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(а):

Студент(ка) 1 курса 8 группы

Семёнов Даниил Вячеславович

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 10



#include <iostream>

using namespace std;

double firstEqua(double x) //функция возвращающая первое уравнение

{

return (sin(x) + x \* x \* x);

}

double secondEqua(double x) //функция возвращающая второе уравнение

{

return (0.4 \* x \* x \* x);

}

double useDichotomy(double(\*funct)(double), double a, double b, double e) // функция содержащая алгоритм метода дихотомии

{

double x,fx,fa;

// алгоритм взят из лабораторной номер 9, 1 семестра

do {

x = (a + b) / 2;

fx = funct(x);

fa = funct(a);

if (fx \* fa <= 0)

b = x;

else

a = x;

} while (abs(a - b) > 2 \* e);

return(x);

}

int main()

{

setlocale(LC\_ALL, "rus");

double a, b, e = 0.0001;

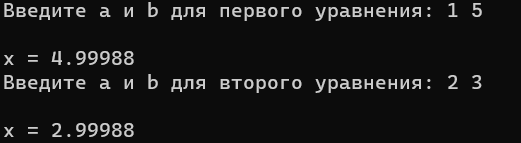
cout << "Введите a и b для первого уравнения: "; cin >> a >> b; // вводим необходимые значения для первого уравнения

cout << endl<< "x = " << useDichotomy(firstEqua, a, b, e) <<endl; // вызываем функцию с методом дихотомии

cout << "Введите a и b для второго уравнения: "; cin >> a >> b; // вводим необходимые значения

cout << endl<<"x = " << useDichotomy(secondEqua, a, b, e); // вызываем функцию с методом дихотомии

}



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> // подключаем библиотеку для работы с вводом/выводом

#include <cstdarg> // подключаем библиотеку для работы с функциями с произвольным количеством аргументов

**using** **namespace** std**;**

int findWord**(**int n**,** **...)** // функция для нахождения необходимого слова

**{**

va\_list args**;** // Объявляем объект типа va\_list для работы с переменными аргументами

va\_start**(**args**,** n**);** // Получаем текущий аргумент из списка

char word**[**100**];** // массив для хранения текущего слова

char strwords**[**600**];** // массив для вывода отсортированной строки

**for** **(**int i **=** 1**;** i **<=** n**;** i**++)**

**{**

memset**(**word**,** 0**,** **sizeof(**word**));** // заполняем массив 0, чтобы не выводился мусор

memset**(**strwords**,** 0**,** **sizeof(**strwords**));** // заполняем массив 0, чтобы не выводился мусор

int len **=** 0**;** // переменная для хранеия длины текущего слова

const char**\*** newstr **=** va\_arg**(**args**,** const char**\*);** // Получаем текущий аргумент из списка

char symb **=** 'a'**;** // создаём переменную для проверки слов

int newsymb **=** symb**;**

bool f **=** **true;**

int len2 **=** 0**;** // переменная для контроля длины отсортированной строки

int w **=** 0**;**

**while** **(**w **<** 5**)** // цикл для заполнения нашими словами строки

**{**

**while** **(**f**)** // цикл для нахождения необходимого слова

**{**

**for** **(**int j **=** 0**;**newstr**[**j**]** **!=** '\0'**;**j**++)** // цикл для того чтобы просмотреть всю строку

**{**

**if** **(**newstr**[**j**]** **==** newsymb **&&** newstr**[**j **-** 1**]** **==** ' '**)** // условие для нахождения необходимого слова

**{**

int m **=** j**;** // создаём переменную для того чтобы знать какие элементы массива нам необходимы

**for** **(**int k **=** 0**;**newstr**[**m**]** **!=** ' '**;**k**++,** m**++)** // цикл для заполнения массива найденным словом

**{**

word**[**k**]** **=** newstr**[**m**];**

**}**

len **=** m**;**

f **=** **false;**

**}**

**}**

newsymb **=** newsymb **+** 1**;** // увеличиваем значение переменной для просмотра следующей буквы по алфавиту

