));

writeln(y);

end.

Результат:



1. x ^ (0,1 \* x) / sin(x) - x ^ 2 / cos(2 \* x), если -10 <= x < 0;Вводим x, при -10 <= x < 0, прописываем условие, считаем y, выводим y.

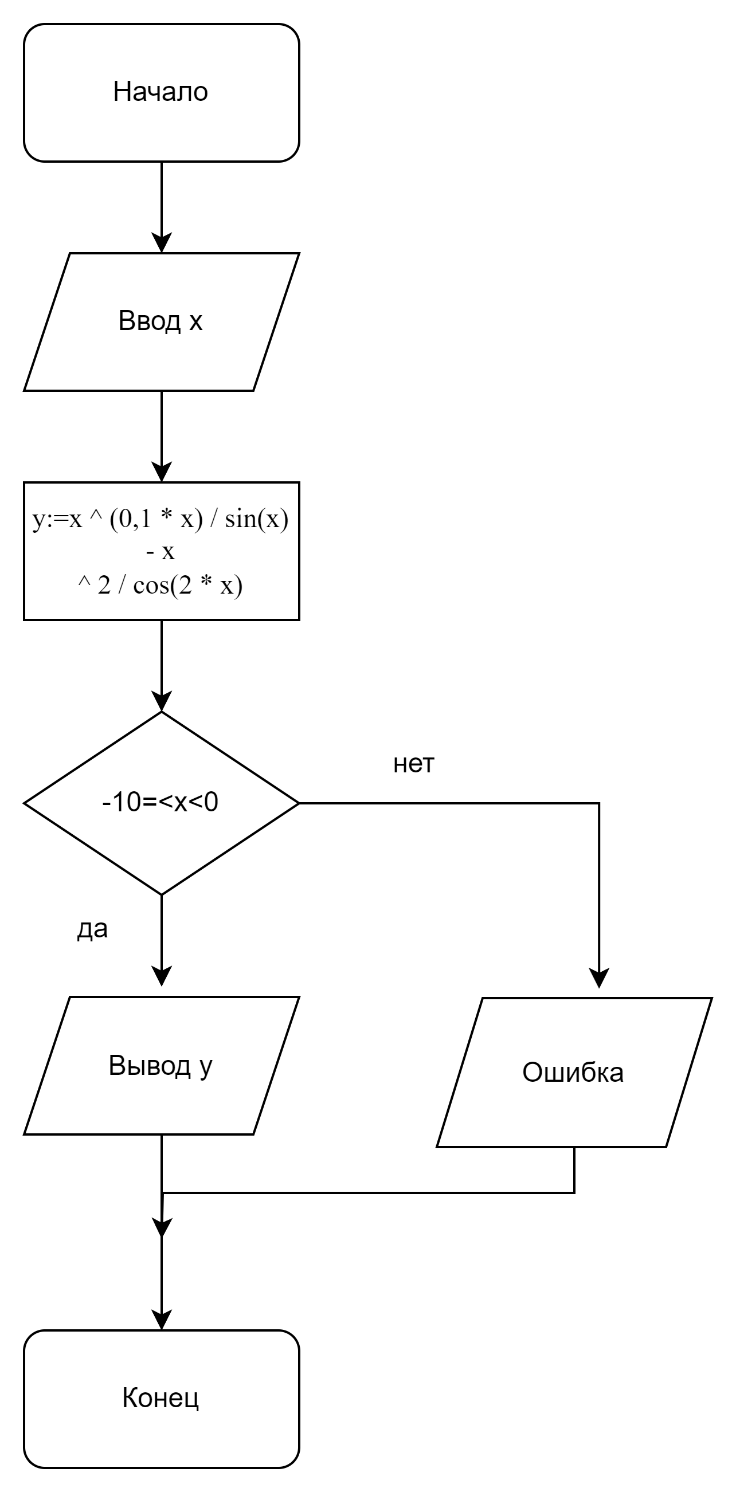


Рисунок 2. Схема алгоритма для задачи 1. 2.

Код:

**Var** x, y: real;

**begin**

writeln('Введите x, при -10 <= x < 0: ');

readln(x);

**if** (x >= -10) **and** (x < 0) **then**

**begin**

y:=(exp(0.1\*ln(abs(x)))/sin(x))-((x\*x)/cos(2\*x));

writeln('Результат y=', y:0:4);

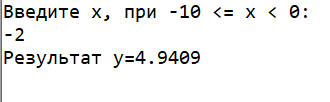
**end**

**else**

writeln('Ошибка: x должен быть в диапазоне -10 <= x < 0');

**end**.

Результат:



1. cos(2 \* x) \* 91 + ln(x) / x ^ 3, если 0 <= x;

Вводим x, при 0<=x, считаем y, выводим y.

