**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ

УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ**(**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ**)» (**МАИ)**

Институт №3   
«Системы управления, информатика и  
электроэнергетика»

Кафедра 304   
Вычислительные машины, системы и сети

**ПРОГРАММИРОВАНИЕ   
НА ЯЗЫКАХ   
ВЫСОКОГО УРОВНЯ**

ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №13

Бригада №10

Выполнил:

Студент группы М30-211Б-19:

Брюхов И.А.

Проверил:

Дмитриева Е.А

Москва 2020

**Оглавление**

**Задание** **3**

**Схема алгоритма** **4-7**

**Текст программы** **8-10**

**Ожидаемый результат11-12**

**Полученный результат** **13**

**Вывод** **14**

Задание:

Вариант №4

Часть 1:

Разработать программу и подпрограмму (подпрограммы), вычисляющую

Методом хорд, корень уравнения f(x)=x^2-sin(5x), на отрезке от 0 до 2, с точностью ε = 0.001.

Методом хорд, найти корень уравнения f(x)=x^2-sin(5x), на отрезке от 0 до 2

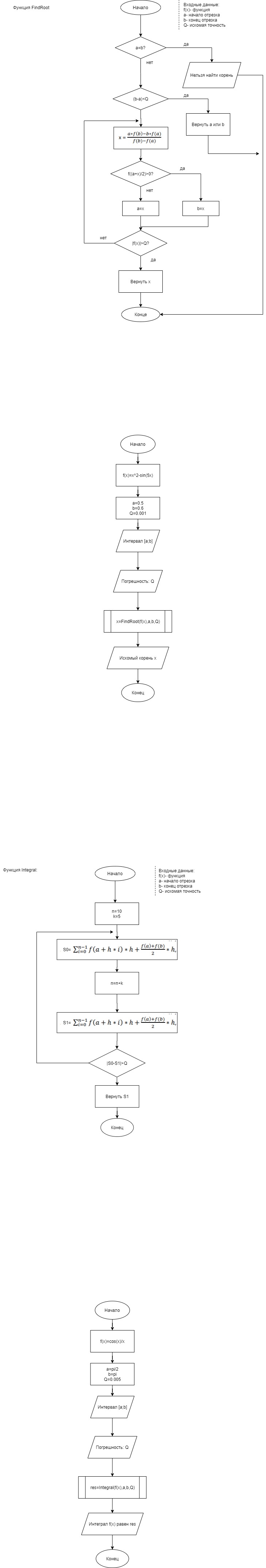
Часть 2:

Разработать программу и подпрограмму (подпрограммы), вычисляющую значение интеграла функции f(x)=cos(x)/x на отрезке от π/2 до π, с точностью ε = 0.0005. Интегрируемая функция и пределы интегрирования приведены в таблице, начальное число отрезков разбиения n = 10

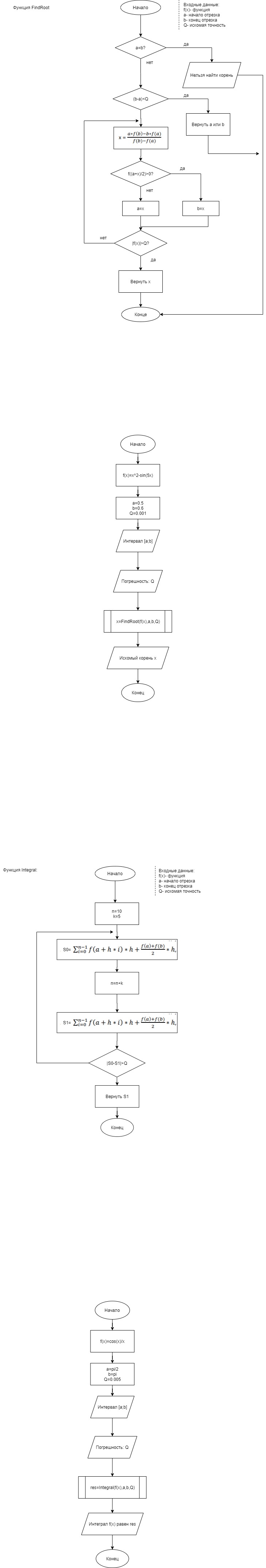
Схема алгоритма:

Часть 1:

Основная часть:

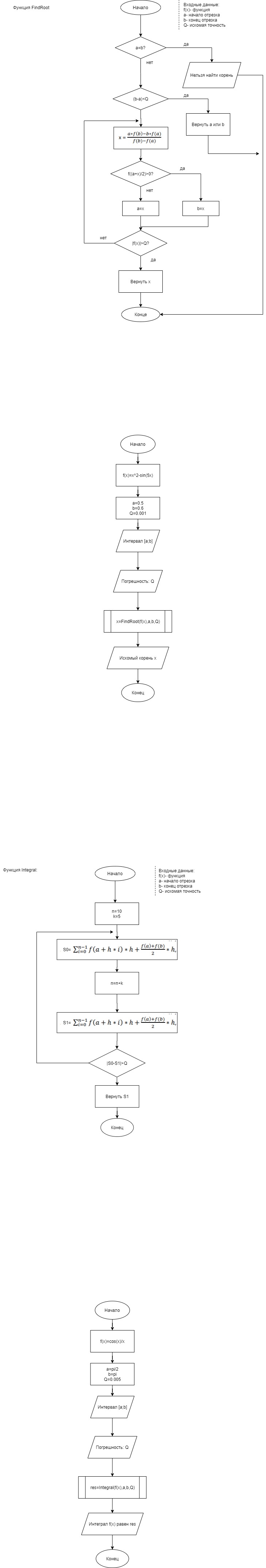


Определение функций:

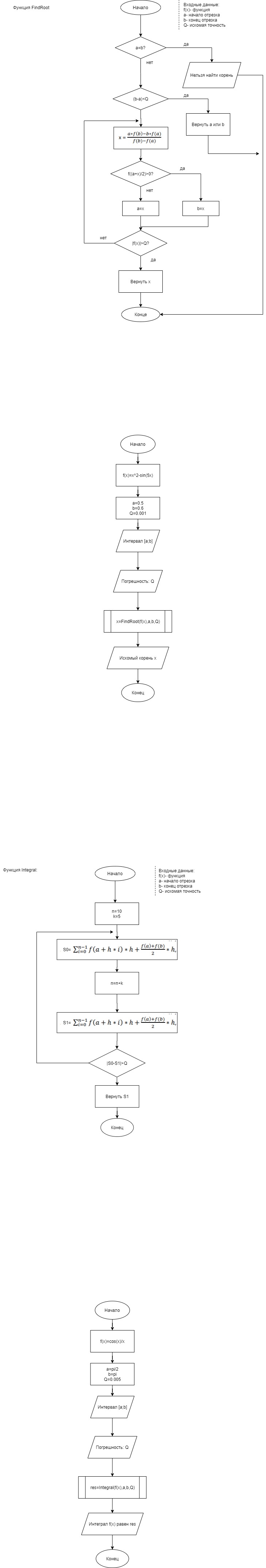


Часть 2:

Основная часть:



Определение функций:



Текст программы:

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*Project name: Laba13 \*

\*Project Type: Win32 Console Application \*

\*File Name:Laba13.ccp \*

\*Language: C++ \*

\*Programmer(1): Брюхов Илья; М3О-211Б-19; Вариант №4\*

\*Modified by:16.11.2020 \*

\*Created:14.11.2020 \*

\*Lastrevision: 16.11.2020 \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Часть 1:

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Подключаемые Библиотеки \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <iostream>

