Белорусский государственный технологический университет

Факультет информационных технологий

Кафедра программной инженерии

 Лабораторная работа №1

По дисциплине «Основы алгоритмизации и программирования»

На тему «Указатели на функции»

Выполнил:

Студент 1 курса 10 группы

Мандрик Алексей Иванович

Преподаватель: асс. Андронова М.В.

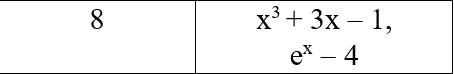
Минск, 2024

В соответствии со своим вариантом ***отделить корни*** двух уравнений и вычислить их методом ***дихотомии*** для исходных данных из таблицы, приведенной ниже. Точность вычислений принять равной **e** = 0,001 для всех вариантов.

Операторы метода вычисления корня оформить в виде ***функции пользователя***, уравнения записать также в виде ***функций пользователя***.

В главной функции предусмотреть ввод исходных данных, обращения к функции, реализующей метод дихотомии для двух уравнений. В процессе выполнения программы определить корни двух уравнений. Использовать ***указатель на функцию****.*

Вариант 8



#include <iostream>

#include <cmath> //Библиотека для работы с математическими операциями

using namespace std;

//Основное задание Вариант 8

double ur1(double x) // Функция пользователя для первого уравнения

{

return x \* x \* x + 3 \* x - 1;

}

double ur2(double x) // Функция пользователя для второго уравнения

{

return exp(x) – 4;

}

void dih(double a, double b, double(\*f)(double x)) // Функция реализации метода дихотомии

{

setlocale(LC\_ALL, "Rus");

const double epsilon = 1e-3;

double c;

while (b - a > epsilon) {

c = (a + b) / 2;

if (f(b) \* f(c) < 0)

a = c;

else

b = c;

}

cout << "Ответ:" << (a + b) / 2 << endl;

}

int main()

{

setlocale(LC\_ALL, "Rus");

double a, b;

cout << "Ввод a, b для первого уравнения" << endl;

cin >> a >> b;

dih(a, b, ur1); //Вызов функции

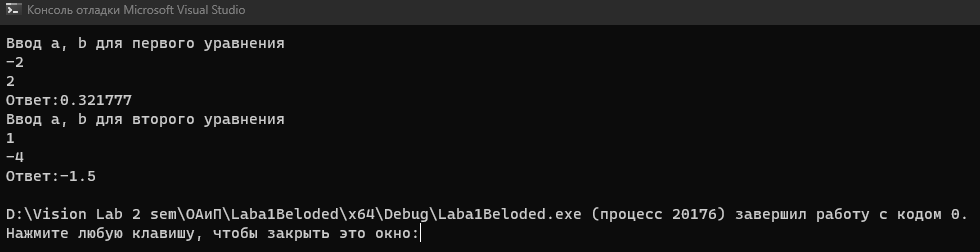
cout << "Ввод a, b для второго уравнения" << endl;

cin >> a >> b;

dih(a, b, ur2); //Вызов функции

return 0;

}



5. В соответствии со своим вариантом написать программы по условиям задач из таблицы ниже. Программа должна содержать функцию пользователя с ***переменным числом параметров*** и не менее трех обращений к ней с различным количеством параметров.



#include <iostream>

#include <cmath>

using namespace std;

int mn(int n, ...) // Функция с переменным числом параметров

{

int \*p = &n;

int Min = 1000000;

for (int i = 0; i < n; i++, \*(p+=2)) // Цикл для нахождения минимального числа

{

if (\*p < Min)

{

Min = \*p;

}

}

return Min; //Возврат значения

};

int main()

{

setlocale(LC\_ALL, "Rus");

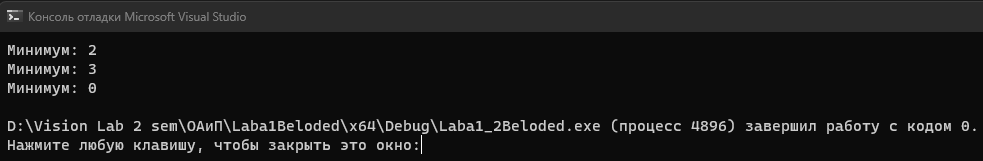
cout << "Минимум: " << mn(6, 3, 4, 5, 4, 2, 6 ) << endl;

cout << "Минимум: " << mn(3, 6, 8, 5) << endl;

cout << "Минимум: " << mn(5, 5, 3, 8, 0, 3) << endl;

return 0;

