Белорусский государственный технологический университет

Факультет информационных технологий

Кафедра программной инженерии

 Лабораторная работа 1

По дисциплине «Основы алгоритмизации и программирования»

На тему «Указатели на функции»

 Выполнила:

Студентка1 курса 6 группы

Альшевская Алина Михайловна

Преподаватель: асс. Андронова М.В.

2023, Минск

Вариант 1

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****.*



#include <iostream>

#include <math.h>

double function2(double x){

return (sin(x)+0.1);//сама функция

}

double function1(double x) {

return (x \* x + 4 \* x - 2);//сама функция

}

double counter(double a, double b,double e ,double(\*f1)(double)) {

double x;

while (fabs(a - b) > 2 \* e)//используем цикл while.Если условие в скобках истинно, то выполняются операторы цикла до тех пор, пока оно не станет ложным.

{

x = (a + b) / 2;

if (f1(x) \* f1(a) <= 0)//елси уловие с вобках верно то b=x

b = x;

else // иначе a=x

a = x;

}

return x;

}

int main()

{

setlocale(LC\_CTYPE, "Russian");

double e = 0.0001, x,a,b;

std::cout << "Введите ограничение ";

std::cin >> a;

std::cout << "Введите ограничение ";

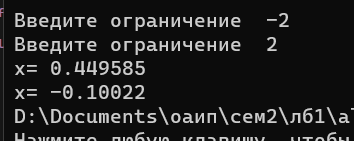
std::cin >> b;

std::cout << "x= " << counter(a, b, e, function1) << std::endl;

std::cout << "x= " << counter(a, b, e, function2);

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>

int fmin(int n, ...);

int main()

{

std::cout<< fmin (6, 4, 5, 1, 2, 3, 0) << std::endl;//используем фунцкцию

std::cout<< fmin (2, 34, 4445);//используем функцию

}

int fmin (int n, ...)

{

int \*p = &n;

int min = \*(++p) ;// в качестве наименьшего сначала выбираем первое значение

for (int i = 1; i <= n; i++) {// используя цикл находим меньшее значение

p++;

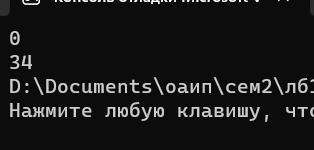
if (min > \*p)

min = \*p;

}

return min;// возвращаем результат

}



Дополнительные задания

Вариант 4



#include <iostream>

#include <math.h>

double function2(double x) {

return (exp(x) -2);//сама функция

}

double function1(double x) {

return (x \* x\*x + 2 \* x - 1);//сама функция

}

double counter(double a, double b, double e, double(\*f1)(double)) {

double x;

while (fabs(a - b) > 2 \* e)//используем цикл while.Если условие в скобках истинно, то выполняются операторы цикла до тех пор, пока оно не станет ложным.

{

x = (a + b) / 2;

if (f1(x) \* f1(a) <= 0)//елси уловие с вобках верно то b=x

b = x;

else // иначе a=x

a = x;

}

return x;

}

int main()

{

setlocale(LC\_CTYPE, "Russian");

double e = 0.0001, x, a, b;

std::cout << "Введите ограничение ";

std::cin >> a;

std::cout << "Введите ограничение ";

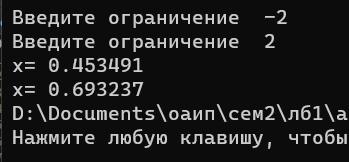
std::cin >> b;

std::cout << "x= " << counter(a, b, e, function1) << std::endl;

std::cout << "x= " << counter(a, b, e, function2);

return 0;

}





#include <iostream>

int mn(int n, ...);

int main()

{

std::cout << mn(6, 4, 5, 1, 2, 3, 0) << std::endl;//используем фунцкцию

std::cout << mn(2, 34, 4445);//используем функцию

}

int mn(int n, ...)