#include<cstdlib>

#include <math.h>

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Глобальные константы \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

const double Q = 0.001;//Погрешность

const double a =0.5;//Начало отрезка

const double b = 0.6;//Конец отрезка

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Определение функций \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

double f(double x);

//Функция поиска корня:

double FindRoot(double(\*f)(double), double a, double b, double Q);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Основная Программа \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int main() {

//Интервал, погрешность и корень:

double x;

cout << "interval:["<<a<<':'<<b<<"]\n";

//Проверка корректности интервала:

cout << "epsilon: "<<Q<<endl;

//Поиск решения:

x = FindRoot(f,a, b, Q);

cout << "x = " << x << endl;

return 0;

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Реализация функций \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

double f(double x) {

return x \* x - sin(5 \* x);

}

//Функция поиска корня:

double FindRoot(double(\*f)(double), double a, double b, double Q) {

double x;

if (a == b) {

cout << "Error. Root can`t be found\n";

return -1;

}

else if (b - a < Q) {

return a;

}

else {

do {

x = (a \* f(b) - b \* f(a)) / (f(b) - f(a));

if (f((a + x) / 2) > 0) {

b = x;

}

else a = x;

} while (fabs(f(x)) > Q);

}

return x;

}

**Часть 2:**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Подключаемые Библиотеки \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <iostream>

#include <math.h>

#include <locale.h>

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Глобальные константы \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

const double pi = 4\*atan(1);//Число пи

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Определение функций \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

double F(double x);//Заданная функция

double Integral(double(\*f)(double), double a, double b, double eps);//Функция поиска интеграла

double CalculateMamber(double(\*f)(double), double a, double b, int n);//Функция вычисления очередного приближения

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Основная Программа \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int main()

{

setlocale(LC\_ALL, "rus");

double a = pi / 2;

double b = pi;

const double eps = 0.005;

printf("Интервал [%.2f;%.2f]\n", a, b);

printf("Погрешность eps=%.3f\n", eps);

double res = Integral(F, a, b,eps);

printf("Интеграл f(x)=cos(x)/x на отрезке [%.2f,%.2f]= %f\n", a, b, res);

}//main

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Реализация функций \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

double F(double x) {

return cos(x) / x;

}//F

double Integral(double(\*f)(double), double a, double b, double eps) {

int n = 10;//количество на которое делится отрезок

int k = 5;//Увеличение разбиения отрезка за одну итерацию

double h;//шаг

double s0, s1;

do {

s0 = CalculateMamber(f, a, b, n);

n += k;

s1 = CalculateMamber(f, a, b, n);

} while (abs(s0 - s1) > eps);

return s1;

}//Integral

double CalculateMamber(double(\*f)(double), double a, double b, int n) {

double s = 0;

double h = (b - a) / n;

for (int i = 0; i < n; i++) {

s += f(a + h \* i) \* h;

}//for

s += ((f(a) + f(b)) / 2) \* h;

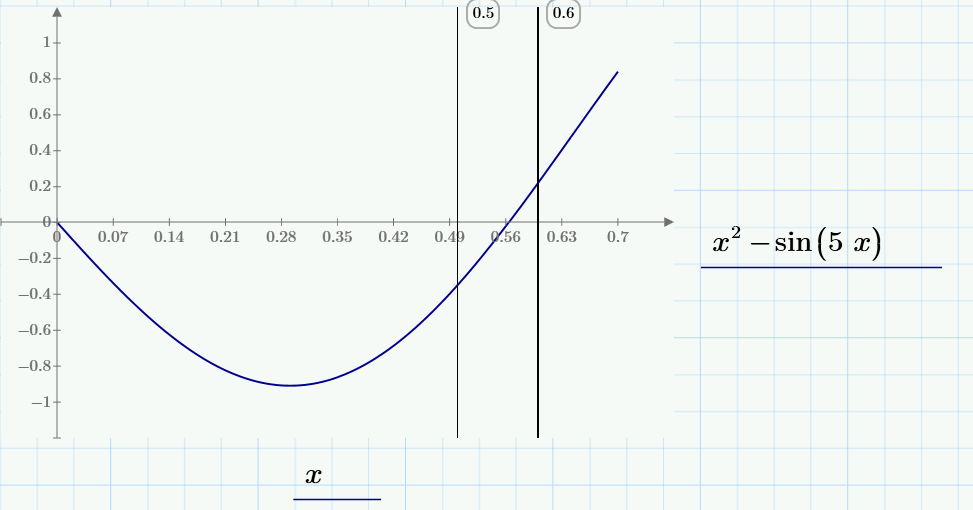
return s;

