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

Жамойдо Артём Игоревич

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 4**



#include <iostream>

using namespace std;

// Прототипы

double func1(double);

double func2(double);

double dychotomy(double (\*)(double), double, double, double);

double func1(double x) // Определение функции уравнения x^3 + 2x - 1

{

return pow(x,3) + 2 \* x - 1;

}

double func2(double x) // Определение функции уравнения e^x - 2

{

return exp(x) - 2;

}

double dychotomy(double (\*func)(double), double a, double b, double e) // Определение функции нахождения корней методом дихотомии

{

if (func(a) \* func(b) >= 0) // Неверный интервал

{

cout << "Неправильно выбраны начальные значения" << endl;

return -1; // Возвращаем -1, если интервал выбран неверно

}

double c{};

while (abs(a - b) > 2 \* e) // Метод дихотомии для нахождения корня уравнения

{

c = (a + b) / 2;

if (func(c) == 0)

{

break; // Найден точный корень

}

else if (func(c) \* func(a) <= 0)

{

b = c;

}

else

{

a = c;

}

}

return c; // Возвращаем найденный корень

}

int main()

{

setlocale(0, "ru");

double e = 0.001;

double a1, b1, a2, b2;

double r1, r2;

cout << "Введите интервалы для уравнения x^3 + 2x - 1 (a, b): ";

cin >> a1 >> b1;

cout << "Введите интервалы для уравнения e^x - 2 (a, b): ";

cin >> a2 >> b2;

r1 = dychotomy(func1, a1, b1, e); // Находим корень для уравнения x^3 + 2x - 1

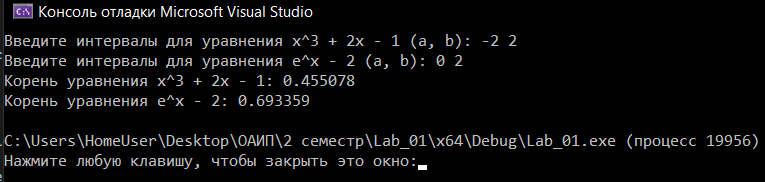
cout << "Корень уравнения x^3 + 2x - 1: " << r1 << endl;

r2 = dychotomy(func2, a2, b2, e); // Находим корень для уравнения e^x - 2

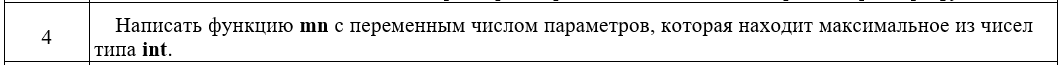
cout << "Корень уравнения e^x - 2: " << r2 << endl;

return 0;

}



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 <cstdarg> // Для использования va\_list

using namespace std;

int mn(int count, ...); // Прототип

int mn(int count, ...) // Функция нахождения максимального числа из переменного числа параметров типа int

{

va\_list arg; // Создаем список аргументов

va\_start(arg, count); // Инициализируем список аргументов

int max = va\_arg(arg, int); // Получаем первый аргумент и считаем его максимальным

for (int i = 1; i < count; ++i)

{

int num = va\_arg(arg, int); // Получаем следующий аргумент

max = (num > max) ? num : max; // Обновляем максимальное число

}

va\_end(arg); // Освобождаем список аргументов

return max;

}

int main()

{

setlocale(0, "ru");

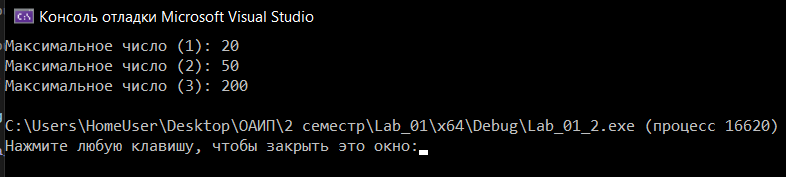
cout << "Максимальное число (1): " << mn(3, 10, 20, 5) << endl; // Первый вызов с тремя параметрами

cout << "Максимальное число (2): " << mn(5, 30, 50, 15, 25, 40) << endl; // Второй вызов с пятью параметрами

cout << "Максимальное число (3): " << mn(4, 100, 200, 150, 75) << endl; // Третий вызов с четырьмя параметрами

return 0;

