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 АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ»**

Выполнила: студент учебной группы

ИСПк-202-52-00

Савватеев Аркадий Сергеевич

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2024

**Цель работы**: изучить базовую структуру организации программы и основным конструкция языка программирования Pascal.

1. **Формулировка задания 1. Вариант 20**

**Постановка задачи**

Написать программу, вычисляющую значение функции:

< -6;

, если -6 <= x < -2;

, если -2 <= x.

1. **Описание алгоритма и ответы на вопросы**

***1. Что такое алгоритм?***

Алгоритм – это набор инструкций, описывающих порядок действий исполнителя для достижения результата решения задачи за конечное число действий.

***2. Перечислите виды алгоритмов.***

* Линейный
* Разветвляющийся
* Циклический

***3. Дайте определение условному алгоритму.***

Условный алгоритм — это алгоритм, содержащий хотя бы одну проверку условия, в результате которой обеспечивается переход на один из возможных вариантов решения;

***4. Виды условных алгоритмов. Какие использовали для решения задачи?***

Существует две реализации структуры ветвления – полная и неполная:

Полная форма ветвления означает, что осуществляется выбор между двумя действиями. Если проверка условия даёт результат «да», то выбирается первое действие, иначе — второе действие.

Краткая форма ветвления предполагает, что если условие истинно, то выполняется первая команда, иначе — никакие действия не выполняются.

При выполнении работы была использована полная форма ветвления.

***5. Дайте определение циклическому алгоритму.***

Циклический — алгоритм, предусматривающий многократное повторение одного и того же действия (одних и тех же операций) над новыми исходными данными. К циклическим алгоритмам сводится большинство методов вычислений и перебора вариантов.

***6. Виды циклических алгоритмов. Какие использовали для решения задачи?***

Циклы с предусловием

Если в цикле с предусловием входящие в тело цикла команды могут не выполняться ни разу (если начальное значение параметра цикла удовлетворяет условию выхода из цикла).

Циклы с постусловием

В цикле с постусловием – выполняются как минимум один раз (даже если начальное значение параметра цикла удовлетворяет условию выхода из него).

В программе по решению задачи был использован цикл с предусловием.

***7. Описание алгоритма***

Для начала работы программы мы вводим значение переменной X. Далее мы переходим к первому условию программы, которое имеет вид, что если x <-6, то используется функция , если -6 <= x < -2, то используется функция , и соответственно если -2 <= x, то используется функция

1. **Схема алгоритма**

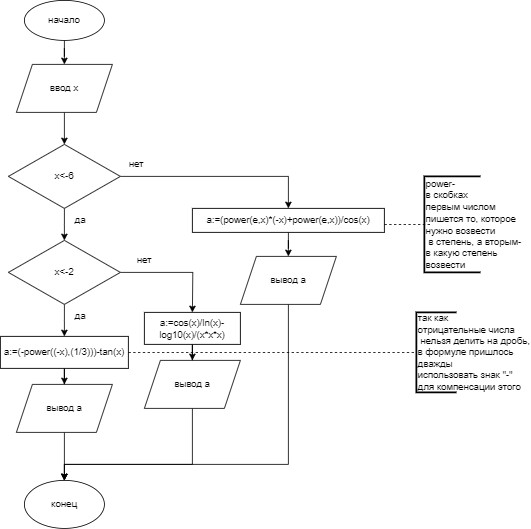


Рисунок 1 – схема алгоритма 1

1. **Код программы**

**program** qq;

**var**

x, a: real;

e: real;

**begin**

e := exp(1);

writeln('Введите значение x');

readln(x);

**if** x < -6 **then**

**begin**

a := (power(e, x) \* (-x) + power(e, x)) / cos(x);

writeln('a=',a)

**end**

**else**

**begin**

**if** (x < -2) **then**

**begin**

a := (- power((-x), (1 / 3))) - tan(x);

writeln('a=',a);

**end**

**else**

**begin**

a := cos(Abs(x)) / ln(Abs(x)) – ln(Abs(x))/log10(x) / (x \* x \* x);

writeln('a=',a);

**end**;

**end**;

**end**.

1. **Результат выполнения программы**

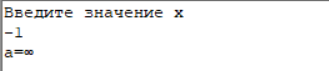
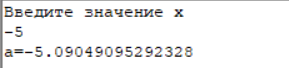


Рисунок 2 – примеры вводимых значений

1. **Формулировка задания 2. Вариант 20**

**Постановка задачи**

Написать программу, вычисляющую значение функции на отрезке [-8; 0] с шагом 0.1:

, если x < -6;

, если -6 <= x < -2;

, если -2 <= x.

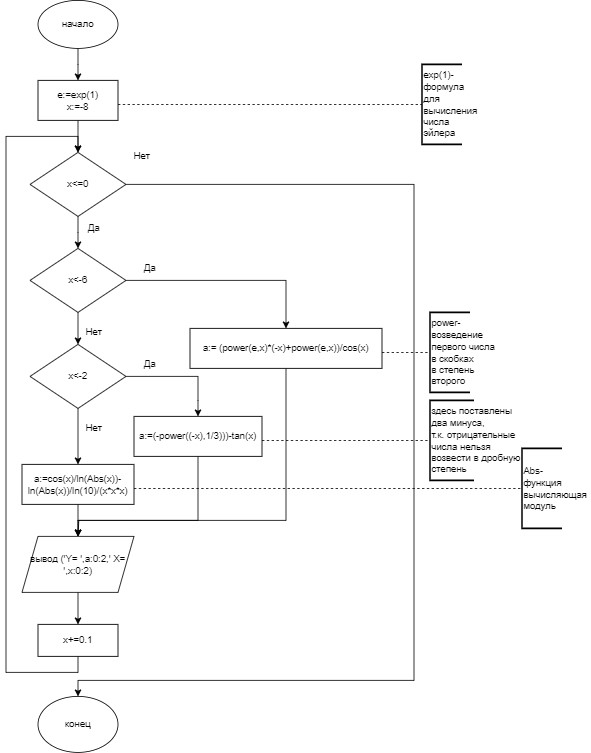
1. **Описание алгоритма**
2. Задаём переменной Х начальное значение = -8.
3. Устанавливаем ограничение для выполнения цикла <=0.
4. Далее мы переходим к первому условию программы, которое имеет вид x <-6, то используется функция *.* Если введённое число удовлетворяет условию, то мы производим вычисления по функции, в ином же случае мы переходим к следующему условию.
5. Второе условие имеет вид -6 <= x < -2, то вычисления производятся по функции . Если введённое число удовлетворяет условию, то мы производим вычисления по функции, в ином же случае мы переходим к следующему условию, третьему условию.
6. Третье условие имеет вид -2 <= x, то мы производим вычисления по формуле функции .
7. После выполнения одного из условий мы прибавляем к значению Х 0.1.
8. После прохождения всех условий программы мы выводим значение Х, которое было использовано, и значение функции, которое мы получили в итоге проведенной работы.
9. **Схема алгоритма с комментариями**
10. 

Рисунок 3 – схема алгоритма

1. **Код программы**

**program** qq;

**var**

x, a: real;

e: real;

**begin**

e := exp(1);

x := -8;

**while** x <= 0 **do**

**begin**

**if** x < -6 **then**

**begin**

a := (power(e, x) \* (-x) + power(e, x)) / cos(x);

writeln(a)

**end**

**else**

**begin**

**if** (x < -2) **then**

**begin**

a := (-power((-x), (1 / 3))) - tan(x);

writeln(a);

**end**

**else**

**begin**

a := cos(x) / ln(Abs(x)) - ln(Abs(x))/ln(10) / (x \* x \* x);

writeln(a);

**end**;

**end**;

writeln('Y= ',a:0:2,' X= ',x:0:2);

x += 0.1;

