**Лабораторная работа №3**

**«ПРОГРАММИРОВАНИЕ АЛГОРИТМОВ ЦИКЛИЧЕСКОЙ СТРУКТУРЫ»**

**Цель работы**: Получение навыков программирования алгоритмов циклической структуры на языке С/С++. Исследование эффективности применения различных видов циклов в задаче табулирования функции.

**1. Постановка задачи**

Вычислить и вывести на экран в виде таблицы значения функции z = f(x) на интервале от xнач до xкон с шагом Δx . Таблицу снабдить заголовком и шапкой. Вид функции z выбирать в соответствии с вариантами задания к лабораторной работе №2 настоящих методических указаний. Значения параметров a , b , а также xнач , xкон и Δx вводятся с клавиатуры. Результаты вычислений выводятся в формате с фиксированной точкой.

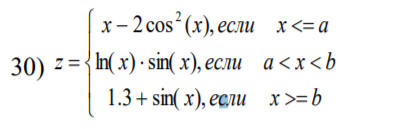


Рисунок 1 – 30-й вариант задания

Исходя из моего варианта (30 Вариант), функция выглядит так:

**2. Область определения функции**

Для выполнения данной задачи сначала рассмотрим область определения данной функции. Для x нельзя задать отрицательное значение, так как у функции Ln областью определения являются все положительные числа: D(f) = (0; +∞). Для этого в программе необходимо будет предусмотреть проверку значения x, введённого с клавиатуры. Так же, поскольку все это будет вычисляться в цикле от начала и до конца , xнач должен быть меньше xкон , а шаг должен быть больше 0.

**3. Алгоритм программы**

В цикле производятся вычисления , при этом меняется значение переменной. Цикл закончиться в том случае , когда значение x будет больше x конечного.

При помощи условных выражений определяется нужная формула для вычисления значения результата. Так же используются бесконечные циклы для проверки правильности ввода начальные данных , да бы в будущем программа выдала верные решения.



Рисунок 2 – Структурная схема программы для решения данной функции

**4. Код программы (на языке C с использованием библиотеки stdio.h)**

#pragma warning(disable : 4996) //отключение ошибки о старой функции

#include<stdio.h>

#include<windows.h> // Подключаем три препроцессора . Один для ввода/вывода , второй для вычисления , третий для подключения консоли

#include<math.h>

main()

{

float a,b, // параметры

x, // аргумент функции z

xn, // начальное значение аргумента x

xk, // конечное значение аргумента x

dx, // шаг

z; // значение функции z

SetConsoleCP(1251); // Задаем таблицу символов для консоли.

SetConsoleOutputCP(1251);

while (1) {

printf("Введите x начальное :\n"); scanf("%f", &xn);

if (xn > 0) { break; }

else printf("Введите xn > 0 \n"); //проверка на правильность введенного xn, если он верен , то цикл заканчивается

}// выводим в консоль текстовые данные и считываем введенные значения

while (1) {

printf("Введите x конечное :\n"); scanf("%f", &xk);

if (xk>xn) { break; }

else printf("Введите xk > xn \n"); //проверка на правильность введенного xk, если он верен , то цикл заканчивается

}// выводим в консоль текстовые данные и считываем введенные значения

while (1) {

printf("Введите шаг :\n"); scanf("%f", &dx);

if (dx > 0) { break; }

else printf("Введите dx > 0 \n"); //проверка на правильность введенного шага, если он верен , то цикл заканчивается

}// выводим в консоль текстовые данные и считываем введенные значения

printf("Введите a:\n"); scanf("%f", &a);// считываем остальные переменные

printf("Введите b:\n"); scanf("%f", &b);

x = xn;

printf("---------------------------------\n");

printf("| знач x | значение z |\n"); // заголовок таблицы

printf("---------------------------------\n");

while (x <= xk) {

if (x <= a) { // производим сравнение двух переменных

z = x - 2 \* pow(cos(x), 2); // Вычисляем значение для переменной исходя из условия

}

else // если не выполнилось условие то проверяем следующее условие

if ((a < x) && (x < b)) {

z = log(x) \* sin(x);// Вычисляем значение для переменной исходя из условия

}

else {// если не выполнилось условие то выполняем действие

z = 1.3 + sin(x);// Вычисляем значение для переменной исходя из условия

}

printf("| %.4f | %.4f |\n", x,z);// Выводим полученный результат функции

x += dx; // изменяем x на шаг

}

printf("---------------------------------\n"); //закрываем таблицу

return 0;

}

**5. Код программы (на языке C++ с использованием библиотеки iostream)**

#include<iostream>

#include<windows.h> // Подключаем три препроцессора . Один для ввода/вывода , второй для вычисления , третий для подключения консоли

#include<math.h>

using namespace std;

int main()

{

float a,b, // параметры

x, // аргумент функции z

xn, // начальное значение аргумента x

xk, // конечное значение аргумента x

dx, // шаг

z; // значение функции z

system("chcp 1251"); // Задаем таблицу символов для работы с консолью

while (1) {

cout << ("Введите x начальное :\n"); cin >>xn ;

if (xn > 0) { break; }

else cout << ("Введите xn > 0 \n"); //проверка на правильность введенного xn, если он верен , то цикл заканчивается

}// выводим в консоль текстовые данные и считываем введенные значения

while (1) {

cout<< ("Введите x конечное :\n"); cin >> xk;

if (xk>xn) { break; }

else cout << ("Введите xk > xn \n"); //проверка на правильность введенного xk, если он верен , то цикл заканчивается

