1 задание Свенн и метод деления пополам

using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;  
  
namespace Optimization\_methods\_test2  
{  
 class Program  
 {  
 static void Main(string[] args)  
 {  
 double a = 0, b = 0, delta = 0, xp, k = 0, y, e = 0.0001d;  
 double xs, yk, zk, lk;  
 double exs;  
 int p = 0;  
   
 Console.Write("x = ");  
 double x = Convert.ToDouble(Console.ReadLine());  
 Console.Write("t = ");  
 double t = Convert.ToDouble(Console.ReadLine());  
   
 //part 1  
 if ((F(x - t) <= F(x)) && (F(x + t) <= F(x)))   
 {  
 Console.WriteLine("unimodal function");  
 }  
  
 if ((F(x - t) >= F(x) && (F(x + t) >= F(x))))   
 {  
 a = x - t;  
 b = x + t;  
 p = 1;  
 }  
  
 xp = x;  
  
 if ((F(x - t) >= F(x)) && (F(x) >= F(x + t)))   
 {  
 delta = t;  
 a = x;  
 x = x + t;  
 }  
 if ((F(x - t) <= F(x)) && (F(x) <= F(x + t)))  
 {  
 delta = -t;  
 b = x;  
 x = x - t;  
 }  
  
 while (p != 1)   
 {  
 if ((F(x) < F(xp)) && (delta \* t > 0))   
 {  
 a = xp;  
 }  
 if ((F(x) < F(xp)) && (delta \* t < 0))  
 {  
 b = xp;  
 }  
 if ((F(x) > F(xp)) && (delta \* t > 0))   
 {  
 b = x;  
 p = 1;  
 }  
 if ((F(x) > F(xp)) && (delta \* t < 0))  
 {  
 a = x;  
 p = 1;  
 }  
  
 k++;  
 xp = x;  
 x = xp + Math.Pow(2, k) \* delta;  
 if (k>1000)  
 {  
 Console.WriteLine("invalid number of steps");  
 Console.ReadKey();  
 break;  
 }  
 }  
 Console.WriteLine("[{0},{1}]; k = {2}", a, b, k);  
 // part 2   
 lk = b - a;  
 k = 0;  
 while (lk > e)   
 {  
 xs = (a + b) / 2;  
 lk = b - a;  
 yk = a + (lk / 4);  
 zk = b - (lk / 4);  
  
 if (F(yk) < F(xs))   
 {  
 b = xs;  
 }  
 if (F(yk) >= F(xs))   
 {  
 if (F(zk) < F(xs))   
 {  
 a = xs;  
 }  
 if (F(zk) >= F(xs))  
 {  
 a = yk;  
 b = zk;  
 }  
 }  
 k++;  
 }  
  
 exs = (a + b) / 2;  
 Console.WriteLine("ext = {0}; f(ext) = {1}; k = {2}", exs, F(exs), k);  
 Console.ReadKey();  
  
 }  
  
 public static double F( double x)  
 {  
 double y;  
 y = Math.Pow((x - 5), 2);  
 //y = 2 \* Math.Pow(x, 2) - 12 \* x;  
 //y = Math.Pow((Math.Pow(x, 2) \* (x + 3)), (1.0 / 3));  
 //y = Math.Pow((Math.Pow(x, 3) + 3 \* Math.Pow(x, 2)), 1.0 / 3);  
 return y;  
 }  
 }  
}

2 задание Метод золотого сечения

using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;  
  
namespace петрович  
{  
 class Program  
 {  
 static void Main(string[] args)  
 {  
  
   
 double exs, a=0, b=0, delta=0, xp, k = 0, y, i;  
 double e = 0.001, Lk;  
   
 int p = 0;  
 //Console.Write("Введите а = ");//cout  
 //double a = Convert.ToDouble(Console.ReadLine());//cin  
  
 Console.WriteLine("Введите х");  
 double x = Convert.ToDouble(Console.ReadLine());  
 Console.WriteLine("Введите t=");  
 double t = Convert.ToDouble(Console.ReadLine());  
  
  
 if ((F(x - t) <= F(x)) && (F(x + t) <= F(x)))  
 {  
 //p = 1;  
 Console.WriteLine("Функция не являтся унимодальной, введите другую");  
 Console.ReadKey();  
 };  
  
 if ((F(x - t) >= F(x)) && (F(x + t) >= F(x)))  
 {  
 a = x - t;  
 b = x + t;  
 p = 1;  
 };  
  
  
 xp = x;  
  
 if ((F(x - t) >= F(x)) && (F(x) >= F(x + t)))  
  
 {  
 delta = t;  
 a = x;  
 x = x + t;  
 }  
  
 if ((F(x - t) <= F(x)) && (F(x) <= F(x + t)))  
 {  
 delta = -t;  
 b = x;  
 x = x - t;  
 }  
  
 while ((p != 1))  
 {  
  
 if ((F(x) < F(xp)) && (delta \* t > 0))  
 a = xp;  
 if ((F(x) < F(xp)) && (delta \* t < 0))  
 b = xp;  
 if ((F(x) > F(xp)) && (delta \* t > 0))  
 {  
 b = x;  
 p = 1;  
 };  
  
 if ((F(x) > F(xp)) && (delta \* t < 0))  
 {  
 a = x;  
 p = 1;  
 };  
  
 k++;  
 xp = x;  
 x = xp + Math.Pow(2.0, k) \* delta;  
 if (k > 1000)  
 {  
 Console.WriteLine("Количество шагов превышает допустимое значение");  
 Console.ReadKey();  
 };  
  
 }  
 Console.WriteLine("Интервал: [ a={0}; b={1}]", a, b);  
 Console.WriteLine("колличесво шагов по алгоритму свена: k={0}", k);  
  
  
 double z, yp, zp;  
  
 delta = b - a;  
 Lk = b - a;  
 k = 0;  
 y = a + ((3 - Math.Sqrt(5)) / 2) \* Lk;  
 //y = a + 0.38196601\*Lk;  
 z = a + b - y;  
 while (delta > e)  
 {  
  
 if (F(y) <= F(z))  
 {  
 b = z;  
 yp = y;  
 y = a + b - yp;  
 z = yp;  
 }  
 else  
 {  
 a = y;  
 y = z;  
 zp = z;  
 z = a + b - zp;  
 };  
 //Lk = b - a;  
 delta = b - a;  
 k++;  
 if (k > 1000)  
 {  
 Console.WriteLine("Количество шагов превышает допустимое значение");  
 Console.ReadKey();  
 };  
  
 };  
 exs = (a + b) / 2;  
 i = k - 1;  
 Console.WriteLine("Точка минимума: exs={0:0.##}; Значение функции в точке минимума F(exs)={1:0.##} ", exs, F(exs));  
 Console.WriteLine("Колличество шагов золотого сечения: i={0}", i);  
 Console.ReadKey();  
  
 }  
  
 public static double F(double x)  
  
 {  
  
  
 //y = x;  
 //y = 100\*pow((100-x\*x),2)+pow((10-x),2);  
 double y = Math.Pow((x \* x \* (x + 3)), 0.333333333);  
 //double y = x\*x + 2 \* x - 5;  
 //y = (x - 5)\*(x - 5);  
 //y = x\*x;  
 //y = 2 \* x\*x - 12 \* x;  
 //y = 2 \* x\*x + 16 / x;  
 return (y);  
  
 }  
 }  
}

3 задание Метод квадратичной интерполяции

#include <iostream.h>

#include <conio.h>

#include <math.h>

#include <iomanip.h>

#include "stdafx.h"

#include <stdio.h>

#include <iostream>

using namespace std;

double f(double);

int \_tmain(int argc, \_TCHAR\* argv[])

{

setlocale(LC\_CTYPE, "Russian");

start:

double t, a, b, x, delta, xp, k = 0, y;

double e = 0.001, Lk;

float exs;

int p = 0;

//cout << "ââåäèòå e=";

//cin >> e;

cout << "Ââåäèòå x=";

cin >> x;

cout << "Ââåäèòå t=";

cin >> t;

if ((f(x - t) <= f(x)) && (f(x + t) <= f(x)))

{

//p = 1;

cout << "Ôóíêöèÿ íå ÿâëÿåòñÿ óíèìîäàëüíîé, ââåäèòå äðóãîå çíà÷åíèå õ\n";

goto start;

};

if ((f(x - t) >= f(x)) && (f(x + t) >= f(x)))

{

a = x - t;

b = x + t;

p = 1;

};

xp = x;

if ((f(x - t) >= f(x)) && (f(x) >= f(x + t)))

{

delta = t;

a = x;

x = x + t;

}

if ((f(x - t) <= f(x)) && (f(x) <= f(x + t)))

{

delta = -t;

b = x;

x = x - t;

}

while ((p != 1))

{

if ((f(x)< f(xp)) && (delta\*t >0))

a = xp;

if ((f(x)< f(xp)) && (delta\*t <0))

b = xp;

if ((f(x)> f(xp)) && (delta\*t >0))

{

b = x;

p = 1;

};

if ((f(x)> f(xp)) && (delta\*t<0))

{

a = x;

p = 1;

};

k++;

xp = x;

x = xp + pow(2.0, k)\*delta;

if (k>1000)

{

cout << "Êîëè÷åñòâî øàãîâ ïðåâûøàåò äîïóñòèìîå çíà÷åíèå( Óõîäèò â ìèíóñ áåñêîíå÷íîñòü)\n";

system("pause");

return 0;

};

}

cout << "Èíòåðâàë: [" << a << ";" << b << "]" << endl;

cout << "Êîëè÷åñòâî øàãîâ ïî àëãîðèòìó Ñâåííà: " << k << endl;

double c, Xmin, Fmin, X\_, e1 = 0.001, e2 = 0.001, V;

cout << "ââåäèòå e1=";

cin >> e1;

cout << "ââåäèòå e2=";

cin >> e2;

c = (a + b) / 2;

start1:

Fmin = f(a);

Xmin = a;

if (f(b) < Fmin)

{

Fmin = f(b); Xmin = b;

};

if (f(c) < Fmin)

{

Fmin = f(c); Xmin = c;

};

X\_ = (((c\*c - b\*b)\*f(a) + (b\*b - a\*a)\*f(c) + (a\*a - c\*c)\*f(b)) / ((c - b)\*f(a) + (b - a)\*f(c) + (a - c)\*f(b))) / 2;

if (((fabs((Fmin - f(X\_)) / X\_))>e1) && (fabs((Xmin - X\_) / X\_)>e2))

{

if (f(X\_) > f(Xmin))

{

V = Xmin;

if (X\_ > Xmin){ b = X\_; c = Xmin; }

else{ a = X\_; c = Xmin; };

}

else

{

V = X\_;

if (X\_ > Xmin){ a = c; c = X\_; }

else{ b = c; c = X\_; };

};

goto start1;

};

cout << "Òî÷êà õ=" << X\_ << "\nÇíà÷åíèå ôóíêöèè â òî÷êå õ: " << f(X\_) << "\n";

system("Pause");

return 0;

}

double f(double x)

{

double y;

y = cbrt(pow(x, 2.0)\*(x + 3));

return (y);

}

4 задание Метод градиентного спуска с постоянным шагом

using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;  
  
namespace \_4zadanie  
{  
 class Program  
 {  
 public static double F(double x,double y, double z)  
 {  
 double q = ((y - x \* x) \* (y - x \* x) + (1 - x) \* (1 - x) + (z - 2) \* (z - 2));  
 //double q = ((y - x \* x) \* (y - x \* x) + (1 - x) \* (1 - x));  
 return (q);  
 }  
  
 static void Main(string[] args)  
 {  
 double gr1, gr2, gr3, e1=0.001f,e2=0.001f,m=1000,grad=1,h=0.0001f,a,b;  
 int k = 0;  
 Console.Write("Введите x = ");  
 double x = Convert.ToDouble(Console.ReadLine());  
  
 Console.Write("Введите y = ");  
 double y = Convert.ToDouble(Console.ReadLine());  
  
 Console.Write("Введите z = ");  
 double z = Convert.ToDouble(Console.ReadLine());  
 while ((grad>e1)&&(k<m))  
 {  
 //Console.WriteLine("K={0}", k);  
 gr1 = (F(x + h, y, z) - F(x, y, z)) / h;  
 gr2 = (F(x, y + h, z) - F(x, y, z)) / h;  
 gr3 = (F(x, y, z + h) - F(x, y, z)) / h;  
 grad = Math.Sqrt(gr1 \* gr1 + gr2 \* gr2 + gr3 \* gr3);  
 //Console.WriteLine("grad={0}", grad);  
 //метод деления пополам  
 double Lk = 1, e = 0.0001, xs, yk, zk, t, xv, yv, zv;  
 int K;  
  
 a = 0;  
 b = 3.901;  
 Lk = b - a;  
 K = 0;  
 while (Lk > e)  
 {  
 xs = (a + b) / 2;  
 Lk = b - a;  
 yk = a + Lk / 4;  
 zk = b - Lk / 4;  
 if (F(x - yk \* gr1, y - yk \* gr2, z - yk \* gr3) < F(x - xs \* gr1, y - xs \* gr2, z - xs \* gr3))  
 {  
 b = xs;  
 };  
  
 if (F(x - yk \* gr1, y - yk \* gr2, z - yk \* gr3) >= F(x - xs \* gr1, y - xs \* gr2, z - xs \* gr3))  
 {  
 if (F(x - zk \* gr1, y - zk \* gr2, z - zk \* gr3) < F(x - xs \* gr1, y - xs \* gr2, z - xs \* gr3))//6a   
 {  
 a = xs;  
 }  
 if (F(x - zk \* gr1, y - zk \* gr2, z - zk \* gr3) >= F(x - xs \* gr1, y - xs \* gr2, z - xs \* gr3))  
 {  
 a = yk;  
 b = zk;  
 };  
 };  
 K++;  
 };  
 t = (a + b) / 2;  
 //Console.WriteLine("t={0}", t);  
  
 //////////   
  
 xv = x;  
 yv = y;  
 zv = z;  
 x = xv - t \* gr1;  
 y = yv - t \* gr2;  
 z = zv - t \* gr3;  
  
 if ((((Math.Sqrt((xv - x) \* (xv - x) + (yv - y) \* (yv - y) + (zv - z) \* (zv - z))) < e2) && (Math.Abs(F(xv, yv, zv) - F(x, y, z))) < e2) && (grad < e1))  
 {  
 goto step2;  
 };  
  
 k++;  
 };  
 step2:  
 Console.WriteLine();  
 Console.Write("Координаты точки минимума:");  
 Console.WriteLine("\nx = {0}, \ny = {1}, \nz = {2}", x,y,z);  
 Console.WriteLine("\nЗначение функции f(x,y,z) = {0:0.0000}", F(x,y,z));  
 Console.WriteLine("k = {0}", k);  
   
 Console.ReadKey();  
  
  
  
   
 }  
 }  
 }

5 задание Метод Флетчера-Ривса

﻿using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;  
  
namespace \_5\_Градиент\_МО  
{  
 class Program  
 {  
 public static double F(double x, double y, double z)  
 {  
 double q;  
 q = ((y - x \* x) \* (y - x \* x) + (1 - x) \* (1 - x) + (z - 2) \* (z - 2));  
 //q = ((y - x \* x) \* (y - x \* x) + (1 - x) \* (1 - x));  
  
 return (q);  
 }  
 static void Main(string[] args)  
 {  
  
 double gr1, gr2, gr3, gr11 = 1, gr22 = 1, gr33 = 1, e1 = 0.001f, e2 = 0.001f, M = 1000, grad, h = 0.0001f, a, b, bk = 0, bk1 = 0, ost, dk1 = 0, dk2 = 0, dk3 = 0, bdx, bdy, bdz, dkx, dky, dkz;  
  
 int k = 0;  
 Console.Write("Введите x = ");  
 double x = Convert.ToDouble(Console.ReadLine());  
  
 Console.Write("Введите y = ");  
 double y = Convert.ToDouble(Console.ReadLine());  
  
 Console.Write("Введите z = ");  
 double z = Convert.ToDouble(Console.ReadLine());  
 grad = 1;  
  
 while ((grad > e1) && (k < M))  
 {  
 ost = k % 3;  
 gr1 = (F(x + h, y, z) - F(x, y, z)) / h;  
 gr2 = (F(x, y + h, z) - F(x, y, z)) / h;  
 gr3 = (F(x, y, z + h) - F(x, y, z)) / h;  
 grad = Math.Sqrt(gr1 \* gr1 + gr2 \* gr2 + gr3 \* gr3);  
  
 if (ost == 0)  
 {  
 bdx = 0;  
 bdy = 0;  
 bdz = 0;  
 }  
 else  
 {  
 bdx = bk1 \* dk1;  
 bdy = bk1 \* dk2;  
 bdz = bk1 \* dk3;  
 };  
  
  
 bk = (gr1 \* gr1 + gr2 \* gr2 + gr3 \* gr3) / (gr11 \* gr11 + gr22 \* gr22 + gr33 \* gr33);  
 bk1 = bk;  
 dkx = -gr1 + bdx;  
 dk1 = dkx;  
 dky = -gr2 + bdy;  
 dk2 = dky;  
 dkz = -gr3 + bdz;  
 dk3 = dkz;  
  
  
 /////алгоритм нахождения минимума (деление пополам)  
 double Lk = 1, e = 0.0001, xs, yk, zk, t, xv, yv, zv;  
 int K;  
  
 a = 0;  
 b = 3.901;  
 Lk = b - a;  
 K = 0;  
 while (Lk > e)  
 {  
 xs = (a + b) / 2;  
 Lk = b - a;  
 yk = a + Lk / 4;  
 zk = b - Lk / 4;  
 if (F(x + yk \* dkx, y + yk \* dky, z + yk \* dkz) < F(x + xs \* dkx, y + xs \* dky, z + xs \* dkz))  
 {  
 b = xs;  
 };  
  
 if (F(x + yk \* dkx, y + yk \* dky, z + yk \* dkz) >= F(x + xs \* dkx, y + xs \* dky, z + xs \* dkz))  
 {  
 if (F(x + zk \* dkx, y + zk \* dky, z + zk \* dkz) < F(x + xs \* dkx, y + xs \* dky, z + xs \* dkz))//6a  
 {  
 a = xs;  
 }  
 if (F(x + zk \* dkx, y + zk \* dky, z + zk \* dkz) >= F(x + xs \* dkx, y + xs \* dky, z + xs \* dkz))  
 {  
 a = yk;  
 b = zk;  
 };  
 };  
 K++;  
 };  
 t = (a + b) / 2;  
  
  
 //////////  
 gr11 = gr1;  
 gr22 = gr2;  
 gr33 = gr3;  
 xv = x;  
 yv = y;  
 zv = z;  
 x = xv + t \* dkx;  
 y = yv + t \* dky;  
 z = zv + t \* dkz;  
  
 if ((((Math.Sqrt((xv - x) \* (xv - x) + (yv - y) \* (yv - y) + (zv - z) \* (zv - z))) < e2) && (Math.Abs(F(xv, yv, zv) - F(x, y, z))) < e2) && (grad < e1))  
 {  
 goto step2;  
 };  
  
 k++;  
 };  
 step2:  
 Console.WriteLine();  
 Console.Write("Координаты точки минимума:");  
 Console.WriteLine("\nx = {0},\ny = {1},\nz = {2}", x, y, z);  
 Console.WriteLine("\nЗначение функции f(x,y,z) = {0:0.0000}", F(x, y, z));  
 Console.WriteLine("k = {0}", k);  
  
 Console.ReadKey();  
 }  
 }  
}

6 задание Комбинированный метод штрафных функций

﻿using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;  
  
namespace \_6.\_3  
{  
 class Program  
 {  
 public static double F(double x, double y, double z)  
 {  
 double q = x \* x + (y - 2) \* (y - 2);  
 return (q);  
 }  
 public static double G1(double x, double y, double z)  
 {  
 double q = x \* x + 2 \* y \* y - 8;  
 return (q);  
 }  
 public static double G2(double x, double y, double z)  
 {  
 double q = 0;  
 return (q);  
 }  
 public static double G3(double x, double y, double z)  
 {  
 double q = 0;  
 return (q);  
 }  
 public static double H1(double x, double y, double z)  
 {  
 double q = 0;  
 return (q);  
 }  
 public static double H2(double x, double y, double z)  
 {  
 double q = x \* x + 2 \* (y - 2) \* (y - 2) - 8;  
 return (q);  
 }  
 public static double H3(double x, double y, double z)  
 {  
 double q = 0;  
 return (q);  
 }  
 public static double G(double x, double y, double z)  
 {  
 double q = G1(x, y, z) + G2(x, y, z) + G3(x, y, z);  
 return (q);  
 }  
 public static double H(double x, double y, double z)  
 {  
 double q = H1(x, y, z) + H2(x, y, z) + H3(x, y, z);  
 return (q);  
 }  
 static void Main(string[] args)  
 {  
 double grad = 1, gr1, gr2, gr3, a = 0, b = 3.901, Lk = 1, yk, zk, xv, yv, zv, xs, t;  
 double h = 0.0001;  
 double e = 0.001, e1 = 0.01, e2 = 0.0001;  
 double k = 0, M = 1000, n = 0, N = 0, r = 1, p;  
   
  
 Console.Write("Введите x = ");  
 double x = Convert.ToDouble(Console.ReadLine());  
 Console.Write("Введите y = ");  
 double y = Convert.ToDouble(Console.ReadLine());  
 Console.Write("Введите z = ");  
 double z = Convert.ToDouble(Console.ReadLine());  
 Console.WriteLine();  
 do  
 {  
 k = 0;  
 n++;  
 while ((grad > e2) && (k < M))  
 {  
  
 gr1 = ((F(x + h, y, z) + G(x + h, y, z) \* G(x + h, y, z) / (2 \* r) - r / H(x + h, y, z)) - (F(x, y, z) + G(x, y, z) \* G(x, y, z) / (2 \* r) - r / H(x, y, z))) / h;  
 gr2 = ((F(x, y + h, z) + G(x, y + h, z) \* G(x, y + h, z) / (2 \* r) - r / H(x, y + h, z)) - (F(x, y, z) + G(x, y, z) \* G(x, y, z) / (2 \* r) - r / H(x, y, z))) / h;  
 gr3 = ((F(x, y, z + h) + G(x, y, z + h) \* G(x, y, z + h) / (2 \* r) - r / H(x, y, z + h)) - (F(x, y, z) + G(x, y, z) \* G(x, y, z) / (2 \* r) - r / H(x, y, z))) / h;  
 grad = Math.Sqrt(gr1 \* gr1 + gr2 \* gr2 + gr3 \* gr3);  
  
 a = 0;  
 b = 3;  
 xs = (a + b) / 2;  
  
 do  
 {  
 yk = a + Lk / 4;  
 zk = b - Lk / 4;  
 if (F(x - yk \* gr1, y - yk \* gr2, z - yk \* gr3) < F(x - xs \* gr1, y - xs \* gr2, z - xs \* gr3))  
 {  
 b = xs;  
 xs = yk;  
 };  
  
 if (F(x - yk \* gr1, y - yk \* gr2, z - yk \* gr3) >= F(x - xs \* gr1, y - xs \* gr2, z - xs \* gr3))  
 {  
 if (F(x - zk \* gr1, y - zk \* gr2, z - zk \* gr3) < F(x - xs \* gr1, y - xs \* gr2, z - xs \* gr3))  
 {  
 a = xs;  
 xs = zk;  
 }  
 if (F(x - zk \* gr1, y - zk \* gr2, z - zk \* gr3) >= F(x - xs \* gr1, y - xs \* gr2, z - xs \* gr3))  
 {  
 a = yk;  
 b = zk;  
 }  
 Lk = b - a;  
 };  
  
 } while (Lk > e);  
  
 xv = x;  
 yv = y;  
 zv = z;  
 x = x - xs \* gr1;  
 y = y - xs \* gr2;  
 z = z - xs \* gr3;  
  
  
 if (((Math.Sqrt((xv - x) \* (xv - x) + (yv - y) \* (yv - y) + (zv - z) \* (zv - z))) < e2) && ((Math.Abs(F(xv, yv, zv) -F(x, y, z))) < e2))  
 {  
 goto end;  
 };  
  
 k = k + 1;  
 };  
 end:  
 r = r / 4;  
 p = G(x, y, z) \* G(x, y, z) / (2 \* r) - r / H(x, y, z);  
 Console.WriteLine("p = {0}", p);  
  
 } while ((p > e1) && (n< 100));  
  
 Console.WriteLine();  
 Console.WriteLine("n = {0}", n);  
 //Console.WriteLine();  
 Console.WriteLine("\nКоординаты точки минимума:");  
 Console.WriteLine("\nx = {0},\ny = {1},\nz = {2}", x, y, z);  
 Console.WriteLine("\nЗначение функции f(x,y,z) = {0}", F(x, y, z));  
 Console.ReadKey();  
 }  
 }  
}

7 задание Метод множителей

﻿using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;  
  
namespace \_7  
{  
 class Program  
 {  
 public static double FF(double x1, double x2, double x3)  
 {  
 double y;  
 y = x1 \* x1 + (x2 - 2) \* (x2 - 2);  
 return (y);  
 }  
 public static double G1(double x1, double x2, double x3)  
 {  
 double y;  
  
 y = x1 \* x1 + 2 \* x2 \* x2 - 8;  
 return (y);  
 }  
 public static double G2(double x1, double x2, double x3)  
 {  
 double y;  
  
 y = 0;  
 return (y);  
 }  
 public static double G3(double x1, double x2, double x3)  
 {  
 double y;  
  
 y = 0;  
 return (y);  
 }  
 public static double H1(double x1, double x2, double x3)  
 {  
 double y;  
  
 y = x1 \* x1 + 2 \* (x2 - 2) \* (x2 - 2) - 8;  
 return (y);  
 }  
 public static double H2(double x1, double x2, double x3)  
 {  
 double y;  
  
 y = 0;  
 return (y);  
 }  
 public static double H3(double x1, double x2, double x3)  
 {  
 double y;  
  
 y = 0;  
 return (y);  
 }  
 public static double G(double x1, double x2, double x3)  
 {  
 double y;  
  
 y = G1(x1, x2, x3) + G2(x1, x2, x3) + G3(x1, x2, x3);  
 return (y);  
 }  
 public static double H(double x1, double x2, double x3)  
 {  
 double y;  
  
