```
Program Barreiras;

{$E+}
{$N+}
uses crt;
const
   nr   = 8;   { numero de restricoes }
   nv   = 4;   { numero de variaveis }

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

var
     y                                : extended;
     G                                : array[1..nr] of extended;
     x, d                             : vetor;
     mu, beta, erro, alfa, delta      : real;
     I                                : word;
     Pare                             : Boolean;
     ch                               : char;
     xasterisco4                      : real;

procedure som;
   var
      j          :byte;
      i          :word;
   begin
      for j:= 1 to 3 do begin
         for i:=400 to 1000 do
            sound(i);
         for i:=1000 downto 400 do
            sound(i);
      end;
      nosound;
   end;


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;    }
      repeat
         ch1 := readkey;
      until ch1 = #13;
      if (ch1=#0) then
         ch2:=readkey
      else
         ch2:=#0;
      write(#13);      {#13 = Enter}
      textattr:=lightgray;
```

```
      clreol;
      gotoxy (col,lin);
   end;

Function F(x : vetor):extended;
begin
   F:= x[4];
end;


Procedure Restricoes(Var x: vetor);
  begin
   G[1] := -x[1]-2;
   G[2] :=  x[1]-5;
   G[3] := -x[2]-3;
   G[4] :=  x[2]-8;
   G[5] := -x[3]-1;
   G[6] :=  x[3]-10;
   G[7] := -(sqr(x[1]) + sqr(x[2]) + sqr(x[3])) - x[4];
   G[8] := -(sqr(x[1]) + sqr(x[2]) + sqr(x[3])) - xasterisco4 - 0.01;


end;


Function B(x : vetor): real; { funcao barreira }
var
  I    : word;
  Soma : real;
begin
   soma := 0;
   Restricoes(x);
   for i:=1 to nr do;
       soma := soma - (1/G[i]);
   B := Soma;
end;


Function Q(x : vetor): extended;  { para o metodo das barreiras }
begin
   Q := F(x) + mu*B(x);
end;


Procedure Imprime1(x: vetor);
var
   ii      :byte;
begin
writeln;
writeln('I       = ',I:3);
writeln('MI      = ',mu:6:8);
write  ('X       = (');
for ii:=1 to nv-1 do
   write(x[ii]:6:12,', ');
writeln(x[nv]:6:12,')');
writeln('F(x)    = ',F(x):6:8);
writeln('Q(x)    = ',Q(x):6:8);
```

```
writeln('mi*B(x) = ',mu*B(x):6:8);
writeln;
pause('Pressione alguma tecla...');
end;

Procedure Imprime2(k:word;x : vetor;Delta:real);
var
   ii      :byte;
begin
writeln;
if (wherey=26) then begin
   pause('Vou limpar a tela...');
   clrscr;
end;
write  (K:3,'   ');
write  ('(');
for ii:=1 to nv-1 do
   write(x[ii]:6:12,', ');
write  (x[nv]:6:12,')   ');
write  (F(x):6:8,'   ');
write  (Delta:6:8);
end;


Function Interno(x:vetor):boolean;
var
   pert : boolean;
   j    : byte;
begin
   Pert := true;
   Restricoes(x);
   for j := 1 to Nr do
      if G[j] > 0 then
         pert := False;
   Interno := Pert;
end;



Procedure Hooke_Jeeves_Discreto(var x : vetor);
const
   db : array[1..4] of vetor  =((1,0,0,0),(0,1,0,0),(0,0,1,0),
                                (0,0,0,1));
                                          { 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))  and  Interno(Aux) then
            y:=Aux
         else
            begin
               for ii:=1 to nv do
                  Aux[ii] := y[ii] - Delta*db[j][ii];

               if (Q(Aux)<Q(y))  and  Interno(Aux) then
                  y:=Aux;
            end;
      end;

      if (Q(y)<Q(x))   and  Interno(y) then
         begin
            for ii:=1 to nv do
               Aux[ii] := y[ii]+Alfa*(y[ii]-x[ii]);

                  If  (Q(Aux)<Q(y))  and  Interno(Aux) 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 }


Procedure Leitura;
var
  ii      :byte;
begin
  Repeat
    writeln(' PONTO INICIAL');
    writeln;
    for ii:=1 to nv do begin
       writeln(' x',ii,'    = ');
       writeln;
    end;
    writeln(' Erro  = ');
    writeln;
    writeln(' æ     = ');
    writeln;
    writeln(' Beta  =                  Multiplica æ');
    writeln;
    writeln(' Alfa  =                  Fator de aceleracao');
    writeln;
    writeln(' delta =                  Passo');
    writeln;
    writeln(' xasterisco4 =            Solucao do problema 2 ');
```

```
    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(mu);
    gotoxy(10, 1+2*nv+6);    readln(beta);
    gotoxy(10, 1+2*nv+8);    readln(alfa);



    repeat
       gotoxy(10,1+2*nv+10);    readln(delta);
       if (delta<=erro) then
          pause('Delta tem que ser maior que erro...');
    until (delta>erro);
    gotoxy(16, 1+2*nv+12);   readln(xasterisco4);
    if not(Interno(x)) then
        begin
           gotoxy(25,1+2*nv+14);
           write('Ponto nao pertence ao interior');
           pause('Pressione alguma tecla...');
        end;
    clrscr;
  until Interno(x);
    writeln('    PONTO INICIAL');
    for ii:=1 to nv do
       write  ('X(',ii,') = ',x[ii]:6:8,'   ');
    writeln;
end; { Leitura }


{************************** Programa Principal
***************************}
begin
   repeat
     Clrscr;
     Pare := false;
     I := 1;
     Leitura;
     writeln;
     repeat
       Hooke_Jeeves_Discreto(x);
       Imprime1(x);
       if (mu*B(x) < Erro) then
          Pare := true
       else
         begin
            mu := beta*mu;
            I  := I + 1;
         end;
     until Pare;
     writeln;
     writeln('                    DESEJA REPETIR ? (S/N) ');
     repeat
        ch:=readkey;
        ch:=UpCase(ch);
```

```
        if not (ch in ['S','N']) then
           som;
     until (ch in ['S','N']);
   until (UpCase(ch)='N');
   clrscr;



{ PROPLEMA 3
  Min. F(x) = x[4];
  Restricoes: x[1]>=-2;
              x[1]<=5 ;
              x[2]>=-3;
              x[2]<=8 ;
              x[3]>=-1;
              x[3]<=10;
              -(sqr(x[1]) + sqr(x[2]) + sqr(x[3])) <=x[4];

              -(sqr(x[1]) + sqr(x[2]) + sqr(x[3])) <= xasterisco -0.01;
                          ;
  Este problema esta sendo resolvido usando-se o metodo das BARREIRAS e
  HOOKE-JEEVES (com passo discreto).


}



end.
```