Министерство образования и науки Российской Федерации

Севастопольский государственный университет

Кафедра ИС

Отчет

по лабораторной работе № 5

«Программирование алгоритмов

циклической структуры»

Выполнил

ст. гр. И12д

Серегин А.В.

Проверил:

асс. Забаштанский А.К.

Севастополь

2015

1. Цель работы

Получение навыков программирования алгоритмов циклической структуры на языке С. Исследование эффективности применения различных видов циклов в задаче табулирования функции.

2. Варианты заданий

Вычислить и вывести на экран в виде таблицы значения функции на интервале от Хнач до Хкон с шагом dx. Таблицу снабдить заголовком и шапкой. Вид функции выбирать в соответствии с вариантами задания к лабораторной работе №1 настоящих методических указаний. Значения параметров a, b, Хнач, Хкон и dx вводятся с клавиатуры. Результаты вычислений выводятся в формате с фиксированной точкой.

Вариант 20:

3. Математическое обоснование

Если , то , иначе

если , то , иначе , где

ОДЗ:

4. Структурная схема

Ниже представлена структурная схема программы.



Рисунок 1 – структурная схема функции main.

5. Код программы

#include <iostream>

#include <iomanip>

#include <cstdio>

#include <conio.h>

#define Pi 3.1415926

int main()

{

setlocale(LC\_ALL, "Russian"); //отображение кириллицы

double x\_begin, x\_end;

double dx;

double a, b, z;

//форматы вывода

std::cout.setf(std::ios::fixed, std::ios::floatfield);

std::cout.setf(std::ios::left);

std::cout.precision(2);

std::cout << "Введите Xнач, Xкон, dx, a, b:" << std::endl;

std::cin >> x\_begin >> x\_end >> dx >> a >> b;

//Цикл многократного решения системы уравнений

for (double x(x\_begin); x <= x\_end; x += dx)

{

if (x >= b) // 3 ур-е

z = (1 + 2 \* x\*x + 3 \* x\*x\*x)\*

sinh(pow(x, pow(x, 1.5) + 7.3));

else

if (x != 0) // наличие корней

if (x <= a) //1 ур-е

z = abs(pow(x, 5.7 / x) - pow(atan(x) \*

180 / Pi, 0.2) + sin((x\*x)\*Pi / 180));

else //2 ур-е

z = log(exp(x\*x) + x\*log10(x) +

cos(x\*Pi / 180));

else // нет корней

{

std::cout << std::setw(10) << x << "Нет корней"

<< std::endl;

continue;

}

std::cout << std::setw(10) << x << z << std::endl;

}

system("pause");

return 0;

}

6. Тестовые примеры

На рисунке 2 приведен тестовый пример работы программы.

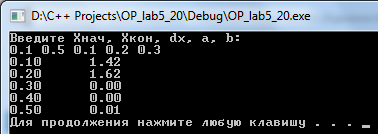


Рисунок 2 – Тест программы.

Выводы

Циклы – удобный способ для многократного повторения каких-либо действий. Благодаря циклам можно произвести огромное количество операций, написав при этом всего несколько строк кода. В представленном выше примере показан многократный вывод решения системы уравнений с помощью циклов.