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

Литвинчук Дарья Валерьевна

Преподаватель: асс. Андронова М.В.

2023, Минск

**Вариант 10**

**Задание 1**

|  |  |
| --- | --- |
| 10 | sin(x) + x3,  0,4 + x3 |

#include <iostream>

using namespace std;

double firstEqua(double x) {

return (sin(x) + pow(x, 3));//возвращаем значение первого уравнения

}

double secondEqua(double x) {

return (0.4 + pow(x, 3));//возвращаем значение второго уравнения

}

double dixMethod(double (\*Equa)(double), double a, double b, double e) {//функция отвечающая за метод Дихотомии

if (Equa(a) \* Equa(b) > 0) {//проверяем применим ли метод

cout << "метод дихотомии нельзя применить" << endl;

return -1;

}

else {

while ((b - a) / 2.0 > e) {//цикл для метода дихотомии

double mid = (a + b) / 2.0;//вычисляем середину интервала

if (Equa(mid) == 0) {

return mid;//возвращаем корень, если от был в середине

}

else if (Equa(a) \* Equa(mid) < 0) {

b = mid;//сдвиг правой границы интервала к середине

}

else {

a = mid;//сдвиг левой границы интервала к середине

}

}

return (a + b) / 2.0;

}

}

int main() {

setlocale(LC\_ALL, "rus");

double a, b, e = 0.001;

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

cin >> a >> b;

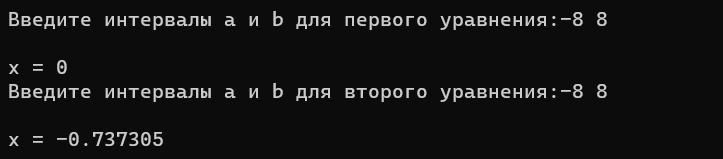
cout << endl << "x = " << dixMethod(firstEqua, a, b, e) << endl;//вызывают функцию для реализации метода Дихотомии

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

cin >> a >> b;

cout << endl << "x = " << dixMethod(secondEqua, a, b, e) << endl;//вызывают функцию для реализации метода Дихотомии

}



**Задание 2**

|  |  |
| --- | --- |
| 10 | Написать функцию, которая находит в строке слово с самой первой по алфавиту буквы. С ее помощью реализовать размещение слов в выходной строке в алфавитном порядке. |

#include <iostream>

#include <string>

#include <cstdarg> // библиотека для работы с переменным количеством аргументов

using namespace std;

//Функция для сортировки слов в алфавитном порядке

string sortWordsAlphabetically(int numWords, ...) {

va\_list args;// Создаем список аргументов

va\_start(args, numWords); // Инициализируем список аргументов

string\* words = new string[numWords]; // Создаем массив строк для слов Сдинамический массив)

// Заполняем массив строк словами из списка аргументов

for (int i = 0; i < numWords; ++i) {

words[i] = va\_arg(args, const char\*); // Получаем следующий аргумент из списка и добавляем его в массив строк

}

va\_end(args); // Завершаем работу с аргументами

// Сортируем массив строк в алфавитном порядке методом пузырька

for (int i = 0; i < numWords - 1; ++i) {

for (int j = i + 1; j < numWords; ++j) {

if (words[i] > words[j]) {// Если текущее слово больше следующего, меняем их местами

string temp = words[i];

words[i] = words[j];

words[j] = temp;

}

}

}

// Собираем отсортированные слова в строку

string sortedString;

for (int i = 0; i < numWords; ++i) {

sortedString += words[i] + " "; // Добавляем текущее слово к строке с пробелом

}

delete[] words; // Освобождаем память, выделенную под.массив строк

return sortedString; // Возвращаем отсортированную строку

}

int main() {

setlocale(LC\_ALL, "RU");

cout << "Первая строка (отсортированная в алфавитном порядке):";

cout << sortWordsAlphabetically(5, "оаип", "опи", "кяр", "кпо", "ксис") << endl;

cout << "Вторая строка (отсортированная в алфавитном порядке):";

