

ДОДАТОК А

Програма

```

unit Unit1copy;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  ExtCtrls, StdCtrls;
type
  TForm1 = class(TForm)
    Button1: TButton;
    Button2: TButton;
    PaintBox1: TPaintBox;
    procedure Button1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
const eps1=1e-10;nmax=600;eps=1e-9;eps2=1e-4; ninit=20;
acur=1e-8;
  ia=50000;
  nmax5=15000;
  epsb=1e-14;small=1e-18;
{BPMPD}
  const maxm= 10000;
  const maxn= 10000;
  const maxnz= 50000;
  const maxqnz=1;
type tnz =array [1..maxnz] of integer;
type tqnz =array [1..maxqnz] of integer;
type tn =array [1..maxn] of integer;
type tmn =array [1..maxm+maxn] of integer;
type tdnz =array [1..maxnz] of double;
type tdqnz=array [1..maxqnz] of double;
type tdm =array [1..maxm] of double;
type tdn =array [1..maxn] of double;
type tdmn =array [1..maxm+maxn] of double;
{BPMPD}
const tochn=1e-4;
type tr2=array[1..2] of double;
  sin5=array[0..nmax5] of integer;
  lnr5=array[0..nmax5]of boolean;
  tiar=array[1..IA] of double;
  tiai=array[1..IA] of integer;
  trn5=array[0..nmax5] of double;
  trn=array[1..nmax] of double;
  tbn=array[1..nmax] of byte;
  tltn=array[-5..nmax,-100..nmax]of boolean;
  trn=array[0..nmax] of double;
  { trn1=array[1..nmax] of double;}
  tni=array[0..nmax] of integer;
  t2ni=array[0..500] of integer;
  t2ri=array[0..500] of double;
  tr200=array[1..500]of double;
  ti200=array[1..500]of integer;
type tr4=array[1..2]of double;
  tr5=array[0..5]of double;

```

```

t_p_tnr ^= tnr;
t_p_xfull ^= tdm;
var Form1: TForm1;
  MyPaintBox: TPaintBox;
  {graphic} fposl2: text;
{
  ass, bs, cs, x, miny, abkv, miy, miy1, alfa, betta, nch, sqrnch,
  sqrnchf, nchf, bettaf, bettaa, ncha, sqrncha, romem, ndel, kest, ndouble, ss, ss1,
  mo1max, zcurr, zcurra, delh, mo_2, sigma: double; }
  {endgraph}
  FName: String;
  ipar, jpar: tnz;
  rpar: tdnz;
  iparami, nparami, isn, icou, tofp
: integer;
  xfull: tdm;
{
  h, m, hund, s3, h1, m1, s31: word; }
  dtime: longint;
  llll, lprob, lprob2, lror: boolean;
  c: char;
  xex, yex: tr5;
  nex, idelmax, iglcount, ilid: integer;
  zest4min, zest5min, zest1min, zest2min, lengmin, widx, widy, widz, xcyl, ycyl, xsph, ysph, zsph, rcyl, ztreebest: double;
var xan, yan, ran: tr4;
  rr1, xx, yy: array[1..30, 1..30] of double;
  nr, ix, ix1, n, nixn, i: integer;
rs, r, x, y, z, xbr, ybr, xbeg, ybeg, zbeg, rbeg, rin, rin1, rin2, xb, yb, rb, rvl, rvh, rvlg, rvhg, rh, xh, yh, rph, rpl, rphh, rpll, rvlh, rvhh, rdut, de
lr, xbddd,
  ybddd, rbddd, zbddd, xbddd1, ybddd1, rbddd1, rlow, rhigh, xtreeb, ytreeb, ztreeb, rtreeb: tnr;
xmeme, ymeme, zmeme, rmeme, xbest, ybest, zbest, rbest, xmeme, ymeme, zmeme, rmeme, xbdut, ybdut, zbdut, rbdut, xbdut
1, ybdut1, rbdut1,
  xbloc, ybloc, rbloc, zbloc, xbro, ybro, zbro, rbro, rinbro: tnr;
mp, nofvr, mph, mpl, mphi, mpli, mprin, mpr: tni;
a0, a01, b0, zest, zestmin, zestddd, rmax, rmin, rmin1, rmm, rpm, xxx, yyy, rrr, rmaxest, rmaxi, rpm1, zestb, lengb: double;
scale, zbestz, delmax: double;
x1r, x2r, rsr: tr200;
imser, nnm, iby, i8, nact, nactl: integer;
lbeg, lx: boolean;
fff, flo, fff1, fgr: text;
g1, g2, ii, jj, j, nmk, nmk1, io, jo, iio, jp, inr, luk, ipa, jpa: integer;
imk, jmk, imk1, jmk1, iim, igm, jgm, kmk, kgm, ioptmm, joptmm, koptmm, iioptmm: t2ni;
mdm, rmk1, {xmk1, ymk1, }rmk, xmk, ymk: t2ri;
  LENL, LENU, NCP, LROW, LCOL: integer;
iw11, iw12: tiar;
lkod, ll, lkod2: boolean;
{IP1, IP2: sin5; }
IND1, IND2, IND1a, IND2a: tiar;
A, aa: tiar;
W, B, ba: trn5;
IW1, IW2, IW3, IW4, IW5, IW6, IW7, IW8: sin5;
G, U, rat, rab, nko, sqrrin, sqrr, rab, zestbeg: double;
mpi, irand: integer;
var mc: tltn;
  ftest, f1: text;
  n2, i1, icf, n3, idut, jdut: integer;
  m1, m2, or1, or2: sin5;
  igl: longint;
  it, proxod, itobl: integer;
  lr, lr1, lres, lrazr, lr1, lrh, lrhg, lri, lrj, lrinj, lrll, lrhh, lrhhh, lrhhh1, lrhhhddd, lrlll, lmemdel: lnr5;
  lty, ltobl: boolean;
  Nrestr, DimmatrB3: integer;
  xtarg, dt, kaux, xtargin, dt1, dtr: trn5;
  min, ss, svch, wtarg, wtarg2, wtarg0, wtargm, rr, zestb1, dis, ha, hb, xrab, yrab, zrab: double;
  idel, iadd, isimp: integer;

```

```

code,res,lkj,sedl:boolean;
hc:array[1..nmax5]of double;
nz,nza:integer;
var lm:lnr5;
mn:real;mnozh:trn5;
nofvio,nofvio1,k:integer;
iv,jv,iv1,jv1,kv,kv1,iact,jact,iactl,jactl:tiai;
ikr:integer;
nachrj1,nachnp2,nimi,nimj:integer;
nresh,nlim,nsd,nreshbr,nreshmax:integer;
xt,yt,xt1,nach,kon,inx,iny,delit,rj,sum,sumkub,sumbegin,sumr:double;
tol,tolcyl,deltax,deltay,rab1:double;
lresh:boolean;
sm1,sm2,imin,iglob,irpl:integer;
lhc,lparami:boolean;
nofneigh:array[1..nmax]of integer;
mofn:array[1..nmax,1..30]of integer;
iper,jper,dper:integer;
xper,yper,rper:array[1..ninit,1..nmax]of double;
sm1m,sm2m,summax:array[1..nmax]of double; kappa,kappab,sqrdel,sqrmin,sqrdelc:double;
{BPMPD}
var {m,n,nz,}qn,qnz,{i,}mems,nbp:integer;
var acolidx:tnz;
var qcolidx:tnqz;
var acolcnt,qcolcnt,ukaz:tn;
var acolnzs:tdnz;
var qcolnzs:tdqnz;
var rhs:tdm;
var obj:tdn;
var lbound,ubound,primal,dual:tdmn;
var status:tmn;
var opt,big,mult:double;
var codebp:integer;
{ type tnz =array [1..maxnz] of integer;
type tnqz =array [1..maxqnz] of integer;
type tn =array [1..maxn] of integer;
type tmn =array [1..maxm+maxn] of integer;
type tdnz =array [1..maxnz] of double;
type tdqnz=array [1..maxqnz] of double;
type tdm =array [1..maxm] of double;
type tdn =array [1..maxn] of double;
type tdmn =array [1..maxm+maxn] of double;
}
{inhib
}
{цилиндр}
sxq,syq,szq,s0xq,s0yq,s0zq,modsq,cosalfq,cosbetq,cosgamq,cosalb,costetq,
sintetq,singamq,aq,bq,rcylq:double;
ll1,ll2:boolean;
wx,wy,wz,cosgamz,singamz:double;
gam2,gam1,psi2:double;
sintet,costet:double;
ll3,ll4:boolean;
dcub:double;
wmod,cosalx,cosbety,cosalbet2:double;
aml,bml,cml,dml,sqrabc,aqp,bqp,s0xqp,s0yqp,s0zqp,singamqp:double;
(* procedure bpmpd(
var m,n,nz,qn,qnz:integer;
var acolcnt:tn; var acolidx:tnz; var acolnzs:tdnz;
var qcolcnt:tn; var qcolidx:tnqz; var qcolnzs:tdqnz;
var rhs:tdm; var obj:tdn; var lbound,ubound:tdmn;
var primal:tdmn; var dual:tdmn; var status: tmn;
var big:double; var code:integer; var opt:double; var usemem:integer ); stdcall; external 'bpmpd.dll';*)

