Вариант №4

Вычислить значения каждой функций при заданных значениях аргумента методом разложения в ряд Маклорена с точностью до (число n  аргумента - номер варианта студента). Для контроля вычислить значение функций при заданных значениях аргумента прямым способом.

1. y =  при x1 = 0,716 + 0,043n, x2 = 2,834 - 0,028n

x1 = 0,888

x2 = 2,722

= 1 + + + + ... +

f(x1) = = 2,430263

f(x2) = = 15,210711

2. y = ln(1 + x) при x = 0,122 + 0,018n

x = 0,194

ln(1 + x) = x - + + ... + ∙

f(x) = ln(1,194) = 0,177309

3. y = sin(x) при x1 = 0,232 + 0,012n, x2 = 0,747 - 0,014n

x1 = 0,28

x2 = 0,691

sin(x) = x - + + ... + ∙

f(x1) = sin(0,28) = 0,276356

f(x2) = sin(0,691) = 0,637308

4. y = cos(x) при x1 = 0,232 + 0,012n, x2 = 0,747 - 0,014n

x1 = 0,28

x2 = 0,691

cos(x) = 1 - + + ... + ∙

f(x1) = cos(0,28) = 0,961055

f(x2) = cos(0,691) = 0,770609

Код программы

#include <conio.h>

#include <iostream>

#include <vector>

#include <math.h>

#include <iomanip>

using namespace std;

void ex(double x)

{

unsigned long long factorial = 1;

double forX = x;

double res = 1.0;

double e = pow(10, -6);

int i = 1;

while (abs(forX / factorial) > e)

{

res = res + (forX / factorial);

forX \*= x;

factorial \*= (i + 1);

i++;

}

cout << fixed << setprecision(6) << res << endl;

}

void ln(double x)

{

double forX = x;

double res = 0.0;

int i = 0;

double e = pow(10, -6);

int sign = 1;

while (abs((sign \* forX) / (i + 1)) > e)

{

res = res + ((sign \* forX) / (i + 1));

forX \*= x;

sign = -sign;

i++;

}

cout << fixed << setprecision(6) << res << endl;

}

void sinx(double x)

{

double forX = x;

double res = 0.0;

int factorial = 1;

int i = 0;

double e = pow(10, -6);

int sign = 1;

while (abs((sign \* forX) / factorial) > e)

{

res = res + ((sign \* forX) / factorial);

forX \*= pow(x, 2);

i++;

factorial \*= (i + 1);

i++;

factorial \*= (i + 1);

sign = -sign;

}

cout << fixed << setprecision(6) << res << endl;

}

void cosx(double x)

{

double forX = 1;

double res = 1.0;

int factorial = 1;

int i = 1;

double e = pow(10, -6);

int sign = -1;

forX \*= pow(x, 2);

while (abs((sign \* forX) / factorial) > e)

{

factorial \*= (i + 1);

res = res + ((sign \* forX) / factorial);

forX \*= pow(x, 2);

i++;

factorial \*= (i + 1);

i++;

sign = -sign;

}

cout << fixed << setprecision(6) << res << endl;

}

int main()

{

double x1 = 0.888f, x2 = 2.722f, x3 = 0.194f, x4 = 0.28f, x5 = 0.691f;

cout << "1. y = e^x" << endl;

cout << " f(x1) = ";

ex(x1);

cout << " f(x2) = ";

ex(x2);

cout << "2. y = ln(1 + x)" << endl;

cout << " f(x) = ";

ln(x3);

cout << "3. y = sin(x)" << endl;

cout << " f(x1) = ";

sinx(x4);

cout << " f(x2) = ";

sinx(x5);

cout << "3. y = cos(x)" << endl;

cout << " f(x1) = ";

cosx(x4);

cout << " f(x2) = ";

cosx(x5);

\_getch();

}

Результат работы программы

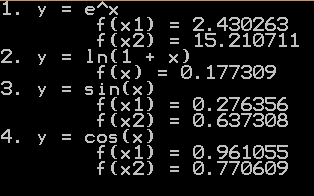


Рис. 1

Вывод: результаты вычисления прямым способом совпадают с результатами программы, следовательно, программа написана правильно.