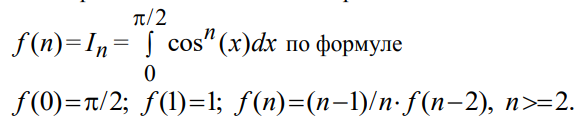
Практическая работа №8

Тема: «Функции с рекурсией»

**Задание 8-13**

Провести вычисление интеграла вида



#include <iostream>

#include <vector>

double integral(int n) {

if (n == 0) {

return 3.14 / 2;

}

else if (n == 1) {

return 1;

}

else {

return ((n - 1) / (double)n) \* integral(n - 2);

}

}

int main()

{

setlocale(LC\_ALL, "RU");

int n;

cout << "Лабораторная работа №8.\nЗадание 1. Введите n:";

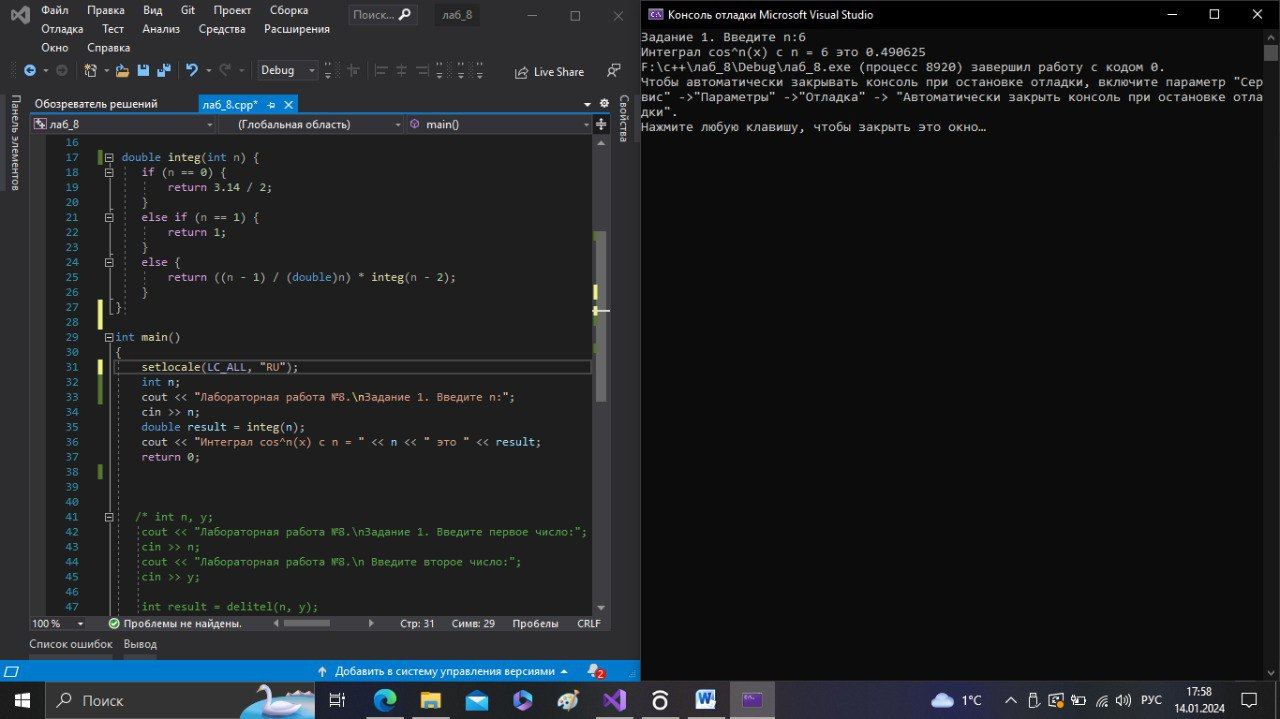
cin >> n;

double result = integral(n);

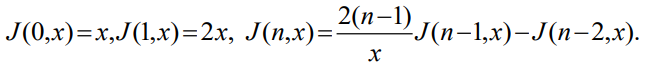
cout << "Интеграл cos^n(x) с n = " << n << " это " << result;

return 0;

}



**Задание 8-18**

Вычислить функцию Бесселя 8-го порядка с аргументом *x* 

#include <iostream>

using namespace std;

double beccel(double n, double x)

{

if (n == 0)

{

return x;

}

else if (n == 1)

{

return 2 \* x;

}

else

{

return ((2 \* (n - 1)) / x) \* beccel(n - 1, x) - beccel(n - 2, x);

}

}

int main()

{

setlocale(LC\_ALL, "RU");

double x;

cout << "Задание 2:\nВведите х = ";

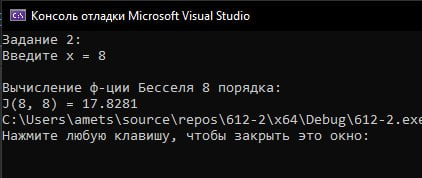
cin >> x;

double result = beccel(8, x);

cout << "\nВычисление ф-ции Бесселя 8 порядка: \nJ(8, " << x << ") = " << result;

return 0;

}



**Задание 8-24**

Найти корень заданного уравнения методом простой итерации с заданной точностью. Напомним, что в этом методе нужно уравнение свести к виду x=f(x) и очередное уточнение корня проводится по формуле xn+1=f(xn) до тех пор, пока |xn+1 – xn| > E, где Е – заданная точность. Рядом с уравнением в скобках указано начальное приближение корня.

#include <iostream>

#include <cmath>

using namespace std;

const double E = 0.001;

double f(double x) {

return cbrt(1000 - x);

}

double simpleIteration(double initialGuess) {

double currentGuess = initialGuess;

double nextGuess = f(currentGuess);

while (abs(nextGuess - currentGuess) > E) {

currentGuess = nextGuess;

nextGuess = f(currentGuess);

}

return nextGuess;

}

int main() {

setlocale(LC\_ALL, "RU");

cout << "Задание 3:\n";

double initialGuess = 9.42;

double root = simpleIteration(initialGuess);

cout << "Корень уравнения x^3 + x = 1000: " << root << endl;

return 0;

}