```

```
{BPMPD}
{function SetParameter(key:integer; val:double):double; stdcall; external 'ILOPT_CIRCLES.dll';
function GetParameter(key:integer):double; stdcall; external 'ILOPT_CIRCLES.dll';
function RunOptimisation(n:integer; p_xfull:t_p_xfull; p_rminn:t_p_tnr; p_rmaxx:t_p_tnr):integer; stdcall; external
'ILOPT_CIRCLES.dll'; }
procedure OptimiseDefault; cdecl; external 'ILOPT_SOLVER.dll';
{ *****}
procedure IP*(
                                m,n,nnz:integer; //peremennyyh, ogranichenij, vseh nenulevyh
                                var apan:tiar ; // 0...nnz-1 -- znachenija nenulevyh - coefficienty
                                var ind_i,ind_j: tiai; // ego index :: 0...nnz-1 -- stroki;
                                // index :: 0...nnz-1 -- tolbza
                                var bpan:trn5; // 0...m-1 -- pravyje chasti
                                var mybl:trn5;
                                var mybh:trn5;
                                var tc:tsin5;
                                var cpan:trn5; // 0...n-1 -- coefficienty celevoi funkzii
                                var tcode:integer;
                                var kappa:double; // obj f-n
                                var s:trn5 // 0...n-1 -- solution
                                *) ; cdecl; external 'InnerPointDLL.dll';
procedure Optimise; cdecl; external 'ILOPT_SPHERES.dll';

procedure IP1; cdecl; external 'InnerPointDLL1.dll';
{ *****}

implementation
{$R *.DFM}

function LeadingZero(w : Word) : String;
var
  s : String;
begin
  Str(w:0,s);
  if Length(s) = 1 then
    s := '0' + s;
  LeadingZero := s;
end;
(*888888888888888888888888*)
function usage:double;
var s:double;i:integer;
begin
  s:=0;
  for i:=1 to ix do
    s:=s+pi*sqr(r[mp[i]]);
    s:=s/zest/b0*100;
  usage:=s;
  end;
  function scal:double;
var scalex,scaley,scale1:double;
begin
  scalex:=a0/(MyPaintBox.Width-2-20);scaley:=b0/((MyPaintbox.Height-2)-50);
  if(scalex>scaley) then
    scale1:=scalex
  else
    scale1:=scaley;
  scal:=scale1;
end;
procedure patt(a0,b0:real);
var i:integer; s:string[9];
begin
  begin
  with MyPaintBox.Canvas do begin
```

```

MoveTo(20,MyPaintbox.Height-50-round(b0/scale));
lineTo(round(a0/scale)+20,MyPaintbox.Height-50-round(b0/scale));
MoveTo(20,MyPaintbox.Height-50);
lineTo(20,MyPaintbox.Height-50-round(b0/scale));
MoveTo(20,MyPaintbox.Height-50);
lineTo(round(a0/scale)+20,MyPaintbox.Height-50);
MoveTo(round(a0/scale)+20,MyPaintbox.Height-50);
lineTo(round(a0/scale)+20,MyPaintbox.Height-50-round(b0/scale));
end;
with MyPaintBox.Canvas do begin
MoveTo(20,MyPaintbox.Height-50-round(b0/scale));
lineTo(round(leng/scale)+20,MyPaintbox.Height-50-round(b0/scale));
MoveTo(20,MyPaintbox.Height-50);
lineTo(20,MyPaintbox.Height-50-round(b0/scale));
MoveTo(20,MyPaintbox.Height-50);
lineTo(round(leng/scale)+20,MyPaintbox.Height-50);
MoveTo(round(leng/scale)+20,MyPaintbox.Height-50);
lineTo(round(leng/scale)+20,MyPaintbox.Height-50-round(b0/scale));
end;
end;
end;
procedure showdir(mn:real);
var i,ix,ix1,iy1,ix2,iy2:integer;
begin
MyPaintbox.Canvas.Pen.Color:=(CLred);
for i:=2 to n2+1 do
begin
if (i-1)mod 2 =1 then
begin
ix1:=round(x[(i-1)div 2+1]/scale)+20;
ix2:=round((x[(i-1)div 2+1]+dt[i])/scale)+20;
end
else
begin
iy1:=MyPaintbox.Height-50-round(y[(i-1)div 2]/scale);
iy2:=MyPaintbox.Height-50-round((y[(i-1)div 2]+dt[i])/scale);
with MyPaintBox.Canvas do begin
MoveTo(ix1,iy1);
lineTo(ix2,iy2);
end;
end;
end;
MyPaintbox.Canvas.Pen.Color:=(CLblack);
end;
procedure showcir(xc,yc,rc,rinc:double;ix,mpix,col:integer);
var ix1,iy1,ix2,iy2,i:integer; s,s1:string;
begin
if ix=0 then
begin
with MyPaintBox.Canvas do begin
str(zest:9:5,s); s:='zest='+s+' tol=';
str(tol:8:6,s1);
s:=s+s1;
MyPaintbox.Canvas.TextOut(10,MyPaintbox.Height-20,s);
end;
end;
ix1:=round((xc)/scale)+20;
iy1:=MyPaintbox.Height-50-round((yc)/scale);
begin
{ if rc>rinc then}
{ MyPaintbox.Canvas.Pen.Color:=(clblack);}
begin
if col=0 then MyPaintbox.Canvas.Pen.Color:=(CLwhite);