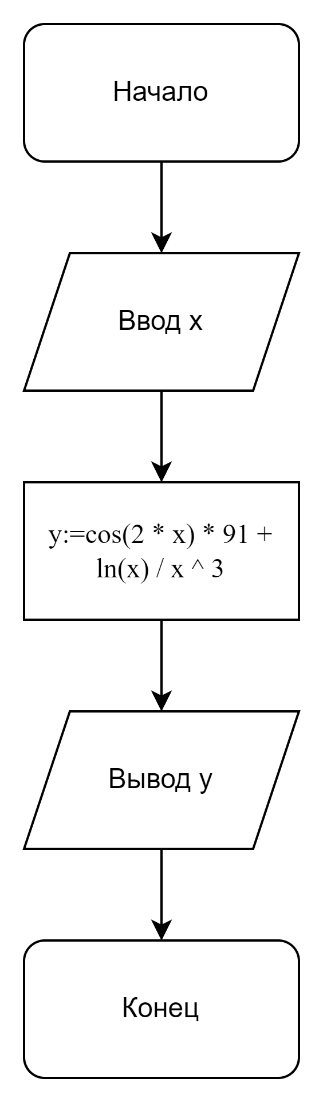


Рисунок 3. Схема алгоритма для задачи 1. 3.

Код:

**var** x,y: real;

**begin**

writeln('Введите x,при 0<= x');

readln(x);

y:=cos(2\*x)\*91+ln(x)/exp(3\*ln(x));

writeln(y);

**end**.

Результат:



2.

1) var x: real;

y:real;

begin

x:=-12;

while x <= 2 do

begin

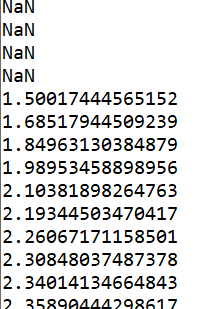
y:= (ln(x)/(ln(x)/ln(10)))-(cos(x)/exp(x));

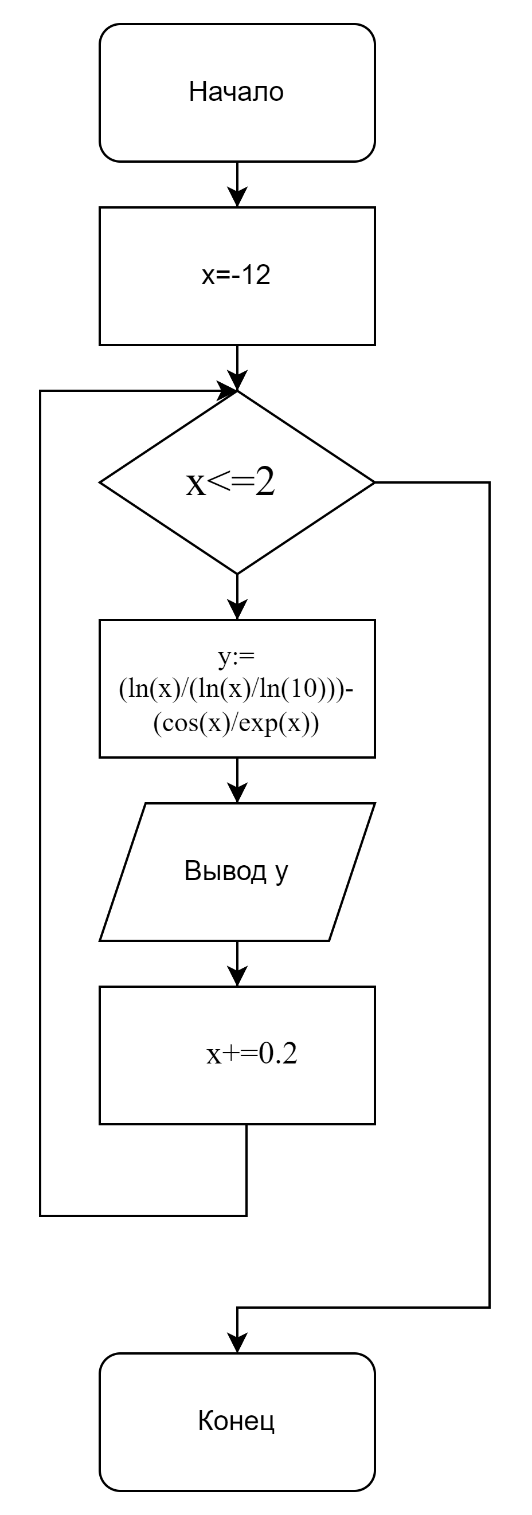
writeln(y);

x+=0.2;

end;

end.

