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:

Кулешов Артём

Студент 1 курса 8 группы

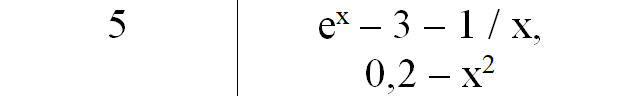
Преподаватель: асс. Андронова М.В.

Минск, 2024

4. 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>

using namespace std;

int GetRoot1() //функция для вычисления корня первого уравнения//

{

setlocale(LC\_CTYPE, "rus");

float a = -100, b = 100, e = 0.0001, x;

cout << "Введите a: ";

cin >> a;

cout << "Введите b: ";

cin >> b;

//вычисления для метода дихотоми//

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

{

x = (a + b) / 2;

if (((exp(x) - 3 - 1) / x) \* ((exp(a) - 3 - 1) / a) <= 0) //подстановка уравнения//

{

b = x;

}

else a = x;

}

cout << x;

return 0;

}

int GetRoot2() //функция для вычисления корня второго уравнения//

{

setlocale(LC\_CTYPE, "rus");

float a = -100, b = 100, e = 0.0001, x;

cout << "Введите a: ";

cin >> a;

cout << "Введите b: ";

cin >> b;

//вычисления для метода дихотоми//

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

{

x = (a + b) / 2;

if ((0.2 \* (x \* x)) \* (0.2 \* (a \* a)) <= 0) //подстановка уравнения//

{

b = x;

}

else a = x;

}

cout << x;

return 0;

}

//функция указателя//

void ShowResults(int (\*ff)())

{

cout << ff() << endl;

}

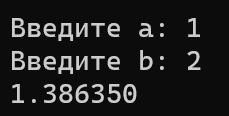
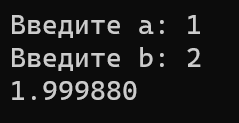
//использование функции указателя для выбора необходимого уравнения//

void main()

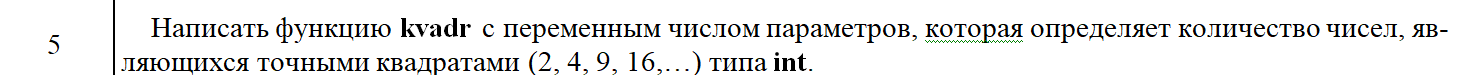
{

ShowResults(GetRoot2);

}

5. 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>

using namespace std;

int kvadr(int count, ...) { //функция для определения количества точных квадратов среди переданных чисел//

va\_list args; //создание списка аргументов//

va\_start(args, count); //инициализация va\_list, устанавливая начальное положение для доступа к аргументам//

int squareCount = 0;

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

int num = va\_arg(args, int); //получение следующего аргумента из va\_list//

int sqrtNum = sqrt(num); //проверка, является ли число точным квадратом//

if (sqrtNum \* sqrtNum == num) {

squareCount++;

}

//конец проверки//

}

va\_end(args); //завершение работы со списком аргументов//

return squareCount;

}

int main() {

setlocale(LC\_CTYPE, "rus");

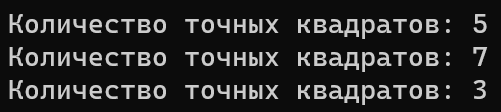
cout << "Количество точных квадратов: " << kvadr(11, 5, 9, 18, 81, 25, 66, 32, 4, 16, 69, 132) << endl; //первый вызов функции kvadr//

cout << "Количество точных квадратов: " << kvadr(12, 9, 2, 6, 25, 36, 256, 121, 72, 64, 289, 3, 17) << endl; //второй вызов функции kvadr//

cout << "Количество точных квадратов: " << kvadr(5, 88, 32, 16, 49, 225) << endl; //третий вызов функции kvadr//

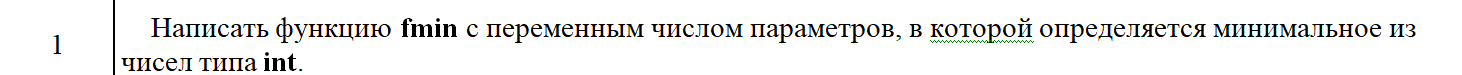
return 0;

}



Дополнительные задачи

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>

using namespace std;

void fmin(int count, ...)