```

```

    if col=1 then MyPaintbox.Canvas.Pen.Color:=(CLblack);
    if col=8 then MyPaintbox.Canvas.Pen.Color:=(CLgray);
    if col=4 then MyPaintbox.Canvas.Pen.Color:=(CLgreen);
    if col=3 then MyPaintbox.Canvas.Pen.Color:=(CLblue);
    end;
MyPaintbox.Canvas.Pen.Color:=col;
    iy2:=trunc(rinc/scale)+1;
    ix2:=trunc(rinc/scale)+1;
    MyPaintbox.Canvas.Pen.Style:=psSolid;
    if (rc<rinc)and(abs(rc-rinc)>1e-9) then
MyPaintbox.Canvas.Pen.Color:=(clred);
        {if (rc>rinc)and(abs(rc-rinc)>1e-2) then MyPaintbox.Canvas.Pen.Color:=(clwhite);}
        {if (rc<rinc)and(abs(rc-rinc)>1e-9) then MyPaintbox.Canvas.Pen.Style:=psDash;
        if (rc>rinc)and(abs(rc-rinc)>1e-2) then MyPaintbox.Canvas.Pen.Style:=psDash;}
        MyPaintBox.Canvas.Ellipse(ix1-ix2,iy1-iy2,ix1+ix2,iy1+iy2);
        iy2:=trunc(rc/scale)+1;
        ix2:=trunc(rc/scale)+1;
        MyPaintbox.Canvas.brush.style:=bsClear;
        if (abs(rc-rinc)>5e-2) then
        begin
        MyPaintbox.Canvas.Pen.Color:=col;
        MyPaintBox.Canvas.Ellipse(ix1-ix2,iy1-iy2,ix1+ix2,iy1+iy2);
        end;
        MyPaintbox.Canvas.Pen.Color:=col{(CLblack)};

        (*if (rc<rinc)and(abs(rc-rinc)>1e-7) then MyPaintbox.Canvas.Pen.Color:=(clred);*)
        if rc>rinc-tochn then
        MyPaintBox.Canvas.Ellipse(ix1-ix2,iy1-iy2,ix1+ix2,iy1+iy2);
        {if rc>rinc then} {MyPaintbox.Canvas.Pen.Color:=(CLblack);}
        begin
        if col=0 then MyPaintbox.Canvas.Pen.Color:=(CLwhite);
        if col=1 then MyPaintbox.Canvas.Pen.Color:=(CLblack);
        if col=8 then MyPaintbox.Canvas.Pen.Color:=(CLgray);
        if col=4 then MyPaintbox.Canvas.Pen.Color:=(CLgreen);
        if col=3 then MyPaintbox.Canvas.Pen.Color:=(CLblue);
        end;
        end;
    {str(mpix,s);
    MyPaintbox.Canvas.TextOut(ix1+2,iy1+2,s);}
    str(ix,s);
    for i:=1 to 20 do
    MyPaintbox.Canvas.TextOut(ix1-12,iy1-12,s);
    ix1:=round((xc+a0)/scale)+20;
    iy1:=MyPaintbox.Height-50-round((yc+b0)/scale);
    {setlinestyle(0,0,3);}
    {line(ix1,iy1,ix1,iy1);}
    if rc>1-tochn then ;
    {MyPaintBox.Canvas.Ellipse(ix1-1,iy1-1,ix1+1,iy1+1);}
    {setlinestyle(0,0,1);}
    end;
    begin
    it:=it+1;
    nz:=nz+1;    b[it]:=0;
    a[nz]:=-1; ind1[nz]:=it; ind2[nz]:=1;
    for i:=1 to dimmatrb3-1 do
    if (abs(xtargin[i])>eps) then
    begin
    nz:=nz+1;
    a[nz]:=-xtargin[i]; ind1[nz]:=it; ind2[nz]:=i+1;
    end;
    end;
end;

```

```

to_mps(it,dimmatrb3,nz,a,ind1,ind2,b);
end;
procedure addconstr1(i,j:integer;eps:double);
begin
  it:=it+1;
  or1[it]:=i;or2[it]:=j;
  b[it]:=0; {1e-5} if lprob then b[it]:=1e-2;
  if j=n+1 then
    begin
      nz:=nz+1;
      a[nz]:=1; ind1[nz]:=it; ind2[nz]:=i*2;
      hc[ind2[nz]]:=hc[ind2[nz]]+a[nz];
      if lr[i] then
        begin
          nz:=nz+1;
          a[nz]:=-1; ind1[nz]:=it; ind2[nz]:=nofvr[i];
          hc[ind2[nz]]:=hc[ind2[nz]]+a[nz];
        end;
      end;
    end;
  if j=n+2 then
    begin
      nz:=nz+1;
      a[nz]:=1; ind1[nz]:=it; ind2[nz]:=i*2+1;
      hc[ind2[nz]]:=hc[ind2[nz]]+a[nz];
      if lr[i] then
        begin
          nz:=nz+1;
          a[nz]:=-1; ind1[nz]:=it; ind2[nz]:=nofvr[i];
          hc[ind2[nz]]:=hc[ind2[nz]]+a[nz];
        end;
      end;
    end;
  if j=n+3 then
    begin
      nz:=nz+1;
      a[nz]:=-1; ind1[nz]:=it; ind2[nz]:=i*2+1;
      hc[ind2[nz]]:=hc[ind2[nz]]+a[nz];
      if lr[i] then
        begin
          nz:=nz+1;
          a[nz]:=-1; ind1[nz]:=it; ind2[nz]:=nofvr[i];
          hc[ind2[nz]]:=hc[ind2[nz]]+a[nz];
        end;
      end;
    end;
  if j=n+4 then
    begin
      nz:=nz+1;
      a[nz]:=-1; ind1[nz]:=it; ind2[nz]:=i*2;
      hc[ind2[nz]]:=hc[ind2[nz]]+a[nz];
      if lr[i] then
        begin
          nz:=nz+1;
          a[nz]:=-1; ind1[nz]:=it; ind2[nz]:=nofvr[i];
          hc[ind2[nz]]:=hc[ind2[nz]]+a[nz];
        end;
      end;
    end;
  if lleng then
    begin
      nz:=nz+1;
      a[nz]:=1; ind1[nz]:=it;
      ind2[nz]:={n2+2}1;
      hc[ind2[nz]]:=hc[ind2[nz]]+a[nz];
    end;
  end;
  if (j<=n)and(j>0) then

```

```

begin
if not(lleng)and((lr[i])or(lr[j])) then
begin
nz:=nz+1;
a[nz]:=-0.1; ind1[nz]:=it; ind2[nz]:=1;
end;
nz:=nz+1;
a[nz]:=2*(x[i]-x[j]); ind1[nz]:=it; ind2[nz]:=i*2;
hc[ind2[nz]]:=hc[ind2[nz]]+a[nz];
nz:=nz+1;
a[nz]:=-2*(x[i]-x[j]); ind1[nz]:=it; ind2[nz]:=j*2;
hc[ind2[nz]]:=hc[ind2[nz]]+a[nz];
nz:=nz+1;
a[nz]:=2*(y[i]-y[j]); ind1[nz]:=it; ind2[nz]:=i*2+1;
hc[ind2[nz]]:=hc[ind2[nz]]+a[nz];
nz:=nz+1;
a[nz]:=-2*(y[i]-y[j]); ind1[nz]:=it; ind2[nz]:=j*2+1;
hc[ind2[nz]]:=hc[ind2[nz]]+a[nz];
if (lr[i]) then
begin
nz:=nz+1;
a[nz]:=-2*(r[i]+r[j]); ind1[nz]:=it;ind2[nz]:=nofvr[i];
hc[ind2[nz]]:=hc[ind2[nz]]+a[nz];
end;
if (lr[j]) then
begin
nz:=nz+1;
a[nz]:=-2*(r[i]+r[j]); ind1[nz]:=it;ind2[nz]:=nofvr[j];
hc[ind2[nz]]:=hc[ind2[nz]]+a[nz];
end;
end;
end;
procedure formcoef1(eps1:double);
var i,j:integer;
begin
if lhc then
for i:=1 to n do
lr[i]:=false;
rvar1;
it:=0; nz:=0;
for i:=1 to n2+n+1 do hc[i]:=0;
for i:=1 to n do
for j:=i+1 to n+4 do
if mc[i,j] then
begin
addconstr1(i,j,eps1);
end;
for i:=1 to n do
if lr[i] then
if (lrh[i])or(lrhh[i])or(lrhhh[i]) then
begin
it:=it+1; nz:=nz+1;
a[nz]:=-1; ind1[nz]:=it; ind2[nz]:=nofvr[i]; b[it]:=0;
end;
for i:=1 to n do
if lr[i] then
if (lrl[i])or(lrll[i])or(lrlll[i]) then
begin
it:=it+1; nz:=nz+1;
a[nz]:=1; ind1[nz]:=it; ind2[nz]:=nofvr[i]; b[it]:=0;
end;