**end**;

**end**.

1. **Результат выполнения программы**

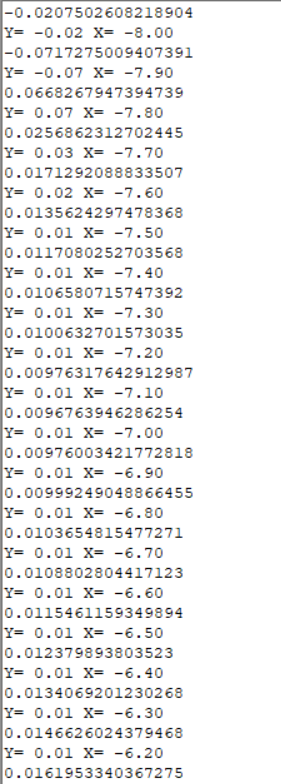


Рисунок 4.1 – примеры вводимых значений

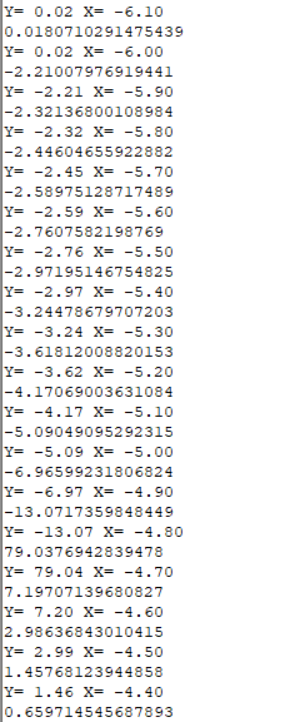


Рисунок 4.2 – примеры вводимых значений

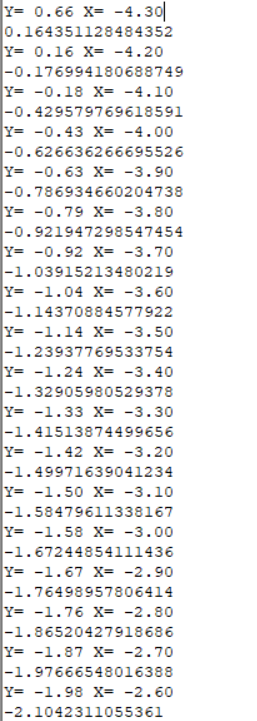
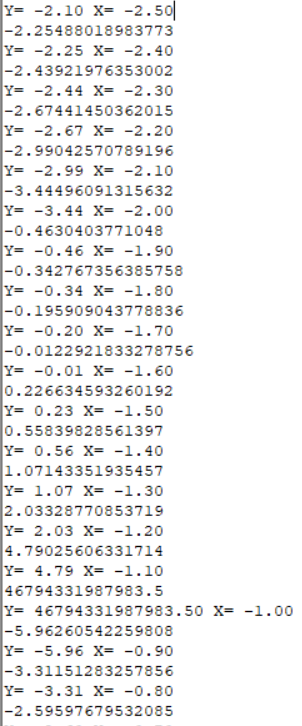


Рисунок 4.3 – примеры вводимых значений



Рис

Рисунок 4.4 – примеры вводимых значений

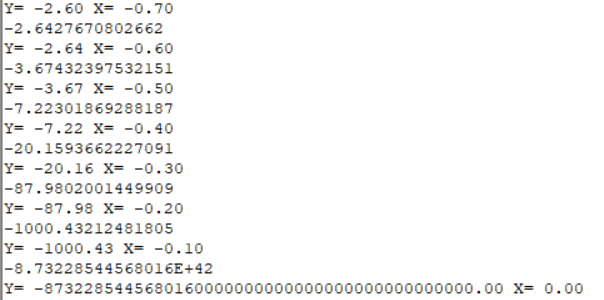


Рисунок 4.5 – примеры вводимых значений

1. **Вывод**

В ходе работы мы изучили базовую структуру организации программы, основные конструкции языка программирования Pascal и основы составления схем данных.

Также были получены знания о том, что такое: алгоритм, условный алгоритм, циклический алгоритм; какие бывают виды условных и циклических алгоритмов.

Для создания алгоритмических схем было изучено правильное построение письменного алгоритма для достижения верного решения задачи, а также и само построение схем в программе «draw.io».

Однако в ходе работы мы столкнулись и с проблемами. Например, при выполнении программы вместо ожидаемого нам ответа выводилось значение NaN, числовой тип данных, значение которого является неопределённым или непредставимым. Решением данной проблемы стало правильно поставленное условие. Так как программа предназначена для вычислений отрицательных чисел, то и в таком операторе как lg(x) должно стоять отрицательное значение Х. Также была встречена особенность интервала функции, где необходимо использовать модуль для получения данных, однако при этом в коде выходит своеобразное данное-исключение, где подставляется ноль, и округление до сотых не работает, так как значение уходит до бесконечности. Та же ситуация встретилась и при подстановке -1.

В конечном итоге при помощи полученных знаний и исправленных ошибок у нас получилось добиться поставленной цели – решить задачу при помощи кода с условиями и циклами, а также узнать о языке программирования Pascal новые знания.