Лабораторная работа 1

По дисциплине «Основы алгоритмизации и программирования»

На тему «Указатели на функции»



Macros.h

#include <iostream>

#include <vector>

#include <ctime>

#include <cmath>

#include <iomanip>

#include <stack>

#include <map>

#include <string>

#include <chrono>

#include <Windows.h>

#include <cstdarg>

#define vi cout <<

#define vv cin >>

#define nl cout << "\n";

#define nw cout << "\t";

#define SCOCP1251 SetConsoleOutputCP(1251);

#define SCCP1251 SetConsoleCP(1251);

#define ret return

#define pause system("pause");

using namespace :: std;

.cpp

#include "Macros.h"

// уравнение x^3 + 2x - 1

double equation1(double x) {

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

}

// уравнение 2: e^x - 2

double equation2(double x) {

return exp(x) - 2;

}

// дихотомия, показатель на функцию, промежуток и значение e

double dichotomyMethod(double (\*equation)(double), double a, double b, double eps) {

double fa, fb, fx, x;

while (fabs(b - a) > eps) {

// середина интервала

x = (a + b) / 2;

// значение уравнения в a, b и x

fa = equation(a);

fb = equation(b);

fx = equation(x);

// Если знаки функции в A и X разные корень между ними

if (fa \* fx < 0) {

b = x; // b = x

}

else {

a = x; // если между b і x, a = x

}

}

return x;

}

int main() {

double a, b, e = 0.001;

SCOCP1251;

SCCP1251

// интервал

printf("Введите интервал [a, b]: ");

// Функция scanf\_s для возврата количества считанных

if (scanf\_s("%lf %lf", &a, &b) != 2) { // если не 2, то не был введен интервал => возврат ошибки.

printf("Ошибка ввода.\n");

return 1;

}

// корни дихотомией

double root\_1 = dichotomyMethod(equation1, a, b, e);

double root\_2 = dichotomyMethod(equation2, a, b, e);

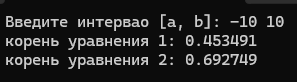
// вывод

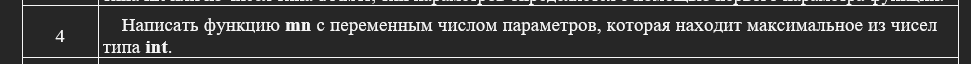
printf("корень уравнения 1: %lf\n", root\_1);

printf("корень уравнения 2: %lf\n", root\_2);

ret 0;

}



#2

.cpp

#include "Macros.h"

int mn(int count, ...) { // количество

va\_list args; // список аргументов

va\_start(args, count); // начинаем с первого аргумента после count

int maxNum = va\_arg(args, int); // первый аргумент = максимальный

for (int i = 1; i < count; i++) { // итераций меньше на 1 т.к. у нас уже проверено первое

int num = va\_arg(args, int); // получаем следующий аргумент

if (num > maxNum) { // если текущий аргумент больше максимального

maxNum = num; // обновляем максимум

}

}

va\_end(args); // завершаем работу со списком аргументов

return maxNum;

}

int main() {

vi "Max:"; nw vi mn(5, 24, 214, 94, -111, 75); nl

vi "Max:"; nw vi mn(3, 134, -5524, 534524); nl

vi "Max:"; nw vi mn(8, 10542, -54575, 8752, -2752, 1787255, -7572, 37527, 0); nl ret 0;

}