```

```

begin
if not(lleng) then
begin
it:=it+1; b[it]:=0;
if lprob then b[it]:=0.01;
for i:=1 to n2+nor do
if ((lhc)and(abs(hc[i])>eps))or((not(lhc))and(abs(xtargin[i])>eps)) then
begin
nz:=nz+1;
if lhc then
a[nz]:=hc[i] else
a[nz]:=-xtargin[i]; ind1[nz]:=it; ind2[nz]:=i+1;
end;
end;
if not(lleng) then
begin
nz:=nz+1;
a[nz]:=-1; ind1[nz]:=it; ind2[nz]:=1;
end;
if {not(lleng)}lprob then
if mc[-1,-1] then
begin
it:=it+1; b[it]:=0;
for i:=1 to n do
if lr[i] then
begin
nz:=nz+1; a[nz]:=-1; ind1[nz]:=it; ind2[nz]:=nofvr[i];
end;
end;
procedure showall;
var i:integer;
begin
MyPaintBox.Refresh;
patt(b0,a0);
for i:=1 to n do
begin
showcir(x[i],y[i],r[i],rin[i],i,i,1);
end;
patt(b0,a0);
end;
procedure i_vstavka;
label novprob,snov,lt2;
var ich:integer;
begin
ich:=0; n:=nbeg;
rmax:=0;rmin:=1e+37;
for i:=1 to n do
begin
if rin[i]>rmax then rmax:=rin[i];
if rin[i]<rmin then rmin:=rin[i];
end;
{isk vstavka}
for i:=1 to n do lr1[i]:=true;
sqrmax:=0;
novprob;;
nadd:=1; nbeg:=nbeg;
{n:=n-1;}
while n<nbeg+4 do
begin
n:=n+1; nprob1:=0; r[n]:=rmax;
snov;;
nprob1:=nprob1+1;
if nprob1=1000 then

```

```

begin
  nprob1:=0; r[n]:=r[n]*0.999;
end;
// x[n]:=random(10000)/10000*(leng-2*r[n])+r[n];
// y[n]:=random(10000)/10000*(wid-2*r[n])+r[n];
// repeat
lt2::
if random(2)=0 then
x[n]:=(leng)*random(10000)/10000 else x[n]:=-(leng)*random(10000)/10000;
if random(2)=0 then
y[n]:=(leng)*random(10000)/10000 else y[n]:=-(leng)*random(10000)/10000;
if random(2)=0 then
z[n]:=(leng)*random(10000)/10000 else z[n]:=-(leng)*random(10000)/10000;
if (sqr(x[n]-xsph)+sqr(y[n]-ysph)+sqr(z[n]-zsph)>sqr(leng-r[n]))and(tofp=2) then goto lt2;
//if (sqr(x[n]-xcyl)+sqr(y[n]-ycyl)>sqr(rcyl-r[n])) then goto lt2;
//until not(sqr(x[n]-xsph)+sqr(y[n]-ysph)+sqr(z[n]-zsph)>sqr(leng*(tofp mod 2 +1)-r[n])) or
// (sqr(x[n]-xcyl)+sqr(y[n]-ycyl)>sqr(rcyl-r[n]));
for i:=1 to n-1 do
  if sqr(x[n]-x[i])+sqr(y[n]-y[i])+sqr(z[n]-z[i])<sqr(r[n]+r[i]) then goto snov;
end;
  convio(nofvio,iv,jv,dis,2e-7);
end;
procedure formact(eps:double);
var i,j,nact:integer;
begin
  nact:=0;
  for i:=1 to n do
    begin
      for j:=i+1 to n+8 do
        begin
          if j<=n then rab1:=sqrt(sqr(x[i]-x[j])+sqr(y[i]-y[j])+sqr(z[i]-z[j]))-(r[i]+r[j]);
          if (j=n+8)and(tofp=2) then rab1:=-sqrt(sqr(x[i]-xsph)+sqr(y[i]-ysph)+sqr(z[i]-zsph))+(leng-r[i]);
          if (rab1<eps) then
            begin
              nact:=nact+1;
              m1[nact]:=i;
              m2[nact]:=j;
            end
          end;
        end;
      end;
    end;
  procedure dutlin;
  var code,ifull,nfull:integer;
  begin
    convio(nofvio,iv,jv,dis,2e-7);
    if (nofvio<>0)and(dis>2e-7) then
      convio(nofvio,iv,jv,dis,2e-7);
    zest:=0;
    for i:=1 to n do
      zest:=zest+(r[i]);
    repeat
      assignfile(fff,'pci.dat');
      rewrite(fff);
      //writeln(fff,'0 0 0 ',widx,' ',widy,' ',leng,' 0.5 0.5 0.5 0.5 0.5 0.5 1 1 0 0 0 ',n:3);
      //writeln(fff,-widx,' ',-widy,' ',-widx,' ',widx,' ',widy,' ',widx,' 0.5 0.5 0.5 0.5 0.5 0.5 2 1 0 0 0 ',n:3);
      writeln(fff,-leng,' ',leng,' ',leng,' ',leng,' ',leng,' ',leng,' 0.5 0.5 0.5 0.5 0.5 0.5 2 1 0 0 0 ',n:3);
      for i:=1 to n do
        writeln(fff,x[i],' ',y[i],' ',z[i],' ',rin2[i],' ',r[i],' 0');
      closefile(fff);
      zestbeg:=zest;
      for i:=1 to n do
        begin
          xbeg[i]:=x[i]; ybeg[i]:=y[i]; zbeg[i]:=z[i]; rbeg[i]:=r[i];

```

```

end;
until (abs(zest-zestbeg)<tochn/100)or(zest<zestbeg);
convio(nofvio,iv,jv,dis,2e-7);
if (nofvio<>0)and(dis>2e-7) then
convio(nofvio,iv,jv,dis,2e-7);
if zest<zestbeg then
for i:=1 to n do
begin
x[i]:=xbeg[i]; y[i]:=ybeg[i]; z[i]:=zbeg[i]; r[i]:=rbeg[i];
end;
lkod:=true;
for i:=1 to n do
if (r[i]<rin[i])and(abs(r[i]-rin[i])>1e-5) then lkod:=false;
//showall;
end;
procedure davit;
var code,ifull,nfull,icount:integer; leng1:double;
begin
zest:=0; icount:=0;
for i:=1 to n do
if sqrt(sqr(x[i]-xsph)+sqr(y[i]-ysph)+sqr(z[i]-zsph))+r[i]>zest then
zest:=sqrt(sqr(x[i]-xsph)+sqr(y[i]-ysph)+sqr(z[i]-zsph))+r[i];
leng:=zest;
convio(nofvio,iv,jv,dis,2e-7);
if (nofvio<>0)and(dis>2e-7) then
convio(nofvio,iv,jv,dis,2e-7);
repeat
icount:=icount+1;
assignfile(fff,'pci.dat');
rewrite(fff);
//
//          x y z r h R td tf f a ch
if tofp=1 then writeln(fff,'0 0 0 ',widx,' ',widy,' ',widz,' 0.5 0.5 0.5 0.5 0.5 0.5 0 0 0 0 0 ',n:3);
if tofp=2 then
//writeln(fff,-widx,' ',-widy,' ',-widz,' ',widx,' ',widy,' ',widz,' 0.5 0.5 0.5 0.5 0.5 0.5 2 4 0 0 0 ',n:3);
writeln(fff,-leng,' ',-leng,' ',-leng,' ',leng,' ',leng,' ',leng,' 0.5 0.5 0.5 0.5 0.5 0.5 2 4 0 0 0 ',n:3);
//writeln(fff,'0 0 0 ',widx,' ',widy,' ',widz,' 0.5 0.5 0.5 0.5 0.5 0.5 2 4 1 0 0 ',n:3);
if tofp=3 then writeln(fff,'0 0 0 ',widx,' ',widy,' ',widz,' 0.5 0.5 0.5 0.5 0.5 0.5 1 0 0 0 0 ',n:3);
for i:=1 to n do
writeln(fff,x[i],' ',y[i],' ',z[i],' ',rin2[i],' ',r[i],rin2[i]);
writeln(fff,zest);
//writeln(fff,'10');
closefile(fff);
zestbeg:=zest;
for i:=1 to n do
begin
xbeg[i]:=x[i]; ybeg[i]:=y[i]; zbeg[i]:=z[i]; rbeg[i]:=r[i];
end;
convio(nofvio,iv,jv,dis,2e-7);
//ip; {SASHA}
optimise;
assignfile(fff,'output.txt');
reset(fff);
readln(fff);readln(fff);readln(fff);
for i:=1 to n do
readln(fff,x[i],y[i],z[i],r[i]);
readln(fff,zest);
closefile(fff);
leng1:=0;
for i:=1 to n do
if sqrt(sqr(x[i]-xsph)+sqr(y[i]-ysph)+sqr(z[i]-zsph))+r[i]>leng1 then
leng1:=sqrt(sqr(x[i]-xsph)+sqr(y[i]-ysph)+sqr(z[i]-zsph))+r[i];
leng:=leng1;
convio(nofvio,iv,jv,dis,2e-7);