**}**

**for** **(**int k **=** 0**;**k **<** len**;**k**++,** len2**++)** // цикл для заполнения строки найденным словом

**{**

strwords**[**len2**]** **=** word**[**k**];**

**}**

f **=** **true;**

strwords**[**len2**]** **=** ' '**;** // добавляем пробел после слова

len2**++;**

w**++;**

**}**

cout **<<** endl**;**

**for** **(**int i **=** 0**;**i **<** len2**;**i**++)**

**{**

cout **<<** strwords**[**i**];** // выводим нашу строку

**}**

cout **<<** endl**;**

**}**

cout **<<** endl**;**

va\_end**(**args**);** // Завершаем работу с объектом va\_list

**return** 0**;**

**}**

int main**()**

**{**

// создаем три массива содержащих строки

char str1**[]** **=** " bghj agfj cdfa ehlhksgdh dldglhfd "**;**

char str2**[]** **=** " ehgh bdggfj ajhkgs dhlkdfj cldkgjh "**;**

char str3**[]** **=** " dhhhghj csgfdsgj edjgfa bgddkg ahf "**;**

// вызываем функции

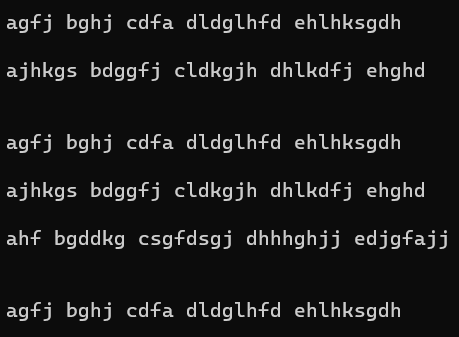
findWord**(**2**,** str1**,** str2**);**

findWord**(**3**,** str1**,** str2**,** str3**);**

findWord**(**1**,** str1**);**

**return** 0**;**

**}**



Доп. Задачи

Вариант 1



// задания отличаются лишь уравнениями

#include <iostream>

using namespace std;

double firstEqua(double x)

{

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

}

double secondEqua(double x)

{

return (sin(x)+0.1);

}

double useDichotomy(double(\*funct)(double), double a, double b, double e)

{

double x, fx, fa, a1;

do {

x = (a + b) / 2;

fx = funct(x);

fa = funct(a);

if (fx \* fa <= 0)

b = x;

else

a = x;

} while (abs(a - b) > 2 \* e);

return(x);

}

int main()

{

setlocale(LC\_ALL, "rus");

double a, b, e = 0.0001;

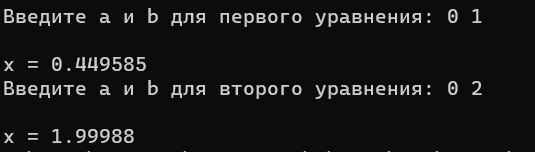
cout << "Введите a и b для первого уравнения: "; cin >> a >> b;

cout << endl << "x = " << useDichotomy(firstEqua, a, b, e) << endl;

cout << "Введите a и b для второго уравнения: "; cin >> a >> b;

cout << endl << "x = " << useDichotomy(secondEqua, a, b, e);

}





#include <iostream>

#include <cstdarg>

using namespace std;

void fmin(int n, ...)

{

va\_list arg; // Объявляем объект типа va\_list для работы с переменными аргументами

va\_start(arg, n); // Получаем текущий аргумент из списка

int min = va\_arg(arg, int); // Присваем переменной первый аргумент из списка

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

{

int val;

val = va\_arg(arg, int); // Получаем текущий аргумент из списка

if (val < min) // условие для проверки

{

min = val;

}

}

cout <<"Минимальное значение: " << min << endl;

va\_end(arg); // Завершаем работу с объектом va\_list

}

int main()

{

setlocale(LC\_ALL, "rus");

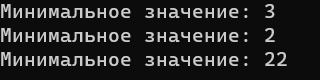
//вызываем функции с переменным числом параметров

fmin(5, 10,12,4, 3, 9);

fmin(4, 15, 3, 2, 12);

fmin(6, 320, 52, 148, 22,34, 30);

}



Вариант 4



#include <iostream>

using namespace std;

double firstEqua(double x)

{

return (x \* x \*x + 2 \* x - 1);

}

double secondEqua(double x)