}



**Дополнительные задания**

В соответствии со своим вариантом ***отделить корни*** двух уравнений и вычислить их методом ***дихотомии*** для исходных данных из таблицы, приведенной ниже. Точность вычислений принять равной **e** = 0,001 для всех вариантов.

Операторы метода вычисления корня оформить в виде ***функции пользователя***, уравнения записать также в виде ***функций пользователя***.

В главной функции предусмотреть ввод исходных данных, обращения к функции, реализующей метод дихотомии для двух уравнений. В процессе выполнения программы определить корни двух уравнений. Использовать ***указатель на функцию****.*

**Вариант 1**

#include <iostream>

#include <cmath> //Библиотека для работы с математическими операциями

using namespace std;

//Основное задание Вариант 8

double ur1(double x) // Функция пользователя для первого уравнения

{

return x \* x + 4 \* x - 2;

}

double ur2(double x) // Функция пользователя для второго уравнения

{

return sin(x) + 0.1;

}

void dih(double a, double b, double(\*f)(double x)) // Функция реализации метода дихотомии

{

setlocale(LC\_ALL, "Rus");

const double epsilon = 1e-3;

double c;

while (b - a > epsilon) {

c = (a + b) / 2;

if (f(b) \* f(c) < 0)

a = c;

else

b = c;

}

cout << "Ответ:" << (a + b) / 2 << endl;

}

int main()

{

setlocale(LC\_ALL, "Rus");

double a, b;

cout << "Ввод a, b для первого уравнения" << endl;

cin >> a >> b;

dih(a, b, ur1); //Вызов функции

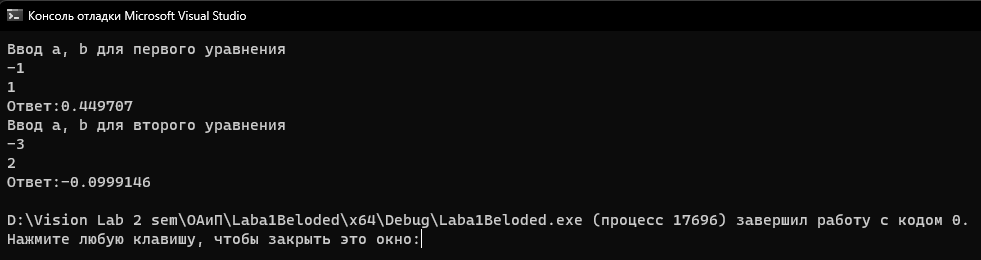
cout << "Ввод a, b для второго уравнения" << endl;

cin >> a >> b;

dih(a, b, ur2); //Вызов функции

return 0;

}



**Вариант 4**

#include <iostream>

#include <cmath> //Библиотека для работы с математическими операциями

using namespace std;

//Основное задание Вариант 8

double ur1(double x) // Функция пользователя для первого уравнения

{

return x \* x \* x + 2 \* x - 1

}

double ur2(double x) // Функция пользователя для второго уравнения

{

return exp(x) - 2;

}

void dih(double a, double b, double(\*f)(double x)) // Функция реализации метода дихотомии

{

setlocale(LC\_ALL, "Rus");

const double epsilon = 1e-3;

double c;

while (b - a > epsilon) {

c = (a + b) / 2;

if (f(b) \* f(c) < 0)

a = c;

else

b = c;

}

cout << "Ответ:" << (a + b) / 2 << endl;

}

int main()