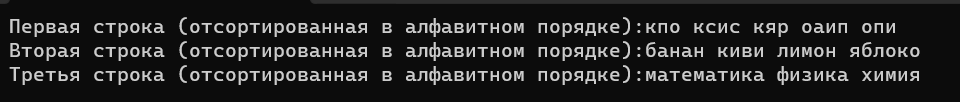
cout << sortWordsAlphabetically(4, "яблоко", "банан", "киви", "лимон") << endl;

cout << "Третья строка (отсортированная в алфавитном порядке):";

cout << sortWordsAlphabetically(3, "физика", "химия", "математика") << endl;

return 0;

}



**Доп. Задачи:**

**Задание 1**

**Вариант 1**

|  |  |
| --- | --- |
| 1 | x2 + 4x – 2,  sin(x) + 0,1 |

#include <iostream>

using namespace std;

double firstEqua(double x) {

return (x \* x + 4 \* x - 2);//возвращаем значение первого уравнения

}

double secondEqua(double x) {

return (sin(x) + 0.1);//возвращаем значение второго уравнения

}

double dixMethod(double (\*Equa)(double), double a, double b, double e) {//функция отвечающая за метод Дихотомии

if (Equa(a) \* Equa(b) > 0) {//проверяем применим ли метод

cout << "метод дихотомии нельзя применить" << endl;

return -1;

}

else {

while ((b - a) / 2.0 > e) {//цикл для метода дихотомии

double mid = (a + b) / 2.0;//вычисляем середину интервала

if (Equa(mid) == 0) {

return mid;//возвращаем корень, если от был в середине

}

else if (Equa(a) \* Equa(mid) < 0) {

b = mid;//сдвиг правой границы интервала к середине

}

else {

a = mid;//сдвиг левой границы интервала к середине

}

}

return (a + b) / 2.0;

}

}

int main() {

setlocale(LC\_ALL, "rus");

double a, b, e = 0.001;

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

cin >> a >> b;

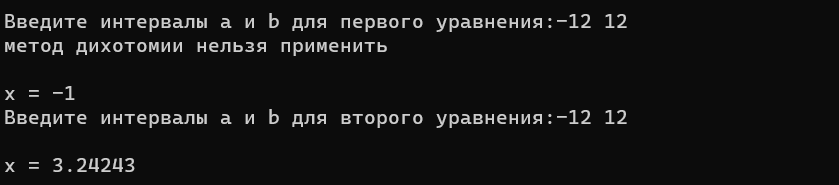
cout << endl << "x = " << dixMethod(firstEqua, a, b, e) << endl;//вызывают функцию для реализации метода Дихотомии

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

cin >> a >> b;

cout << endl << "x = " << dixMethod(secondEqua, a, b, e) << endl;//вызывают функцию для реализации метода Дихотомии

}

****

**Вариант 2**

|  |  |
| --- | --- |
| 2 | x3 + x – 3,  cos3(x) |

#include <iostream>

using namespace std;

double firstEqua(double x) {

return (x \* x \* x + x - 3);//возвращаем значение первого уравнения

}

double secondEqua(double x) {

return (cos(x) \* cos(x) \* cos(x));//возвращаем значение второго уравнения

}

double dixMethod(double (\*Equa)(double), double a, double b, double e) {//функция отвечающая за метод Дихотомии

if (Equa(a) \* Equa(b) > 0) {//проверяем применим ли метод

cout << "метод дихотомии нельзя применить" << endl;

return -1;

}

else {

while ((b - a) / 2.0 > e) {//цикл для метода дихотомии

double mid = (a + b) / 2.0;//вычисляем середину интервала

if (Equa(mid) == 0) {

return mid;//возвращаем корень, если от был в середине

}

else if (Equa(a) \* Equa(mid) < 0) {

b = mid;//сдвиг правой границы интервала к середине

}

else {

a = mid;//сдвиг левой границы интервала к середине

}

}

return (a + b) / 2.0;

}

}

int main() {

setlocale(LC\_ALL, "rus");

double a, b, e = 0.001;

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

cin >> a >> b;