```

```

//showall;
{zest:=leng;}
until (abs(zest-zestbeg)<tochn/10)or(zest>zestbeg)or (nofvio>0)or(icount>100);
convio(nofvio,iv,jv,dis,2e-7);
if (nofvio<>0)and(dis>2e-7) then
convio(nofvio,iv,jv,dis,2e-7);
if (zest>zestbeg)or(nofvio>0) then
for i:=1 to n do
begin
zest:=zestbeg;
x[i]:=xbeg[i]; y[i]:=ybeg[i]; z[i]:=zbeg[i]; r[i]:=rbeg[i];
end;
leng:=0;
for i:=1 to n do
begin
if (tofp=1)or(tofp=3) then begin if z[i]+r[i]>leng then leng:=z[i]+r[i]; end;
if tofp=2 then begin if sqrt(sqr(xsph-x[i])+sqr(ysph-y[i])+sqr(zsph-z[i]))+r[i]>leng then leng:=sqrt(sqr(xsph-
x[i])+sqr(ysph-y[i])+sqr(zsph-z[i]))+r[i]; end;
end;
zest:=leng;
//lkod:=true;
//for i:=1 to n do
//if (r[i]<rin[i])and(abs(r[i]-rin[i])>1e-5) then lkod:=false;
//showall;
end;
procedure dutunlin;
var iter:integer; zestiter,zbegiter:double;
var code,ifull,nfull:integer;
begin
convio(nofvio,iv,jv,dis,2e-7);
if (nofvio<>0)and(dis>2e-7) then
convio(nofvio,iv,jv,dis,2e-7);
iter:=0;
zest:=0;
for i:=1 to n do
zest:=zest+sqr(r[i])*r[i];
zestiter:=zest; zbegiter:=zest;
rmax:=0;rmin:=1e+37;
for i:=1 to n do
begin
if rin[i]>rmax then rmax:=rin[i];
if (rin[i]<rmin)and(rin[i]>0.2) then rmin:=rin[i];
end;
rmin:=rmin/1.1{.99}; rmax:=rmax+0.001;
for i:=1 to n do
begin
rlow[i]:=rmin;
if rin[i]<rmin then rlow[i]:=rin[i]/1.1;
rhigh[i]:=rmax+1e-4;
if r[i]>rhigh[i] then rhigh[i]:=r[i]+1e-4;
if lrhhh[i] then rhigh[i]:=r[i]+1e-5;
end;
for i:=1 to n-1 do
begin
{ if (1.3333*3.1416*(r[mprin[i]]*sqr(r[mprin[i]])-r[mprin[i+1]]*sqr(r[mprin[i+1]]))>((lengkon-
leng)*widx*widy/10))and(r[mprin[i+1]]>rin[mprin[i+1]]) then
begin
rhigh[mprin[i+1]]:=r[mprin[i+1]]+1e-4;
end}
end;
end;
repeat

```

```

assignfile(fff,'pci.dat');
rewrite(fff);
//writeln(fff,a0,' ',leng,' 0.5 0.5 0.05 2 100');x y z r h R
if tofp=1 then writeln(fff,' 0 0 0 ',widx,' ',widy,' ',leng,' 0.5 0.5 0.5 0.5 0.5 0.5 0 3 0 0 0 ', n:3);
//if tofp=1 then writeln(fff,' 0 0 0 ',widx,' ',widy,' ',widz,' 0.5 0.5 0.5 0.5 0.5 0.5 0 0 1 0 0 ', n:3);
//if tofp=2 then writeln(fff,' 0 0 0 ',widx,' ',widy,' ',widz,' 0.5 0.5 0.5 0.5 0.5 0.5 2 4 1 0 0 ', n:3);
//if tofp=2 then writeln(fff,-widx,' ',-widy,' ',-leng,' ',widx,' ',widy,' ',leng,' 0.5 0.5 0.5 0.5 0.5 0.5 2 3 0 0 0 ', n:3);
if tofp=2 then writeln(fff,-leng,' ',-leng,' ',-leng,' ',leng,' ',leng,' ',leng,' 0.5 0.5 0.5 0.5 0.5 0.5 2 3 0 0 0 ', n:3);
//writeln(fff,' 0 0 0 ',widx,' ',widy,' ',leng,' 0.5 0.5 0.5 0.5 0.5 0.5 2 3 0 0 0 ', n:3);
if tofp=3 then writeln(fff,' 0 0 0 ',widx,' ',widy,' ',leng,' 0.5 0.5 0.5 0.5 0.5 0.5 1 3 0 0 0 ', n:3);
for i:=1 to n do
writeln(fff,x[i],' ',y[i],' ',z[i],' ',rhigh[i],' ',r[i],' ',rlow[i]);
closefile(fff);
//showall;
//ExecAndWait( 'IPOPT_Circles.exe', '', SW_SHOWNORMAL);
zestbeg:=zest;
for i:=1 to n do
begin
xbeg[i]:=x[i]; ybeg[i]:=y[i]; zbeg[i]:=z[i]; rbeg[i]:=r[i];
end;
//ip;
optimise;
{ SetParameter(1,leng);
SetParameter(2,a0);
SetParameter(3,0.5);SetParameter(4,0.5);SetParameter(5,0.5);
SetParameter(6,1e-2);
SetParameter(7,2); SetParameter(8,0);SetParameter(9,0);
ifull:=0;
for i:=1 to n do
begin
ifull:=ifull+1;
xfull[ifull]:=x[i];
ifull:=ifull+1;
xfull[ifull]:=y[i];
ifull:=ifull+1;
xfull[ifull]:=r[i];
end;
//showall;
nfull:=ifull;
p_rmaxx:=@rhigh[1];p_rminn:=@rlow[1];
p_xfull:=@xfull;
code:=RunOptimisation(n,p_xfull,p_rminn,p_rmaxx);
ifull:=0;i:=1;
repeat
ifull:=ifull+1;
x[i]:=xfull[ifull];
ifull:=ifull+1;
y[i]:=xfull[ifull];
ifull:=ifull+1;
r[i]:=xfull[ifull];
i:=i+1;
until ifull=nfull;}
assignfile(fff,'output.txt');
reset(fff);
readln(fff);readln(fff);readln(fff);
for i:=1 to n do
readln(fff,x[i],y[i],z[i],r[i]);
closefile(fff);
convio(nofvio,iv,jv,dis,2e-7);
zest:=0;
for i:=1 to n do
zest:=zest+sqr(r[i])*r[i];
//showall;

```

```

{if zest>zestiter then zestiter:=zest;
if iter mod 5 =0 then
begin
if (abs(zestiter-zbegiter)<0.001) then break;
zbegiter:=zestiter;
zestiter:=0;
end;
}
until (abs(zest-zestbeg)<tochn)or(zest<zestbeg);
convio(nofvio,iv,jv,dis,2e-7);
if (nofvio<>0)and(dis>2e-7) then
convio(nofvio,iv,jv,dis,2e-7);
if zest<zestbeg then
for i:=1 to n do
begin
x[i]:=xbeg[i]; y[i]:=ybeg[i]; z[i]:=zbeg[i]; r[i]:=rbeg[i];
end;
//showall;
end;
procedure place_n;
begin
for i:=1 to n do
begin
r[i]:=0.01;
repeat
if random(2)=0 then
x[i]:=(leng)*random(10000)/10000 else x[i]:=-leng)*random(10000)/10000;
if random(2)=0 then
y[i]:=(leng)*random(10000)/10000 else y[i]:=-leng)*random(10000)/10000;
if random(2)=0 then
z[i]:=(leng)*random(10000)/10000 else z[i]:=-leng)*random(10000)/10000;
until (sqr(x[i]-xsph)+sqr(y[i]-ysph)+sqr(z[i]-zsph)<sqr(leng-rin[i]));
end;
convio(nofvio,iv,jv,dis,2e-7);
formact(rmax);
dutlin;
end;
procedure grafi;
begin
assignfile(fgr,'sph.sph');
rewrite(fgr);
writeln(fgr,n:3);
writeln(fgr,'0 0 0');
for i:=1 to n do
writeln(fgr,'r[' ,i:2,']= ',r[i]:8:4,' x[' ,i:2,']= ',x[i]:8:4,' y[' ,i:2,']= ',y[i]:8:4,' z[' ,i:2,']= ',z[i]:8:4);
closefile(fgr);
(*closegraph;*)
procedure TForm1.Button1Click(Sender: TObject);
label
prokhod2,endtree,endbranch,y6,nextj,endix,vne,endi,mloc,l2,l3,snov,novprob,proddut,nflo,sdut,df1,mprob,parami,hlh,e
nddut,lidut1,lidut,parami1,novt;
begin
ztreebest:=0.48; ntrunc:=0; dcub:=0.5;
for i:=1 to 10 do
zest10[i]:=0;
//randomize;
assignfile(fff,'pbestgstat.dat');
rewrite(fff);
closefile(fff);
MyPaintBox:=Form1.Paintbox1;
lengbestg:=1e+37;