}



Доп. задачи

**Вариант 8**



#include <iostream>

using namespace std;

// Прототипы

double func1(double);

double func2(double);

double dychotomy(double (\*)(double), double, double, double);

double func1(double x) // Определение функции уравнения x^3 + 3x - 1

{

return pow(x, 3) + 3 \* x - 1;

}

double func2(double x) // Определение функции уравнения e^x - 4

{

return exp(x) - 4;

}

double dychotomy(double (\*func)(double), double a, double b, double e) // Определение функции нахождения корней методом дихотомии

{

if (func(a) \* func(b) >= 0) // Неверный интервал

{

cout << "Неправильно выбраны начальные значения" << endl;

return -1; // Возвращаем -1, если интервал выбран неверно

}

double c{};

while (abs(a - b) > 2 \* e) // Метод дихотомии для нахождения корня уравнения

{

c = (a + b) / 2;

if (func(c) == 0)

{

break; // Найден точный корень

}

else if (func(c) \* func(a) <= 0)

{

b = c;

}

else

{

a = c;

}

}

return c; // Возвращаем найденный корень

}

int main()

{

setlocale(0, "ru");

double e = 0.001;

double a1, b1, a2, b2;

double r1, r2;

cout << "Введите интервалы для уравнения x^3 + 3x - 1 (a, b): ";

cin >> a1 >> b1;

cout << "Введите интервалы для уравнения e^x - 4 (a, b): ";

cin >> a2 >> b2;

r1 = dychotomy(func1, a1, b1, e); // Находим корень для уравнения x^3 + 3x - 1

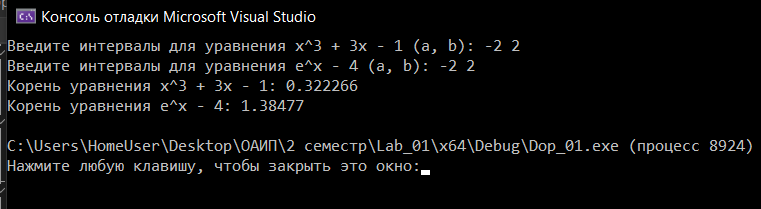
cout << "Корень уравнения x^3 + 3x - 1: " << r1 << endl;

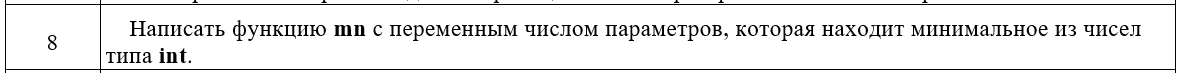
r2 = dychotomy(func2, a2, b2, e); // Находим корень для уравнения e^x - 4

cout << "Корень уравнения e^x - 4: " << r2 << endl;

return 0;

}





#include <iostream>

#include <cstdarg> // Для использования va\_list

using namespace std;

int mn(int count, ...); // Прототип

int mn(int count, ...) // Функция нахождения минимального числа из переменного числа параметров типа int

{

va\_list arg; // Создаем список аргументов

va\_start(arg, count); // Инициализируем список аргументов

int min = va\_arg(arg, int); // Получаем первый аргумент и считаем его минимальным

for (int i = 1; i < count; ++i)

{

int num = va\_arg(arg, int); // Получаем следующий аргумент

min = (num < min) ? num : min; // Обновляем минимальное число

}

va\_end(arg); // Освобождаем список аргументов

return min;

}

int main()

{

setlocale(0, "ru");

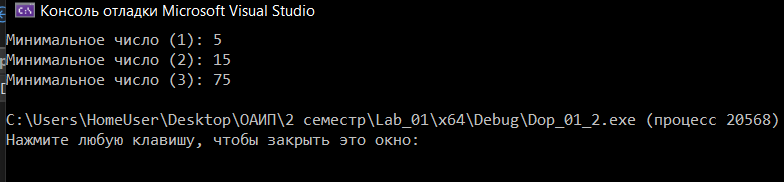
cout << "Минимальное число (1): " << mn(3, 10, 20, 5) << endl; // Первый вызов с тремя параметрами

cout << "Минимальное число (2): " << mn(5, 30, 50, 15, 25, 40) << endl; // Второй вызов с пятью параметрами

cout << "Минимальное число (3): " << mn(4, 100, 200, 150, 75) << endl; // Третий вызов с четырьмя параметрами

return 0;