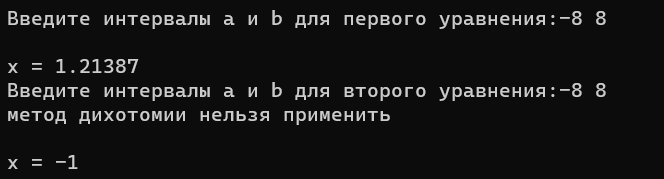
cout << endl << "x = " << dixMethod(firstEqua, a, b, e) << endl;//вызывают функцию для реализации метода Дихотомии

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

cin >> a >> b;

cout << endl << "x = " << dixMethod(secondEqua, a, b, e) << endl;//вызывают функцию для реализации метода Дихотомии

}



**Вариант 4**

|  |  |
| --- | --- |
| 4 | x3 + 2x – 1,  ex − 2 |

#include <iostream>

using namespace std;

double firstEqua(double x) {

return (x \* x \* x + 2 \* x - 1);//возвращаем значение первого уравнения

}

double secondEqua(double x) {

return (exp(x) - 2);//возвращаем значение второго уравнения

}

double dixMethod(double (\*Equa)(double), double a, double b, double e) {//функция отвечающая за метод Дихотомии

if (Equa(a) \* Equa(b) > 0) {//проверяем применим ли метод

cout << "метод дихотомии нельзя применить" << endl;

return -1;

}

else {

while ((b - a) / 2.0 > e) {//цикл для метода дихотомии

double mid = (a + b) / 2.0;//вычисляем середину интервала

if (Equa(mid) == 0) {

return mid;//возвращаем корень, если от был в середине

}

else if (Equa(a) \* Equa(mid) < 0) {

b = mid;//сдвиг правой границы интервала к середине

}

else {

a = mid;//сдвиг левой границы интервала к середине

}

}

return (a + b) / 2.0;

}

}

int main() {

setlocale(LC\_ALL, "rus");

double a, b, e = 0.001;

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

cin >> a >> b;

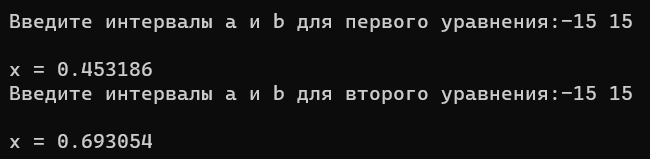
cout << endl << "x = " << dixMethod(firstEqua, a, b, e) << endl;//вызывают функцию для реализации метода Дихотомии

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

cin >> a >> b;

cout << endl << "x = " << dixMethod(secondEqua, a, b, e) << endl;//вызывают функцию для реализации метода Дихотомии

}

****

**Задание 2**

**Вариант 1**

|  |  |
| --- | --- |
| 1 | Написать функцию **fmin** с переменным числом параметров, в которой определяется минимальное из чисел типа **int**. |

#include <iostream>

#include <cstdarg>

using namespace std;

void fmin(int n, ...)

{

va\_list arg;

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);

}

int main()

{

setlocale(LC\_ALL, "rus");

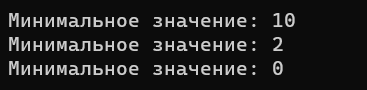
//вызываем функции с переменным числом параметров

fmin(5, 10, 15, 20, 25, 30);

fmin(4, 5, 3, 9, 2);

fmin(6, 0, 2, 18, 21, 4, 30);

}



**Вариант 8**

|  |  |
| --- | --- |
| 8 | Написать функцию **mn** с переменным числом параметров, которая находит минимальное из чисел типа **int**. |

#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");

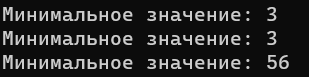
//вызываем функции с переменным числом параметров

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

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

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

}



**Вариант 4**

|  |  |
| --- | --- |
| 4 | Написать функцию **mn** с переменным числом параметров, которая находит максимальное из чисел типа **int**. |

#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);

}

int main()

{

setlocale(LC\_ALL, "rus");

// Вызываем функции с переменным числом параметров

mn(3, 15, 56, 23);

mn(4, 116, 34, 35, 23);

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

}