Результат:  




2) var x: real;

y:real;

begin

x:=-12;

while x <= 2 do

begin

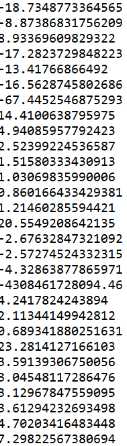
y:=(exp(0.1\*ln(abs(x)))/sin(x))-((x\*x)/cos(2\*x));

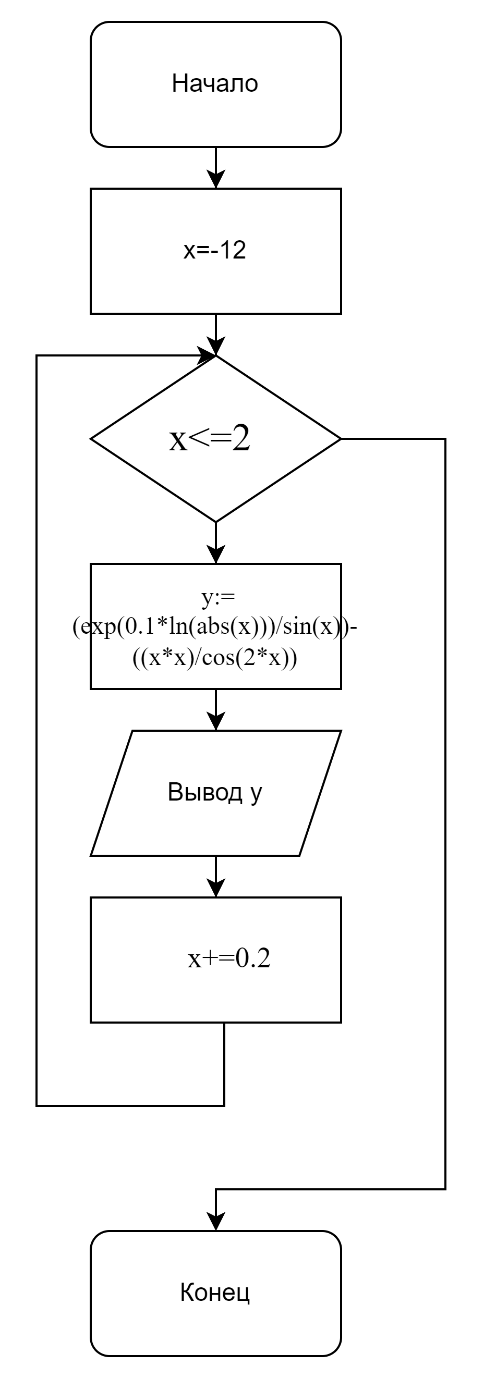
writeln(y);

x+=0.2;

end;

end.

Результат:  




3) var x: real;

y:real;

begin

x:=-12;

while x <= 2 do

begin

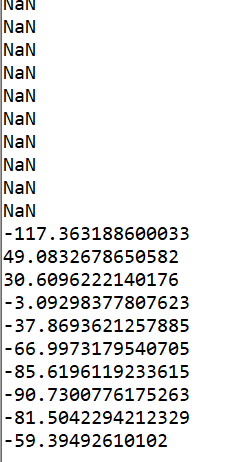
y:=cos(2\*x)\*91+ln(x)/exp(3\*ln(x));