```

```

iglcoun:=0;
zest1min:=1e+37;zest2min:=1e+37;
zest4min:=1e+37;zest5min:=1e+37;
lengmin:=1e+37;
{
assignfile(fff,'p.txt');
reset(fff);
for i:=1 to 33 do
readln(fff,r[i],x[i],y[i],z[i]);
closefile(fff);
assignfile(fff,'p33.sph');
rewrite(fff);
for i:=1 to 33 do
  writeln(fff,r[i],i:3,']= ',r[i]:7:4,' x[' ,i:3,']= ',x[i]:7:4,' y[' ,i:3,']= ',y[i]:7:4,' z[' ,i:3,']= ',z[i]:7:4);
closefile(fff);
}
novt;;
iglcoun:=iglcoun+1;
icou:=0;
lparami:=false;
assignfile(fff,'dd.dat');assignfile(fff1,'dd3.dat');
reset(fff);
readln(fff,ntype);
// nko:=0.10;
//for i:=1 to n do
//rin[i]:=0.01;
//readln(fff,tofp);
tofp:=2;
if tofp=2 then
begin
readln(fff,leng);
r[0]:=leng;
xsph:=0; ysph:=0; zsph:=0;
widx:=leng; widy:=leng; widz:=leng;
end;
{dact:=leng/10;}
for i:=1 to ntype do
readln(fff,rtype[i],nmaxt[i]);
{считывание данных о цилиндре}
readln(fff,rcylq); {радиус цилиндра}
readln(fff,s0xq,s0yq,s0zq); {точка на оси}
readln(fff,sxq,syq,szq); {вектор на оси}
{построение уравнения цилиндра}
modsq:=sqrt(sqr(sxq)+sqr(syq)+sqr(szq));
cosalfq:=sxq/modsq; cosbetq:=syq/modsq; cosgamq:=szq/modsq;
cosalb:=sqrt(sqr(cosalfq)+sqr(cosbetq));
costetq:=cosalfq/cosalb; sintetq:=cosbetq/cosalb; singamq:=sqrt(1-sqr(cosgamq));
aq:=cosgamq*costetq; bq:=cosgamq*sintetq; {вспомогательные числа для неравенства}
{end построение уравнения цилиндра}
{определение нижней гарантированной оценки kolichestva krugov i-go tipa}
for itype:=1 to ntype do
begin
n:=trunc(0.33*sqr(r[0])*r[0]/sqr(rtype[itype])/rtype[itype]);
for j:=1 to n do
begin
rin[j]:=rtype[itype]+2e-7;
rin2[j]:=rin[j];
end;
{place_n;}
end;
//ztreebest:=0;
{constructing tree}
npokh:=1;

```

```

prokhod2;;
ntrunc:=0;
for i:=1 to ntype do
tree[i]:=0;
tree[ntype]:=-1;
while (true) do
begin
utr:=ntype+1;
repeat
utr:=utr-1;
if utr=0 then {end of tree}
begin
if nprokh=4 then goto endtree;
nprokh:=nprokh+1;
goto prokhod2;
end;
tree[utr]:=tree[utr]+1;
until tree[utr]<=nmaxt[utr];
for jtr:=utr+1 to ntype do
tree[jtr]:=0;
{tree[ntype]:=-1;}
utr:=utr;
{realise a tree branch}
n:=0;
for i:=1 to ntype do
for j:=1 to tree[i] do
begin
n:=n+1;
rin[n]:=rtype[i];
rin[n]:=rin[n]+2e-7;
end;
if n=0 then goto endbranch;
sqrsum:=0;
for i:=1 to n do
sqrsum:=sqrsum+sqr(rin[i])*rin[i];
if (sqrsum/(sqr(r[0])*r[0]))<ztreebest then goto endbranch;
if (nprokh>=2)and((sqrsum/(sqr(r[0])*r[0])-ztreebest)>0.02) then goto endbranch;
if nprokh=1 then
for i := 1 to ntrunc do
begin
ltrunc:=true;
for j := 1 to ntype do
if tree[j]<treetrunc[i,j] then ltrunc:=false;
if ltrunc then goto endbranch;
end;
widz:=leng;
lengkon:=leng;
{for i:=1 to n do
begin
read(fff,rin[i]);
rin[i]:=rin[i]+2e-7;
end;}
for i:=1 to n do
rin2[i]:=rin[i];
{for i:=1 to n do
rin[i]:=rin[i]+tochn/10;}
nbeg:=n;
n:=n;
nbegin:=n;
rmax:=0;rmin:=1e+37;
a0:=widz; b0:=widy;
{a0:=a0*0.6; }
a01:=a0;

```

```

for i:=1 to n do
zest:=leng; {sumbegin:=0;}
for i:=1 to n do
  begin
    {read(fff,rin[i]);}
    r[i]:=rin[i];
    rin1[i]:=rin[i];
    mp[i]:=i; mprin[i]:=i;
  { sumbegin:=sumbegin+(r[i]);}
  end;
for j:=n-1 downto 1 do
for i:=1 to j do
  begin
    if rin[mprin[i]]<rin[mprin[i+1]] then
      begin
        irab:=mprin[i]; mprin[i]:=mprin[i+1]; mprin[i+1]:=irab;
      end;
    end;
for i:=1 to n-1 do
rp11[i]:=rin[mprin[i+1]]-1e-2;
rp11[n]:=rin[mprin[n]];
for i:=n downto 2 do
rphh[i]:=rin[mprin[i-1]]+1e-2;
rphh[1]:=rin[mprin[1]];
sqrdel:=0; sqrmin:=1e+37;
for i:=1 to n-1 do
  begin
    if sqrdel<sqrt(rin[mprin[i]]*rin[mprin[i]]-sqrt(rin[mprin[i+1]])*rin[mprin[i+1]]) then
      sqrdel:=sqrt(rin[mprin[i]]*rin[mprin[i]]-sqrt(rin[mprin[i+1]])*rin[mprin[i+1]]);
    if sqrmin>sqrt(rin[mprin[i]]*rin[mprin[i]]-sqrt(rin[mprin[i+1]])*rin[mprin[i+1]]) then
      sqrmin:=sqrt(rin[mprin[i]]*rin[mprin[i]]-sqrt(rin[mprin[i+1]])*rin[mprin[i+1]]);
    end;
    sqrdelc:=sqrdel/0.99;
    sumbegin:=1.1*sumbegin;
    (*closefile(fff);
    assignfile(fff,'ploc.dat');
    {random(1);}
    assignfile(flo,'plo.dat');*)
    //randomize;
    random(1);
    y6;;
  for i:=1 to n do
  begin
    r[i]:=0.5;
    repeat
    if random(2)=0 then
    x[i]:=(leng)*random(10000)/10000 else x[i]:=-(leng)*random(10000)/10000;
    if random(2)=0 then
    y[i]:=(leng)*random(10000)/10000 else y[i]:=-(leng)*random(10000)/10000;
    if random(2)=0 then
    z[i]:=(leng)*random(10000)/10000 else z[i]:=-(leng)*random(10000)/10000;
    ll4:=true;
    ll1:=(sqrt(x[i]-xsph)+sqrt(y[i]-ysph)+sqrt(z[i]-zsph)<sqrt(leng-r[i]));
    {непересечение с зоной запрета - цилиндром}
    ll2:=sqrt(sqrt((x[i]-s0xq)*cosgamq*costetq+(y[i]-s0yq)*cosgamq*sintetq-(z[i]-s0zq)*singamq)+
      sqrt((x[i]-s0xq)*sintetq-(y[i]-s0yq)*costetq))-(rcylq+r[i])>0;
    ll3:=psi2>0;
    until (ll1)and(ll2)and(ll3)and(ll4);
    end;
    zestmin:=1e+37;
  nflo;;
  for k:=1 to n do
  begin