{

setlocale(LC\_ALL, "Rus");

double a, b;

cout << "Ввод a, b для первого уравнения" << endl;

cin >> a >> b;

dih(a, b, ur1); //Вызов функции

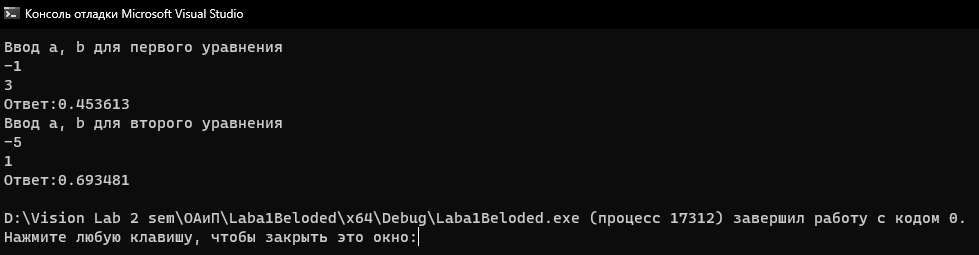
cout << "Ввод a, b для второго уравнения" << endl;

cin >> a >> b;

dih(a, b, ur2); //Вызов функции

return 0;

}



**Вариант 5**

#include <iostream>

#include <cmath> //Библиотека для работы с математическими операциями

using namespace std;

//Основное задание Вариант 8

double ur1(double x) // Функция пользователя для первого уравнения

{

return exp(x) - 3 - 1.0 / x;

}

double ur2(double x) // Функция пользователя для второго уравнения

{

return 0.2 - x \* x;

}

void dih(double a, double b, double(\*f)(double x)) // Функция реализации метода дихотомии

{

setlocale(LC\_ALL, "Rus");

const double epsilon = 1e-3;

double c;

while (b - a > epsilon) {

c = (a + b) / 2;

if (f(b) \* f(c) < 0)

a = c;

else

b = c;

}

cout << "Ответ:" << (a + b) / 2 << endl;

}

int main()

{

setlocale(LC\_ALL, "Rus");

double a, b;

cout << "Ввод a, b для первого уравнения" << endl;

cin >> a >> b;

dih(a, b, ur1); //Вызов функции

cout << "Ввод a, b для второго уравнения" << endl;

cin >> a >> b;

dih(a, b, ur2); //Вызов функции

return 0;

}



**Второе задание** Вариант 1, 4, 5

#include <iostream>

#include <cmath>

using namespace std;

int fmin(int n, ...) //Функция для нахождения минимального числа типа int

{

int\* p = &n;

int Min = 1000000;

p += 2;

for (int i = 0; i < n; i++, \*(p+=2)) //Цикл для перебора каждого числа

{

if (\*p < Min)

{

Min = \*p;

}

}

return Min;

};

int mn(int n, ...) //Функция для нахождения максимального числа типа int

{

int\* p = &n;

int Max = -1000000;

p += 2;

for (int i = 0; i < n; i++, \*(p+=2)) //Цикл для перебора каждого числа

{

if (\*p >= Max)

{

Max = \*p;

}

}

return Max;

}

int kvadr(int n, ...) //Функция для определения количества чисел, которые являются полными квадратами числа

{

int\* p = &n;

p += 2;

int count = 0;

for (int i = 0; i < n; i++, \*(p+=2)) //Цикл для перебора каждого числа

{

if (sqrt(\*p) == (int)sqrt(\*p))

{

count++;

}

}

return count;

}

int main()

{

setlocale(LC\_ALL, "Rus");

cout << "Минимум: " << fmin(6, 1, 2, 3, 4, 5, 0) << endl;

cout << "Минимум: " << fmin(5, 1, 2, 3, 4, 5) << endl;

cout << "Минимум: " << fmin(4, 1, 2, 3, 4) << endl;

cout << "Максимум: " << mn(6, 1, 2, 3, 4, 5, 0) << endl;

cout << "Максимум: " << mn(5, 1, 2, 3, 4, 5) << endl;

cout << "Максимум: " << mn(4, 1, 2, 3, 4) << endl;

cout << "Кол-во целых квадратов: " << kvadr(6, 1, 2, 3, 4, 5, 9) << endl;

cout << "Кол-во целых квадратов: " << kvadr(5, 1, 2, 3, 4, 5) << endl;

cout << "Кол-во целых квадратов: " << kvadr(4, 1, 2, 3, 4) << endl;

}