{

return (exp(x) - 2);

}

double useDichotomy(double(\*funct)(double), double a, double b, double e)

{

double x, fx, fa;

do {

x = (a + b) / 2;

fx = funct(x);

fa = funct(a);

if (fx \* fa <= 0)

b = x;

else

a = x;

} while (abs(a - b) > 2 \* e);

return(x);

}

int main()

{

setlocale(LC\_ALL, "rus");

double a, b, e = 0.0001;

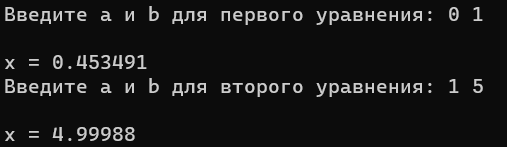
cout << "Введите a и b для первого уравнения: "; cin >> a >> b;

cout << endl << "x = " << useDichotomy(firstEqua, a, b, e) << endl;

cout << "Введите a и b для второго уравнения: "; cin >> a >> b;

cout << endl << "x = " << useDichotomy(secondEqua, a, b, e);

}





#include <iostream>

#include <cstdarg>

using namespace std;

void mn(int n, ...)

{

va\_list arg; // Объявляем объект типа va\_list для работы с переменными аргументами

va\_start(arg, n); // Получаем текущий аргумент из списка

int max = va\_arg(arg, int); // Присваем переменной первый аргумент из списка

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

{

int val;

val = va\_arg(arg, int); // Получаем текущий аргумент из списка

if (val > max) // Условие для проверки

{

max = val;

}

}

cout << "Максимальное значение: " << max << endl;

va\_end(arg); // Завершаем работу с объектом va\_list

}

int main()

{

setlocale(LC\_ALL, "rus");

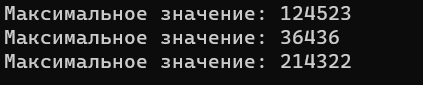
// Вызываем функции с переменным числом параметров

mn(3, 1560, 5126, 124523);

mn(4, 15236, 36436, 7124, 12235);

mn(5, 23250, 2575, 175478, 214322, 6345);

}



Вариант 8



#include <iostream>

using namespace std;

double firstEqua(double x)

{

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

}

double secondEqua(double x)

{

return (exp(x) - 4);

}

double useDichotomy(double(\*funct)(double), double a, double b, double e)

{

double x, fx, fa;

do {

x = (a + b) / 2;

fx = funct(x);

fa = funct(a);

if (fx \* fa <= 0)

b = x;

else

a = x;

} while (abs(a - b) > 2 \* e);

return(x);

}

int main()

{

setlocale(LC\_ALL, "rus");

double a, b, e = 0.0001;

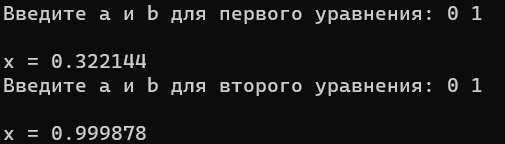
cout << "Введите a и b для первого уравнения: "; cin >> a >> b;

cout << endl << "x = " << useDichotomy(firstEqua, a, b, e) << endl;

cout << "Введите a и b для второго уравнения: "; cin >> a >> b;

cout << endl << "x = " << useDichotomy(secondEqua, a, b, e);

}





#include <iostream>

#include <cstdarg>

using namespace std;

void mn(int n, ...)

{

va\_list arg; // Объявляем объект типа va\_list для работы с переменными аргументами

va\_start(arg, n); // Получаем текущий аргумент из списка

int min = va\_arg(arg, int); // Присваем переменной первый аргумент из списка

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

{

int val;

val = va\_arg(arg, int); // Получаем текущий аргумент из списка

if (val < min) // условие для проверки

{

min = val;

}

}

cout << "Минимальное значение: " << min << endl;

va\_end(arg); // Завершаем работу с объектом va\_list

}

int main()

{

setlocale(LC\_ALL, "rus");

//вызываем функции с переменным числом параметров

mn(5, 10, 12, 4, 3, 9);

mn(4, 15, 3,62, 1762);

mn(6, 36520, 5234, 1448, 2276, 56,65);

}