{

va\_list arg; //создание списка аргументов переменной длинны//

va\_start(arg, count); //инициализация va\_list, устанавливая начальное положение для доступа к аргументам//

int min = va\_arg(arg, int); //получение первого аргумента и обозначение предположительного минимума//

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

int num = va\_arg(arg, int); //получение следующего аргумента//

if (min > num) {

min = num; //если текущий аргумент меньше минимума - минимум обновляется//

}

}

va\_end(arg); //завершение работы со списком аргументов//

cout << min << endl;

}

int main() {

setlocale(LC\_CTYPE, "rus");

cout << "Минимальное число из ряда: ";

fmin(5, 12, 21, 5, 9, 7); //первый вызов функции minNum//

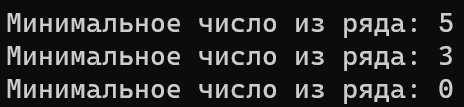
cout << "Минимальное число из ряда: ";

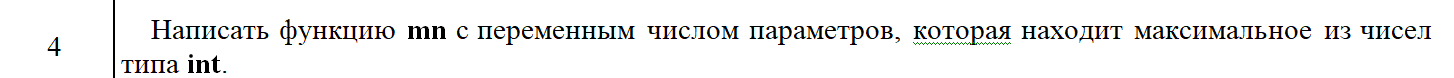
fmin(6, 8, 9, 3, 7, 54, 60); //второй вызов функции minNum//

cout << "Минимальное число из ряда: ";

fmin(6, 8, 0, 12, 54, 1); //третий вызов функции minNum//

}





#include<iostream>

#include <cstdarg>

using namespace std;

void mn(int count, ...)

{

va\_list arg; //создание списка аргументов переменной длинны//

va\_start(arg, count); //инициализация va\_list, устанавливая начальное положение для доступа к аргументам//

int max = va\_arg(arg, int); //получение первого аргумента и обозначение предположительного минимума//

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

int num = va\_arg(arg, int); //получение следующего аргумента//

if (max < num) {

max = num; //если текущий аргумент больше максимума - минимум обновляется//

}

}

va\_end(arg); //завершение работы со списком аргументов//

cout << max << endl;

}

int main()

{

setlocale(LC\_CTYPE, "rus");

cout << "Максимальное число из ряда: ";

mn(5, 12, 13, 5, 9, 7); //первый вызов функции minNum//

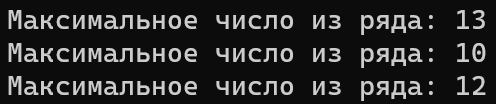
cout << "Максимальное число из ряда: ";

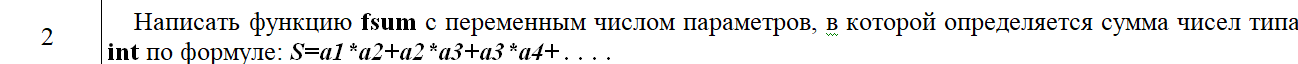
mn(6, 8, 9, 3, 7, 10, 2); //второй вызов функции minNum//

cout << "Максимальное число из ряда: ";

mn(6, 8, 0, 12, 2, 1, 7); //третий вызов функции minNum//

}





#include <iostream>

#include <cstdarg>

using namespace std;

int fsum(int count, ...)

{

int result = 0;

va\_list args; //создание списка аргументов переменной длинны//

va\_start(args, count); //создание списка аргументов//

int current = va\_arg(args, int);

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

int next = va\_arg(args, int);

result += current \* next;

current = next;

}

va\_end(args);

return result;

}

int main() {

setlocale(LC\_CTYPE, "rus");

int sum1 = fsum(4, 22, 521, 327, 0); //первый вызов функции fsum//

cout << "Сумма по формуле: " << sum1 << endl;

int sum2 = fsum(5, 21, 84, 96, 7, 6); //второй вызов функции fsum//

cout << "Сумма по формуле: " << sum2 << endl;

int sum3 = fsum(7, 78, 1, 36, 25, 4, 61, 3); //третий вызов функции fsum//

cout << "Сумма по формуле: " << sum3 << endl;

return 0;

}

