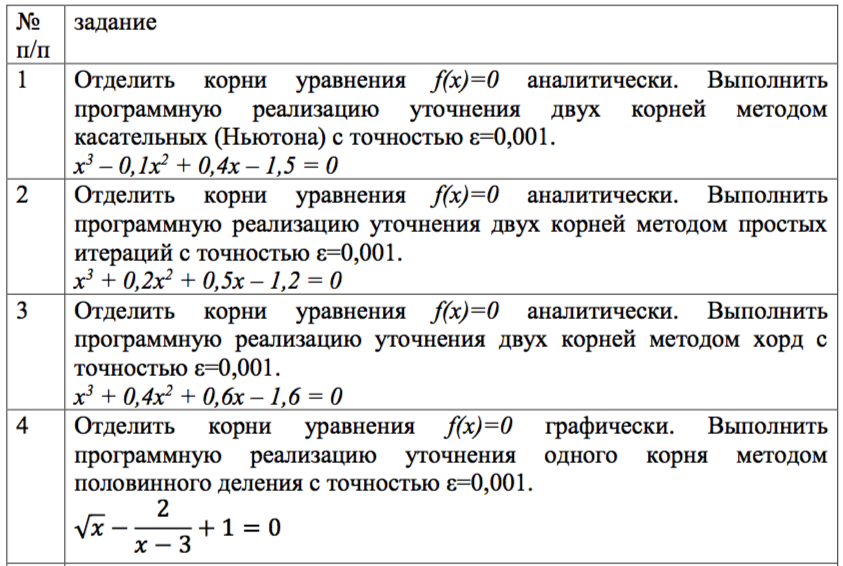
Условия:

Исходный код программы:  
  
#include **"stdafx.h"**#include **<iostream>**#include **<math.h>**#include **<windows.h>  
using namespace** std;  
*//Функция, определяющая левую часть уравнения f (x) = 0.***double** f(**double** x)  
{  
 **return** (x\*x\*x - 0.1\*x\*x + 0.4\*x - 1.5);  
}  
**double** fchord(**double** x)  
{  
 **return** (x\*x\*x + 0.4\*x\*x + 0.6\*x - 1.6);  
}  
**double** FITER(**double** x)  
{  
 **return** (x\*x\*x + 0.2\*x\*x + 0.5\*x - 1.2);  
}  
**double** fDichotomy(**double** x)  
{  
 **return** (sqrt(x) - 2 / (x - 3) + 1);  
}  
  
*//Функция, реализующая метод половинного деления.***int** Dichotomy(**double** a, **double** b, **double** \* c, **double** eps)  
{  
 **int** k = 0;  
 **do** {  
 \*c = (a + b) / 2;  
 **if** (fDichotomy(\*c) \* fDichotomy(a) <0) b = \*c;  
 **else** a = \*c;  
 k++;  
 } **while** (fabs(a - b) >= eps);  
 **return** k;  
}  
*//Функция, реализующая метод хорд.***int** Chord(**double** a, **double** b, **double** \* c, **double** eps)  
{  
 **int** k = 0;  
 **do** {  
 \*c = a - fchord(a) / (fchord(b) - fchord(a)) \* (b - a);  
 **if** (fchord(\*c) \* fchord(a) >0) a = \*c;  
 **else** b = \*c;  
 k++;  
 } **while** (fabs(fchord(\*c)) >= eps);  
 **return** k;  
}  
**double** f1(**double** x) *//Первая производная функции f (x).*{  
 **return** (3 \* x\*x - 0.2\*x + 0.4);  
}  
**double** f2(**double** x) *//Вторая производная функции f (x).*{  
 **return** (6 \* x - 0.2);  
}  
*//Функция, реализующая метод касательных.***int** Tangent(**double** a, **double** b, **double** \* c, **double** eps)  
{  
 **int** k = 0;  
 **if** (f(a) \* f2(a) >0) \* c = a;  
 **else** \* c = b;  
 **do** {  
 \*c = \*c - f(\*c) / f1(\*c);  
 k++;  
 } **while** (fabs(f(\*c)) >= eps);  
 **return** k;  
}  
  
**double** fi(**double** x, **double** L)  
{  
 **return** (x + L\* FITER(x));  
}  
*//Функция, реализующая метод простой итерации.***int** Iteration(**double** \*x, **double** L, **double** eps)  
{  
 **int** k = 0; **double** x0;  
 **do** {  
 x0 = \*x;  
 \*x = fi(x0, L);  
 k++;  
 } **while** (fabs(x0 - \*x) >= eps);  
 **return** k;  
}  
  
**int** main()  
{  
 SetConsoleCP(1251);  
 SetConsoleOutputCP(1251);  
 **double** A, B, X, P;  
 **double** ep = 0.001; *//Точность вычислений.* **int** K;  
  
 cout << **"Решение уравнения x\*x\*x - 0.1\*x\*x+0.4\*x-1.5 =0. "** << endl;  
 cout << **"Метод касательных:"** << endl;  
 K = Tangent(A, B, &X, ep);  
 cout << **" Найденное решение x="** << X;  
 cout << **", количество итераций k="** << K << endl;  
  
 cout << **"Решение уравнения x\*x\*x + 0.4\*x\*x + 0.6\*x - 1.6 =0. "** << endl;  
 cout << **"Метод хорд:"** << endl;  
 K = Chord(A, B, &X, ep);  
 cout << **" Найденное решение x="** << X;  
 cout << **", количество итераций k="** << K << endl;  
  
 cout << **"Решение уравнения sqrt(x) - 2/(x-3) + 1 =0. "** << endl;  
 cout << **"Метод дихотомии:"** << endl;  
 cout << **"a="**; cin >> A; *//Интервал изоляции корня.* cout << **"b="**; cin >> B;  
 K = Dichotomy(A, B, &X, ep);  
 cout << **"Найденное решение x="** << X;  
 cout << **", количество итераций k="** << K << endl;  
  
 cout << **"Решение уравнения x\*x\*x + 0.2\*x\*x + 0.5\*x - 1.2 =0. "** << endl;  
 cout << **"Метод простой итерации:"** << endl;  
 cout << **"a="**; cin >> A; *//Интервал изоляции корня.* cout << **"b="**; cin >> B;  
 X = A;  
 cout << **"L="**; cin >> P;*//подбираем число максимально приближённое к нулю!!* K = Iteration(&X, P, ep);  
 cout << **" Найденное решение x="** << X;  
 cout << **", количество итераций k="** << K << endl;  
 system(**"pause"**);  
 **return** 0;  
}

Скриншоты программы:

