Program HOOKJIVES;

{$E+}

{$N+}

uses crt;

const

nv = 2; { numero de variaveis }

type

 vetor = array[1..nv] of real;

var

 y : extended;

 x : vetor;

 erro, alfa, delta : real;

 ch : char;

procedure pause(texto:string);

 var

 ch1,ch2 :char;

 lin,col :byte;

 begin

 col:=wherex;

 lin:=wherey;

 gotoxy (35,25);

 textattr:=white + red\*16 + blink;

 write(texto);

{ som; }

 ch1:=readkey;

 if (ch1=#0) then

 ch2:=readkey

 else

 ch2:=#0;

 write(#13); {#13 = Enter}

 textattr:=lightgray;

 clreol;

 gotoxy (col,lin);

 end;

Function Q(x : vetor):extended;

begin

 { Q:= sqr(sqr(x[1]-2)) + sqr(x[1]-2\*x[2]);}

 Q:=sqr(x[1]) + sqr(x[2]) + 1;

end;

Procedure Imprime2(k:word;x : vetor;Delta:real);

var

 ii :byte;

begin

writeln;

if (wherey=25) then

 clrscr;

write (K:3,' ');

write ('(');

for ii:=1 to nv-1 do

 write(x[ii]:6:12,', ');

write (x[nv]:6:12,') ');

write (Q(x):6:8,' ');

write (Delta:6:8);

pause('Parei...');

end;

Procedure Leitura;

var

 ii :byte;

begin

 writeln(' PONTO INICIAL');

 writeln;

 for ii:=1 to nv do begin

 writeln(' x',ii,' = ');

 writeln;

 end;

 writeln(' Erro = ');

 writeln;

 writeln(' Alfa = Fator de aceleracao');

 writeln;

 writeln(' Delta = Passo');

 for ii:=1 to nv do begin

 gotoxy(10, 1+2\*ii);

 readln(x[ii]);

 end;

 gotoxy(10, 1+2\*nv+2); readln(erro);

 gotoxy(10, 1+2\*nv+4); readln(alfa);

 repeat

 gotoxy(10,1+2\*nv+6);

 readln(delta);

 if (delta<=erro) then

 pause('Delta tem que ser maior que erro...');

 until (delta>erro);

 pause('Pressione alguma tecla...');

 clrscr;

 writeln(' PONTO INICIAL');

 for ii:=1 to nv do

 write ('X(',ii,') = ',x[ii]:6:8,' ');

 writeln;

end;

Procedure Hooke\_Jeeves\_Discreto(var x : vetor);

const

 db : array[1..2] of vetor =((1,0),(0,1));

 { base canonica }

var

 ii,j,k : word;

 Aux,y : vetor;

begin

 k:=1;

 writeln(' k x(k) Q(x) Delta');

 repeat

 y:=x;

 for j:=1 to nv do

 begin

 for ii:=1 to nv do

 Aux[ii] := y[ii] + Delta\*db[j][ii];

 if (Q(Aux)<Q(y)) then

 y:=Aux

 else

 begin

 for ii:=1 to nv do

 Aux[ii] := y[ii] - Delta\*db[j][ii];

 if (Q(Aux)<Q(y)) then

 y:=Aux;

 end;

 end;

 if Q(y)<Q(x) then

 begin

 for ii:=1 to nv do

 Aux[ii] := y[ii]+Alfa\*(y[ii]-x[ii]);

 If Q(Aux)<Q(y) then

 x := Aux

 else

 x := y;

 end

 else

 if Delta>=Erro then

 Delta:=Delta/2;

 Imprime2(k,x,Delta);

 k:=k+1;

 until Delta<Erro;

end; { Hooke\_Jeeves\_Discreto }

 {\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Programa Principal \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*}

begin

 Clrscr;

 Leitura;

 writeln;

 Hooke\_Jeeves\_Discreto(x);

 pause('Fim!!!...');

end.