}



**Вариант 1**



#include <iostream>

using namespace std;

// Прототипы

double func1(double);

double func2(double);

double dychotomy(double (\*)(double), double, double, double);

double func1(double x) // Определение функции уравнения x^2 + 4x - 2

{

return pow(x, 2) + 4 \* x - 2;

}

double func2(double x) // Определение функции уравнения sin(x) + 0.1

{

return sin(x) + 0.1;

}

double dychotomy(double (\*func)(double), double a, double b, double e) // Определение функции нахождения корней методом дихотомии

{

if (func(a) \* func(b) >= 0) // Неверный интервал

{

cout << "Неправильно выбраны начальные значения" << endl;

return -1; // Возвращаем -1, если интервал выбран неверно

}

double c{};

while (abs(a - b) > 2 \* e) // Метод дихотомии для нахождения корня уравнения

{

c = (a + b) / 2;

if (func(c) == 0)

{

break; // Найден точный корень

}

else if (func(c) \* func(a) <= 0)

{

b = c;

}

else

{

a = c;

}

}

return c; // Возвращаем найденный корень

}

int main()

{

setlocale(0, "ru");

double e = 0.001;

double a1, b1, a2, b2;

double r1, r2;

cout << "Введите интервалы для уравнения x^2 + 4x - 2 (a, b): ";

cin >> a1 >> b1;

cout << "Введите интервалы для уравнения sin(x) + 0.1 (a, b): ";

cin >> a2 >> b2;

r1 = dychotomy(func1, a1, b1, e); // Находим корень для уравнения x^2 + 4x - 2

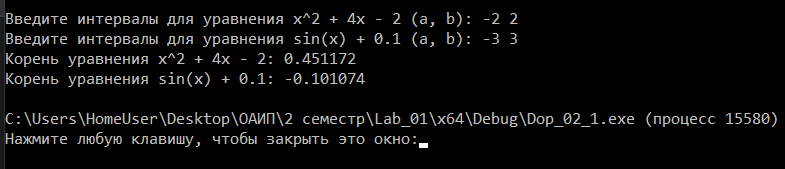
cout << "Корень уравнения x^2 + 4x - 2: " << r1 << endl;

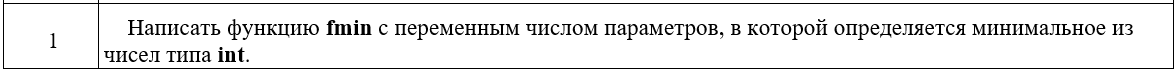
r2 = dychotomy(func2, a2, b2, e); // Находим корень для уравнения sin(x) + 0.1

cout << "Корень уравнения sin(x) + 0.1: " << r2 << endl;

return 0;

}





#include <iostream>

#include <cstdarg> // Для использования va\_list

using namespace std;

int fmin(int count, ...); // Прототип

int fmin(int count, ...) // Функция нахождения минимального числа из переменного числа параметров типа int

{

va\_list arg; // Создаем список аргументов

va\_start(arg, count); // Инициализируем список аргументов

int min = va\_arg(arg, int); // Получаем первый аргумент и считаем его минимальным

for (int i = 1; i < count; ++i)

{

int num = va\_arg(arg, int); // Получаем следующий аргумент

min = (num < min) ? num : min; // Обновляем минимальное число

}

va\_end(arg); // Освобождаем список аргументов

return min;

}

int main()

{

setlocale(0, "ru");

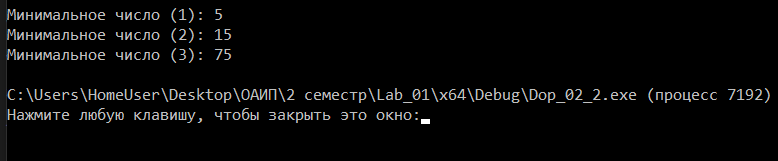
cout << "Минимальное число (1): " << fmin(3, 10, 20, 5) << endl; // Первый вызов с тремя параметрами

cout << "Минимальное число (2): " << fmin(5, 30, 50, 15, 25, 40) << endl; // Второй вызов с пятью параметрами

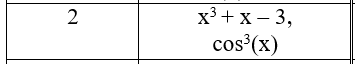
cout << "Минимальное число (3): " << fmin(4, 100, 200, 150, 75) << endl; // Третий вызов с четырьмя параметрами

return 0;

}



**Вариант 2**

****

#include <iostream>

using namespace std;

// Прототипы

double func1(double);

double func2(double);

double dychotomy(double (\*)(double), double, double, double);

double func1(double x) // Определение функции уравнения x^3 + x - 3

{

return pow(x, 3) + x - 3;

}

double func2(double x) // Определение функции уравнения cos^3(x)

{

return pow(cos(x), 3);

}

double dychotomy(double (\*func)(double), double a, double b, double e) // Определение функции нахождения корней методом дихотомии

{

if (func(a) \* func(b) >= 0) // Неверный интервал

{

cout << "Неправильно выбраны начальные значения" << endl;

return -1; // Возвращаем -1, если интервал выбран неверно

}

double c{};

while (abs(a - b) > 2 \* e) // Метод дихотомии для нахождения корня уравнения

{

c = (a + b) / 2;

if (func(c) == 0)

{

break; // Найден точный корень

}

else if (func(c) \* func(a) <= 0)

{

b = c;

}

else

{

a = c;

}

}

return c; // Возвращаем найденный корень

}

int main()

{

setlocale(0, "ru");

double e = 0.001;

double a1, b1, a2, b2;

double r1, r2;

cout << "Введите интервалы для уравнения x^3 + x - 3 (a, b): ";

cin >> a1 >> b1;

cout << "Введите интервалы для уравнения cos^3(x) (a, b): ";

cin >> a2 >> b2;

r1 = dychotomy(func1, a1, b1, e); // Находим корень для уравнения x^3 + x - 3

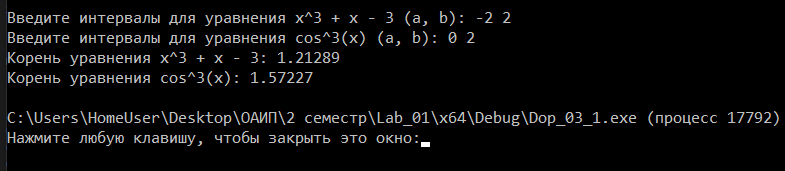
cout << "Корень уравнения x^3 + x - 3: " << r1 << endl;

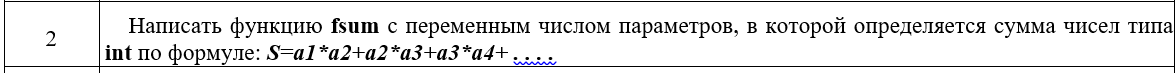
r2 = dychotomy(func2, a2, b2, e); // Находим корень для уравнения cos^3(x)

cout << "Корень уравнения cos^3(x): " << r2 << endl;

return 0;

}

****

****

#include <iostream>

#include <cstdarg>

using namespace std;

int fsum(int count, ...); // Прототип

int fsum(int count, ...) // Функция fsum с переменным числом параметров

{

va\_list arg; // Создаем список аргументов

va\_start(arg, count); // Инициализируем список аргументов

int sum = 0;

int current = va\_arg(arg, int); // Получаем первый аргумент из списка

for (int i = 1; i < count; i++)

{

int next = va\_arg(arg, int); // Получаем следующий аргумент из списка

sum += current \* next; // Умножаем текущий и следующий аргументы и добавляем результат к сумме

current = next;

}

va\_end(arg); // Освобождаем список аргументов

return sum;

}

int main()

{

setlocale(0, "ru");

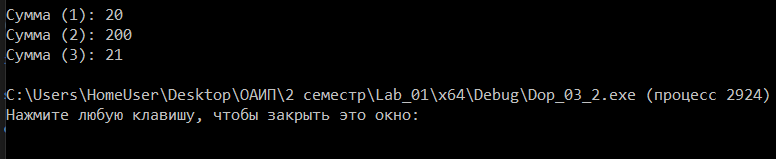
cout << "Сумма (1): " << fsum(4, 1, 2, 3, 4) << endl; // Первый вызов с четырьмя параметрами

cout << "Сумма (2): " << fsum(3, 5, 10, 15) << endl; // Второй вызов с тремя параметрами

cout << "Сумма (3): " << fsum(2, 3, 7) << endl; // Третий вызов с двумя параметрами

return 0;

}

****