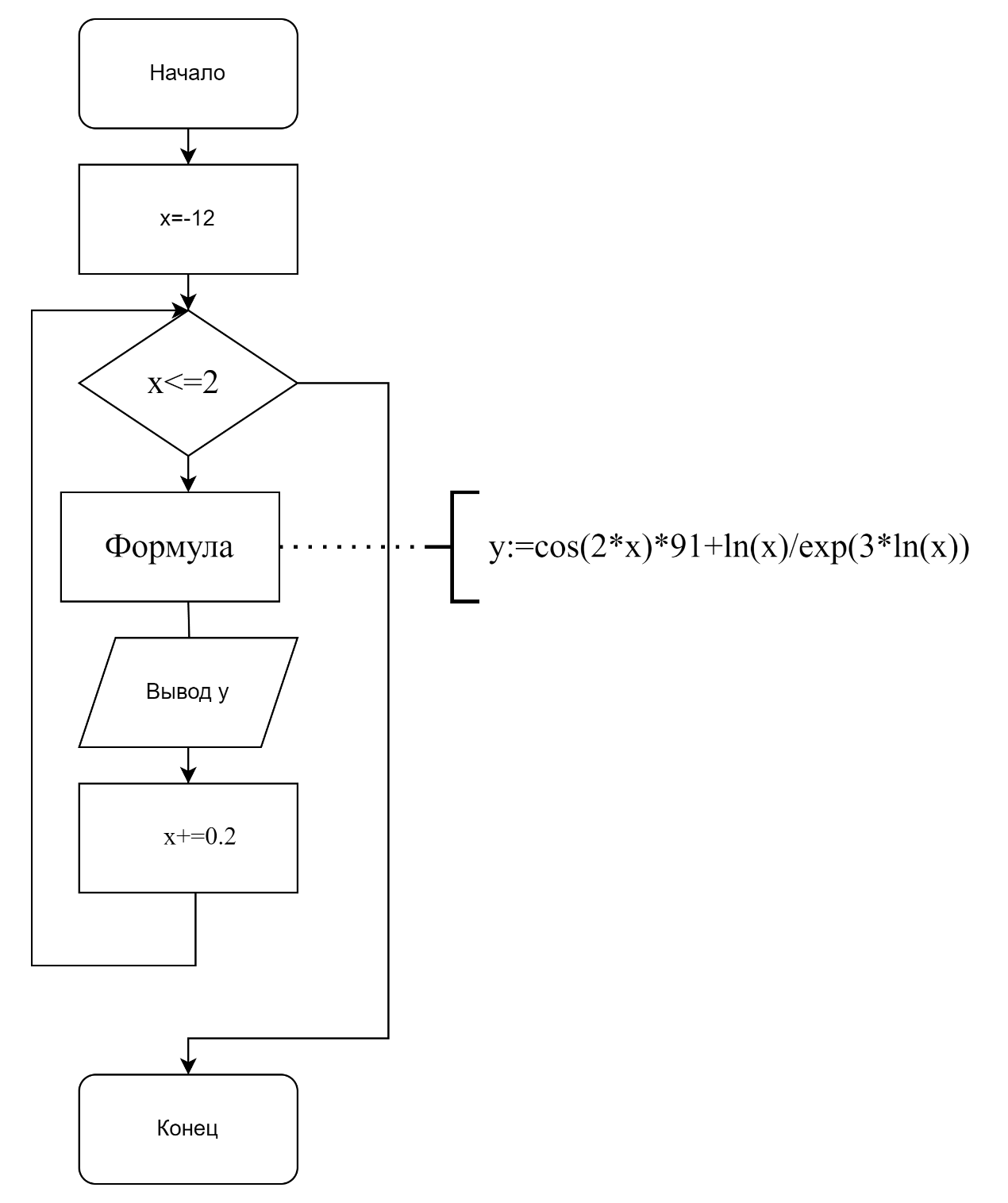
writeln(y);

x+=0.2;

end;

end.

Результат:  




Ответы на вопросы:  
1. Точный набор инструкций, описывающих последовательность действий для достижения результата, решения задачи.

1. Линейный, ветвящийся, циклический.
2. Это алгоритм, порядок выполнения команд которого зависит от истинности или ложности некоторого условия.
3. Условные алгоритмы могут быть с полным выражением if – else. C неполным выражением if. C оператором case. Использовались: полное выражение.
4. Это алгоритм, в котором некоторая часть операций повторяется многократно.
5. Циклический алгоритм с постусловием. Циклический алгоритм с предусловием. Циклический алгоритм со счетчиком. Циклический алгоритм с выходом из середины. Использовался: цикл с предусловием.

Вывод:

NaN - это "not a number", получается при 0/0.

1)В первой задаче в результате программа выводит NaN, так как в условиях задачи написано, что x должен быть меньше -10, но по правилам математики, логарифма от отрицательного числа быть не может.

2) Во второй и в третьей задаче при возведении в степень, нужно знать, что это делается через формулу exp(b\*ln(a)), где a – основание, b – степень.

3) Для уточнения результата, все значения функций были найдены вручную, все результаты сошлись.