```

```

    xbest[k]:=x[k]; ybest[k]:=y[k]; zbest[k]:=z[k]; rbest[k]:=r[k];
    end;
convio(nofvio,iv,jv,dis,2e-7);
if (nofvio<>0)and(dis>2e-7) then
convio(nofvio,iv,jv,dis,2e-7);
if {eof(flo)}false then
begin
convio(nofvio,iv,jv,dis,2e-7);
for k:=1 to n do
    begin
        x[k]:=xbest[k]; y[k]:=ybest[k]; z[k]:=zbest[k]; r[k]:=rbest[k];
    end;
{exit;}
end;
{leng:=objfun;}
lleng:=false; lprob:=false;
{vognat(1);}
kappa:=0;
lengbestg:=lengkon;
{reset(fff1);}
readln(fff1,n,nprobp,delpar);
readln(fff1,wid,leng);
closefile(fff1);}
lengkon:=leng;
repeat
nbros:=0; ltr:=0; zestbro:=0;
repeat
nbros:=nbros+1;
{for i:=1 to n do
begin
r[i]:=0.01;
repeat
if random(2)=0 then
x[i]:=(leng)*random(10000)/10000 else x[i]:=-(leng)*random(10000)/10000;
if random(2)=0 then
y[i]:=(leng)*random(10000)/10000 else y[i]:=-(leng)*random(10000)/10000;
if random(2)=0 then
z[i]:=(leng)*random(10000)/10000 else z[i]:=-(leng)*random(10000)/10000;
until (sqr(x[i]-xsph)+sqr(y[i]-ysph)+sqr(z[i]-zsph)<sqr(leng-rin[i]));
end;
convio(nofvio,iv,jv,dis,2e-7);}
{
formact(rmax);
dutlin;}
repeat {local maximum}
grafi;
convio(nofvio,iv,jv,dis,2e-7);
for i:=1 to n do
    begin
        xmem[i]:=x[i]; ymem[i]:=y[i]; zmem[i]:=z[i]; rmem[i]:=r[i];
    end;
zest:=0;
for i:=1 to n do
zest:=zest+r[i]*r[i]*r[i];
zest:=(zest/(sqr(r[0])*r[0]));
zestmem:=zest;
assignfile(fff,'Polysystem.dat');
rewrite(fff);
for i:=1 to n do
    begin
        writeln(fff,sqr(leng):9:6,'-',2*leng:9:6,['',n*3+i:1,'']+['',n*3+i:1,']^2-['',i*3-2:1,']^2-['',i*3-1:1,']^2-['',i*3:1,']^2');
        writeln(fff,rin[i]:9:7,'-',['',n*3+i:1,']);
    end;
end;

```

```

for i:=1 to n do
{ if (sqrt(sqrt((y[i]-s0yq)*szq-syq*(z[i]-s0zq))+sqrt((x[i]-s0xq)*szq-sxq*(z[i]-s0zq))+sqrt((x[i]-s0xq)*syq-sxq*(y[i]-s0yq)))/sqrt(sqrt(sxq)+sqrt(syq)+sqrt(szq)))>0 then}
begin
if (sqrt(sqrt((x[i]-s0xq)*cosgamq*costetq+(y[i]-s0yq)*cosgamq*sintetq-(z[i]-s0zq)*singamq)+sqrt((x[i]-s0xq)*sintetq-(y[i]-s0yq)*costetq))<(rcylq+r[i]+(sqrt(3)+1)*dcub)) then
begin {неравенство для непересечения с цилиндром}
rab:=sqrt(aq);
write(fff,rab:9:6,['i',i*3-2:1,']^2');
rab:=rab*(-2)*s0xq; if rab>=0 then write(fff,'+');
write(fff,rab:9:6,['i',i*3-2:1,']');
rab:=sqrt(bq); if rab>=0 then write(fff,'+');
write(fff,rab:9:6,['i',i*3-1:1,']^2');
rab:=rab*(-2)*s0yq; if rab>=0 then write(fff,'+');
write(fff,rab:9:6,['i',i*3-1:1,']');
rab:=sqrt(singamq); write(fff,'+');
write(fff,rab:9:6,['i',i*3:1,']^2');
rab:=rab*(-2)*s0zq; if rab>=0 then write(fff,'+');
write(fff,rab:9:6,['i',i*3:1,']');
rab:={-}*2*aq*bq; if rab>=0 then write(fff,'+');
write(fff,rab:9:6,['i',i*3-2:1,'] ['i',i*3-1:1,']');
rab:=-rab*s0yq; if rab>=0 then write(fff,'+');
write(fff,rab:9:6,['i',i*3-2:1,']');
rab:={+}*2*aq*bq*s0xq; if rab>=0 then write(fff,'+');
write(fff,rab:9:6,['i',i*3-1:1,']');
rab:=-2*aq*singamq; if rab>=0 then write(fff,'+');
write(fff,rab:9:6,['i',i*3-2:1,'] ['i',i*3:1,']');
rab:=-rab*s0zq; if rab>=0 then write(fff,'+');
write(fff,rab:9:6,['i',i*3-2:1,']');
rab:=2*aq*singamq*s0xq; if rab>=0 then write(fff,'+');
write(fff,rab:9:6,['i',i*3:1,']');
rab:={+}*2*bq*singamq; if rab>=0 then write(fff,'+');
write(fff,rab:9:6,['i',i*3-1:1,'] ['i',i*3:1,']');
rab:=-rab*s0zq; if rab>=0 then write(fff,'+');
write(fff,rab:9:6,['i',i*3-1:1,']');
rab:={-}*2*bq*singamq*s0yq; if rab>=0 then write(fff,'+');
write(fff,rab:9:6,['i',i*3:1,']');
rab:=sqrt(sintetq); write(fff,'+');
write(fff,rab:9:6,['i',i*3-2:1,']^2');
rab:=-2*sqrt(sintetq)*s0xq; if rab>=0 then write(fff,'+');
write(fff,rab:9:6,['i',i*3-2:1,']');
rab:=sqrt(costetq); if rab>=0 then write(fff,'+');
write(fff,rab:9:6,['i',i*3-1:1,']^2');
rab:=-2*sqrt(costetq)*s0yq; if rab>=0 then write(fff,'+');
write(fff,rab:9:6,['i',i*3-1:1,']');
rab:=-2*sintetq*costetq; if rab>=0 then write(fff,'+');
write(fff,rab:9:6,['i',i*3-2:1,'] ['i',i*3-1:1,']');
rab:=2*sintetq*costetq*s0yq; if rab>=0 then write(fff,'+');
write(fff,rab:9:6,['i',i*3-2:1,']');
rab:=2*sintetq*costetq*s0xq; if rab>=0 then write(fff,'+');
write(fff,rab:9:6,['i',i*3-1:1,']');
rab:=-2*rcylq; if rab>=0 then write(fff,'+');
write(fff,rab:9:6,['i',n*3+i:1,']-['i',n*3+i:1,']^2');
rab:=sqrt(aq)*sqrt(s0xq)+sqrt(bq)*sqrt(s0yq)+sqrt(singamq)*sqrt(s0zq){-}+2*aq*bq*s0xq*s0yq-
2*aq*singamq*s0xq*s0zq{+}-2*bq*singamq*s0yq*s0zq+
sqrt(sintetq)*sqrt(s0xq)+sqrt(costetq)*sqrt(s0yq)-2*sintetq*costetq*s0xq*s0yq-sqr(rcylq);
if rab>=0 then write(fff,'+');
writeln(fff,rab:9:6);
{end неравенство для непересечения с цилиндром}
end;
end {i};
for i:=1 to n-1 do