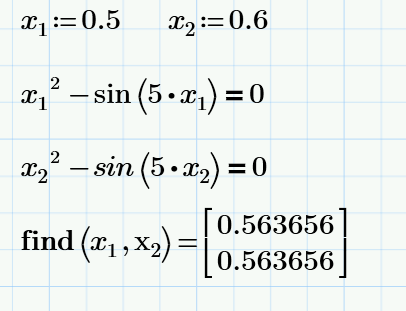
}//CalculateMamber

**Результат работы программы:**

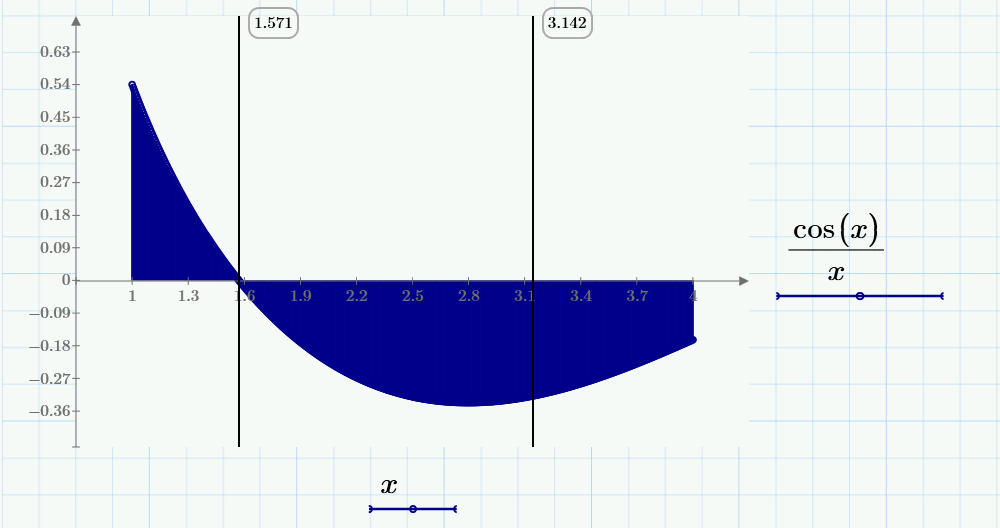
*Ожидаемый результат:*

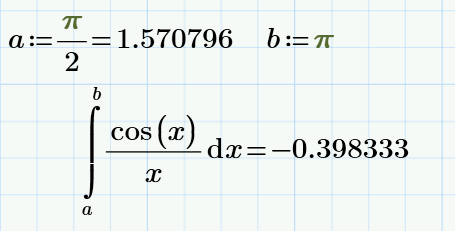
*Часть 1:*





*Часть 2:*

**



Ожидаемый результат был посчитан в программе Mathcad prime.

**Полученный результат:**

Часть 1:

**interval:[0.5:0.6]**

**epsilon: 0.001**

**x = 0.563598**

Часть 2:

**Интервал [1,57;3,14]**

**Погрешность eps=0,005**

**Интеграл f(x)=cos(x)/x на отрезке [1,57,3,14]= -0,397658**

Вывод:

Разработка программы завершена на том основании, что полученный результат совпадает с результатом вычисления mathlab с учетом заданной точности.