 y = H1(x1, x2, x3) + H2(x1, x2, x3) + H3(x1, x2, x3);  
 return (y);  
 }  
 public static double Gl1(double x1, double x2, double x3, double l)  
 {  
 double y;  
  
 y = G1(x1, x2, x3) \* l;  
 return (y);  
 }  
 public static double Gl2(double x1, double x2, double x3, double l)  
 {  
 double y;  
  
 y = G2(x1, x2, x3) \* l;  
 return (y);  
 }  
 public static double Gl3(double x1, double x2, double x3, double l)  
 {  
 double y;  
  
 y = G3(x1, x2, x3) \* l;  
 return (y);  
 }  
 public static double Gl(double x1, double x2, double x3, double l1, double l2, double l3)  
 {  
 double y;  
  
 y = Gl1(x1, x2, x3, l1) + Gl2(x1, x2, x3, l2) + Gl3(x1, x2, x3, l3);  
 return (y);  
 }  
 public static double Hn1(double max, double nn)  
 {  
 double y;  
  
 y = max \* max - nn \* nn;  
 return (y);  
 }  
 public static double Hn2(double max, double nn)  
 {  
 double y;  
  
 y = max \* max - nn \* nn;  
 return (y);  
 }  
 public static double Hn3(double max, double nn)  
 {  
 double y;  
  
 y = max \* max - nn \* nn;  
 return (y);  
 }  
 public static double Hn(double max, double n1, double n2, double n3)  
 {  
 double y;  
  
 y = Hn1(max, n1) + Hn2(max, n2) + Hn3(max, n3);  
 return (y);  
 }  
 public static double Max1(double x1, double x2, double x3, double nn, double r)  
 {  
 double y;  
 y = 0;  
 if ((nn + r \* H1(x1, x2, x3) > 0))  
 {  
 y = nn + r \* H1(x1, x2, x3);  
 }  
  
 return (y);  
 }  
 public static double Max2(double x1, double x2, double x3, double nn, double r)  
 {  
 double y;  
 y = 0;  
 if ((nn + r \* H2(x1, x2, x3) > 0))  
 {  
 y = nn + r \* H2(x1, x2, x3);  
 }  
  
 return (y);  
 }  
 public static double Max3(double x1, double x2, double x3, double nn, double r)  
 {  
 double y;  
 y = 0;  
 if ((nn + r \* H3(x1, x2, x3) > 0))  
 {  
 y = nn + r \* H3(x1, x2, x3);  
 }  
  
 return (y);  
 }  
 public static double Max(double x1, double x2, double x3, double n1, double n2, double n3, double r)  
 {  
 double y;  
  
 y = Max1(x1, x2, x3, n1, r) + Max2(x1, x2, x3, n2, r) + Max3(x1, x2, x3, n3, r);  
 return (y);  
 }  
  
 static void Main(string[] args)  
 {  
 double dz11, max=1, max10, max20, max30, l1=0, l2=0, l3=0, n1=0, n2=0, n3=0, e1=0.0001, e2=0.0001, e3= 0.01, h=0.0001, z10, z20, z30, z11, z22, z33, dz1, dz2, dz3, ndf=1, f0, fy, fz, a, b, y, z, l, L, x, ndz, z01, z02, z03, mf, r=1, c, p;  
  
 int k = 0,M = 1000,n=0;  
  
 Console.Write("Введите x = ");  
 double z1 = Convert.ToDouble(Console.ReadLine());  
 Console.Write("Введите y = ");  
 double z2 = Convert.ToDouble(Console.ReadLine());  
 Console.Write("Введите z = ");  
 double z3 = Convert.ToDouble(Console.ReadLine());  
 Console.WriteLine();  
   
 do  
 {  
 k = 0;  
 n++;  
 while ((ndf > e1) && (k < M))  
 {  
  
 k = k + 1;  
  
  
 z10 = z1 + h;  
 z20 = z2 + h;  
 z30 = z3 + h;  
  
 max = Max(z1, z2, z3, n1, n2, n3, r);  
  
 max10 = Max(z10, z2, z3, n1, n2, n3, r);  
 max20 = Max(z1, z20, z3, n1, n2, n3, r);  
 max30 = Max(z1, z2, z30, n1, n2, n3, r);  
  
 dz1 = ((FF(z10, z2, z3) + Gl(z10, z2, z3, l1, l2, l3) + (r \* G(z10, z2, z3) \* G(z10, z2, z3) / 2) + Hn(max10, n1, n2, n3) / (2 \* r)) - (dz11 = FF(z1, z2, z3) + Gl(z1, z2, z3, l1, l2, l3) + (r \* G(z1, z2, z3) \* G(z1, z2, z3) / 2) + Hn(max, n1, n2, n3) / (2 \* r))) / h;  
 dz2 = ((FF(z1, z20, z3) + Gl(z1, z20, z3, l1, l2, l3) + (r \* G(z1, z20, z3) \* G(z1, z20, z3) / 2) + Hn(max20, n1, n2, n3) / (2 \* r)) - (dz11 = FF(z1, z2, z3) + Gl(z1, z2, z3, l1, l2, l3) + (r \* G(z1, z2, z3) \* G(z1, z2, z3) / 2) + Hn(max, n1, n2, n3) / (2 \* r))) / h;  
 dz3 = ((FF(z1, z2, z30) + Gl(z1, z2, z30, l1, l2, l3) + (r \* G(z1, z2, z30) \* G(z1, z2, z30) / 2) + Hn(max30, n1, n2, n3) / (2 \* r)) - (dz11 = FF(z1, z2, z3) + Gl(z1, z2, z3, l1, l2, l3) + (r \* G(z1, z2, z3) \* G(z1, z2, z3) / 2) + Hn(max, n1, n2, n3) / (2 \* r))) / h;  
  
  
 ndf = Math.Sqrt(dz1 \* dz1 + dz2 \* dz2 + dz3 \* dz3);  
  
   
 l = 0.001;  
 a = 0;  
 b = 3;  
 x = (a + b) / 2;  
 L = b - a;  
 while (L > l)  
 {  
  
 f0 = FF(z1 - x \* dz1, z2 - x \* dz2, z3 - x \* dz3);  
 y = a + L / 4;  
 z = b - L / 4;  
 fy = FF(z1 - y \* dz1, z2 - y \* dz2, z3 - y \* dz3);  
 fz = FF(z1 - z \* dz1, z2 - z \* dz2, z3 - z \* dz3);  
  
 if (fy < f0)  
 {  
 b = x;  
 x = y;  
 }  
 else if (fz < f0)  
 {  
 a = x;  
 x = z;  
 }  
 else  
 {  
 a = y;  
 b = z;  
 }  
  
 L = b - a;  
 }  
  
 z11 = z1 - x \* dz1;  
 z22 = z2 - x \* dz2;  
 z33 = z3 - x \* dz3;  
  
 mf = Math.Abs(FF(z11, z22, z33) - FF(z1, z2, z3));  
  
 z01 = z11 - z1;  
 z02 = z22 - z2;  
 z03 = z33 - z3;  
  
 z1 = z11;  
 z2 = z22;  
 z3 = z33;  
  
 ndz = Math.Sqrt(z01 \* z01 + z02 \* z02 + z03 \* z03);  
  
 if ((ndz < e2) && (mf < e2))  
 {  
 goto step2;  
 }  
 }  
 step2:  
  
 p = (r \* G(z1, z2, z3) \* G(z1, z2, z3) / 2) + Hn(max, n1, n2, n3) / (2 \* r);  
  
 Console.WriteLine("\nhn = {0}", Hn(max, n1, n2, n3));  
 Console.WriteLine("\nG^2 = {0}", G(z1, z2, z3) \* G(z1, z2, z3));  
   
 l1 = l1 + r \* G1(z1, z2, z3);  
 l2 = l2 + r \* G2(z1, z2, z3);  
 l3 = l3 + r \* G3(z1, z2, z3);  
 n1 = Max1(z1, z2, z3, n1, r);  
 n2 = Max2(z1, z2, z3, n2, r);  
 n3 = Max3(z1, z2, z3, n3, r);  
 c = 4;  
 r = r \* c;  
   
 Console.WriteLine("\np = {0}", p);  
  
 } while ((p > e3) && (n < 100));  
  
 Console.WriteLine();  
 Console.WriteLine("n = {0}", n);  
   
 Console.WriteLine("\nL=({1},{3},{5}) \n\nn=({0},{2},{4})", n1, l1, n2, l2, n3, l3);  
 Console.WriteLine("\nКолличество шагов = {0}", k);  
 Console.WriteLine("\nКоординаты точки минимума:");  
 Console.WriteLine("\nx = {0},\ny = {1},\nz = {2}", z1, z2, z3);  
 Console.WriteLine("\nЗначение функции f(x,y,z) = {0}", FF(z1, z2, z3));  
 Console.ReadKey();   
   
 }  
 }  
}