}// выводим в консоль текстовые данные и считываем введенные значения

while (1) {

cout << ("Введите шаг :\n"); cin >> dx;

if (dx > 0) { break; }

else cout << ("Введите шаг > 0 \n"); ///проверка на правильность введенного шага, если он верен , то цикл заканчивается

}// выводим в консоль текстовые данные и считываем введенные значения

// считываем остальные переменные

cout << ("Введите a:\n"); cin >> a;

cout << ("Введите b:\n"); cin >> b;

x = xn;

cout << ("---------------------------------\n");

cout << ("| знач x | значение z |\n"); // заголовок таблицы

cout << ("---------------------------------\n");

while (x <= xk) {

if (x <= a) { // производим сравнение двух переменных

z = x - 2 \* pow(cos(x), 2); // Вычисляем значение для переменной исходя из условия

}

else // если не выполнилось условие то проверяем следующее условие

if ((a < x) && (x < b)) {

z = log(x) \* sin(x);// Вычисляем значение для переменной исходя из условия

}

else

{// если не выполнилось условие то выполняем действие

z = 1.3 + sin(x);// Вычисляем значение для переменной исходя из условия

}

cout << "| "<<x << " | "<<z <<" |\n" << endl;// Выводим полученный результат функции

x += dx; // изменяем x на шаг

}

cout << ("---------------------------------\n"); //закрываем таблицу

return 0;

}

**6.Результаты тестирования программы**

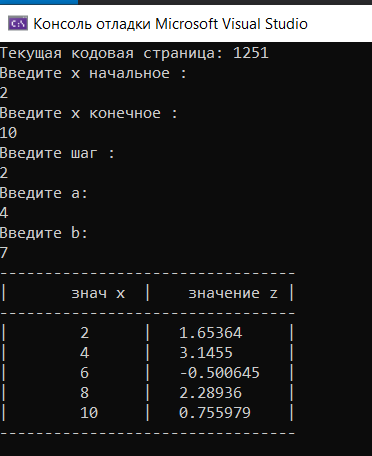
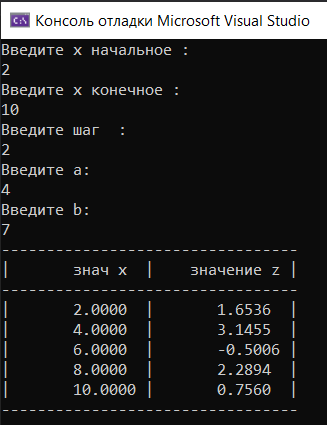
Для проверки правильности работы программы составим таблицу данных и полученный результат на основе ручных расчетов в калькуляторе. Так же необходимо учитывать, что программа выполняет все расчеты в радианах а не в градусах.

Для тестового примера возьмем такие параметры: xнач = 2, xкон = 10, dx=2, a = 4, b=7. Для этого примера вручную была просчитана таблица значений.

Таблица 1 – Таблица значений

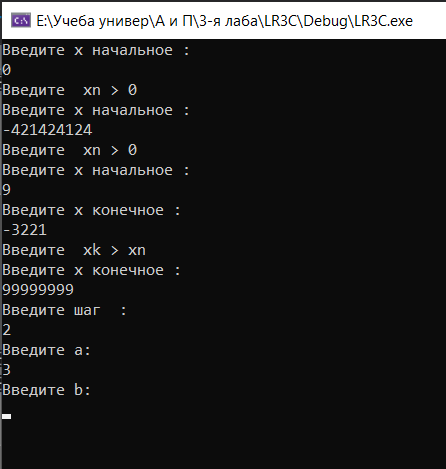
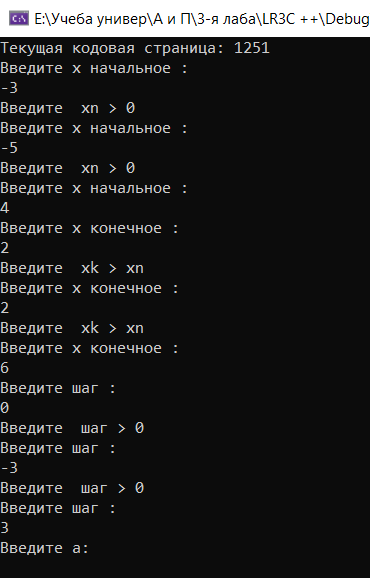
|  |  |  |
| --- | --- | --- |
| X | Функция исходя из условия | Z |
| 2 |  | 1,65364 |
| 4 |  | 3,14550 |
| 6 |  | -0,50065 |
| 8 |  | 2,28936 |
| 10 |  | 0,75598 |

Рисунок 3 – Результаты вычисления программ и вывод данных с помощью StdIO(слева) и IOStream (справа)



В результате тестирования, видно , что обе программы выполняют одинаковые расчеты и они совпадают с тестовым вариантом. Так же, видно , что программы не позволяют вбить значение xнач <=0, xнач> xкон , dx<=0 требуя повторный ввод значения.

Рисунок 4 - Ввод неверных данных в обе программы. IOStream (слева) , StdIO(справа)



**Вывод**

Мы познакомились с циклами while в C и C++.Узнали об использовании различных циклов для решенья каких либо задач, принцип их работы . Научились использовать циклы для проверки правильности ввода значения, построению структурных схем программ с использованием циклов. Так же освоили вывод данных в консоль в виде таблицы.