{

int\* p = &n;

int max = \*(++p);// в качестве наибольшего сначала выбираем первое значение

for (int i = 1; i < n; i++) {// используя цикл находим большее значение

p++;

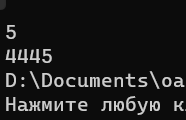
if (max < \*p)

max = \*p;

}

return max;// возвращаем результат

}



Вариант 8



#include <iostream>

#include <math.h>

double function2(double x) {

return (exp(x)-4);//сама функция

}

double function1(double x) {

return (x \* x\*x + 3 \* x - 1);//сама функция

}

double counter(double a, double b, double e, double(\*f1)(double)) {

double x;

while (fabs(a - b) > 2 \* e)//используем цикл while.Если условие в скобках истинно, то выполняются операторы цикла до тех пор, пока оно не станет ложным.

{

x = (a + b) / 2;

if (f1(x) \* f1(a) <= 0)//елси уловие с вобках верно то b=x

b = x;

else // иначе a=x

a = x;

}

return x;

}

int main()

{

setlocale(LC\_CTYPE, "Russian");

double e = 0.0001, x, a, b;

std::cout << "Введите ограничение ";

std::cin >> a;

std::cout << "Введите ограничение ";

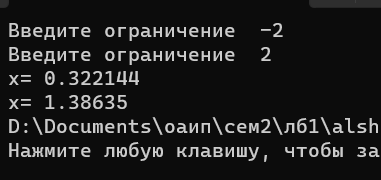
std::cin >> b;

std::cout << "x= " << counter(a, b, e, function1) << std::endl;

std::cout << "x= " << counter(a, b, e, function2);

return 0;

}





#include <iostream>

int mn(int n, ...);

int main()

{

std::cout << mn(6, 4, 5, 1, 2, 3, 0) << std::endl;//используем фунцкцию

std::cout << mn(2, 34, 4445);//используем функцию

}

int mn(int n, ...)

{

int\* p = &n;

int min = \*(++p);// в качестве наименьшего сначала выбираем первое значение

for (int i = 1; i <= n; i++) {// используя цикл находим меньшее значение

p++;

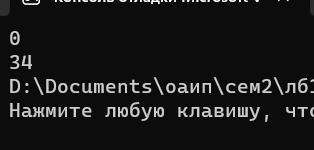
if (min > \*p)

min = \*p;

}

return min;// возвращаем результат

}



Вариант 2



#include <iostream>

#include <math.h>

double function2(double x) {

return pow(cos(x),3);//сама функция

}

double function1(double x) {

return (x \* x\*x + x - 3);//сама функция

}

double counter(double a, double b, double e, double(\*f1)(double)) {

double x;

while (fabs(a - b) > 2 \* e)//используем цикл while.Если условие в скобках истинно, то выполняются операторы цикла до тех пор, пока оно не станет ложным.

{

x = (a + b) / 2;

if (f1(x) \* f1(a) <= 0)//елси уловие с вобках верно то b=x

b = x;

else // иначе a=x

a = x;

}

return x;

}

int main()

{

setlocale(LC\_CTYPE, "Russian");

double e = 0.0001, x, a, b;

std::cout << "Введите ограничение ";

std::cin >> a;

std::cout << "Введите ограничение ";

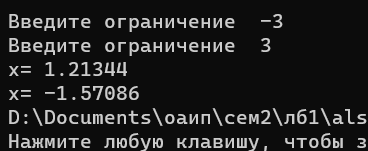
std::cin >> b;

std::cout << "x= " << counter(a, b, e, function1) << std::endl;

std::cout << "x= " << counter(a, b, e, function2);

return 0;

}





#include <iostream>

int fsum(int n, ...);

int main()

{

std::cout << fsum(6, 4, 5, 1, 2, 3, 0) << std::endl;//используем фунцкцию

std::cout << fsum(2, 34, 4445);//используем функцию

}

int fsum(int n, ...)

{

int\* p = &n;

int sum = 0;

p++;

int k = \*p;

for (int i = 1; i < n; i++) {// используя цикл находим суммы

k=k\* \*(++p);

sum = sum + k;

k = \*p;

}

return sum;// возвращаем результат

}