```

```

begin
for j:=i+1 to n do
{if sqrt(sqrt(x[i]-x[j])+sqrt(y[i]-y[j])+sqrt(z[i]-z[j]))<(r[i]+r[j]+(sqrt(3)+2)*dcub) then}
begin
write(fff,['i*3-2:1,']^2-2['i*3-2:1,']['j*3-2:1,']+['j*3-2:1,']^2+',
['i*3-1:1,']^2-2['i*3-1:1,']['j*3-1:1,']+['j*3-1:1,']^2+');
write(fff,['i*3:1,']^2-2['i*3:1,']['j*3:1,']+['j*3:1,']^2-);
writeln(fff,['n*3+i:1,']^2-2['n*3+i:1,']['n*3+j:1,']-['n*3+j:1,']^2');
{вспомогательное}
write(fff,['i*3-2:1,']^2-2['i*3-2:1,']['j*3-2:1,']+['j*3-2:1,']^2+',
['i*3-1:1,']^2-2['i*3-1:1,']['j*3-1:1,']+['j*3-1:1,']^2+');
write(fff,['i*3:1,']^2-2['i*3:1,']['j*3:1,']+['j*3:1,']^2-);
writeln(fff,['n*3+i:1,']^2-2['n*3+i:1,']['n*3+j:1,']-['n*3+j:1,']^2');
} closefile(fff); }
end;{j}
end;{i}
{дополнительные ограничения на переменные}
for i:=1 to n do
begin
if r[i]+dcub<0 then writeln(fff,['n*3+i:1,'],r[i]+dcub:9:6) else writeln(fff,['n*3+i:1,'],r[i]+dcub:9:6);
if x[i]+dcub<0 then writeln(fff,['i*3-2:1,'],x[i]+dcub:9:6) else writeln(fff,['i*3-2:1,'],x[i]+dcub:9:6);
if -x[i]+dcub<0 then writeln(fff,['i*3-2:1,'],-x[i]+dcub:9:6) else writeln(fff,['i*3-2:1,'],-x[i]+dcub:9:6);
if -y[i]+dcub<0 then writeln(fff,['i*3-1:1,'],-y[i]+dcub:9:6) else writeln(fff,['i*3-1:1,'],-y[i]+dcub:9:6);
if y[i]+dcub<0 then writeln(fff,['i*3-1:1,'],y[i]+dcub:9:6) else writeln(fff,['i*3-1:1,'],y[i]+dcub:9:6);
if -z[i]+dcub<0 then writeln(fff,['i*3:1,'],-z[i]+dcub:9:6) else writeln(fff,['i*3:1,'],-z[i]+dcub:9:6);
if z[i]+dcub<0 then writeln(fff,['i*3:1,'],z[i]+dcub:9:6) else writeln(fff,['i*3:1,'],z[i]+dcub:9:6);
end;
{конец дополнительных ограничений на переменные}
for i:=1 to n do
writeln(fff,['i*3-2:1,']:='x[i]:9:6;', ['i*3-1:1,']:='y[i]:9:6;', ['i*3:1,']:='z[i]:9:6');
for i:=1 to n do
{ if lc[i] then } writeln(fff,['3*n+i:1,']:='rin[i]:9:7');
write(fff,'@-');
for i:=1 to n do
write(fff,['n*3+i:1,']-);
writeln(fff,'0');
writeln(fff,['0']=0');
(*****)
convio(nofvio,iv,jv,dis,2e-7);
if (nofvio<>0)and(dis>2e-7) then
convio(nofvio,iv,jv,dis,2e-7);
n:=nbegin;
for i:=1 to n do
begin
xbdut[i]:=x[i]; ybdut[i]:=y[i]; zbdut[i]:=z[i]; rbdut[i]:=r[i];
end;
if lengkon-leng<1e-2 then goto parami;
convio(nofvio,iv,jv,dis,2e-7);
if (nofvio<>0)and(dis>2e-7) then
convio(nofvio,iv,jv,dis,2e-7);
nbegin:=nbegin;
i_vstavka;
n:=nbegin;
for i:=1 to n do
begin
xbdut[i]:=x[i]; ybdut[i]:=y[i]; zbdut[i]:=z[i]; rbdut[i]:=r[i];
end;
convio(nofvio,iv,jv,dis,2e-7);
if (nofvio<>0)and(dis>2e-7) then
convio(nofvio,iv,jv,dis,2e-7);
goto l2;
end;
for i:=1 to n do lr[i]:=true; lleng:=false;

```

```

lprob:=true;
for isn:=1 to 1 do
begin
  for i:=1 to n do
  begin
    lrhhh[i]:=false;
    lrlll[i]:=false;
  end;
  convio(nofvio,iv,jv,dis,2e-7);
  if (nofvio<>0)and(dis>2e-7) then
  convio(nofvio,iv,jv,dis,2e-7);
  for i:=1 to n do mpr[i]:=i;
  for j:=n-1 downto 1 do
  for i:=1 to j do
  begin
    if r[mpr[i]]<r[mpr[i+1]] then
    begin
      irab:=mpr[i]; mpr[i]:=mpr[i+1]; mpr[i+1]:=irab;
    end;
  end;
  for i:=1 to n do
  begin
    xmemdel[mprin[i]]:=x[mpr[i]];
    ymemdel[mprin[i]]:=y[mpr[i]];
    zmemdel[mprin[i]]:=z[mpr[i]];
    rmemdel[mprin[i]]:=r[mpr[i]];
  end;
  for i:=1 to n do
  begin
    r[i]:=rmemdel[i];x[i]:=xmemdel[i]; y[i]:=ymemdel[i]; z[i]:=zmemdel[i];
  end;
  for i:=1 to n do mpr[i]:=mprin[i];
  convio(nofvio,iv,jv,dis,2e-7);
  if (nofvio<>0)and(dis>2e-7) then
  convio(nofvio,iv,jv,dis,2e-7);
  //dut;
  dutunlin;
  convio(nofvio,iv,jv,dis,2e-7);
  if (nofvio<>0)and(dis>2e-7) then
  convio(nofvio,iv,jv,dis,2e-7);
  {for i:=1 to n do
  begin
    xbddd1[i]:=x[i]; ybddd1[i]:=y[i]; rbddd1[i]:=r[i];
  end;
  for i:=1 to n do
  if r[i]>rin2[i] then r[i]:=rin2[i];
  dutlin;
  if lkod then goto mloc;}
  {dutunlin1;}
  for i:=1 to n do
  if r[i]>rin[i] then r[i]:=rin[i];
  {for i:=1 to n do lr[i]:=true; lleng:=false;
  vognat;
  kappa:=0;
  dutlin;
  if lkod then
  begin
    for i:=1 to n do ;
    goto mloc;
  end;
  zest:=0;}
  if icou=1 then
  begin

```

```

leng:=lengkon;
formact(rmax);
dutlin;
lkod2:=lkod;
for i:=1 to n do lr[i]:=false; lleng:=true;
//dut;
davit;
{if zest>35.8 then goto novt;
if zest<35.4 then icou:=2;}
{if zest>32.7 then goto novt;
if zest<32.5 then icou:=2;
}
if zest<zest1min then zest1min:=zest;
if zest<zest2min then zest2min:=zest;
{ if zest>100 then goto novt;}
if (zest>zest1min*1.01)or(iglcount<10) then goto novt;
{if zest<99.7 then icou:=2;}
if zest<zest2min*1.01 then icou:=2;
goto mloc;
end;
if icou=2 then
begin
leng:=lengkon;
formact(rmax);
dutlin;
lkod2:=lkod;
for i:=1 to n do lr[i]:=false; lleng:=true;
//dut;
convio(nofvio,iv,jv,dis,2e-7);
if (nofvio<>0)and(dis>2e-7) then
convio(nofvio,iv,jv,dis,2e-7);
davit;
convio(nofvio,iv,jv,dis,2e-7);
if (nofvio<>0)and(dis>2e-7) then
convio(nofvio,iv,jv,dis,2e-7);
if zest<zest2min then zest2min:=zest;
{if zest>99.7 then goto novt;}
if zest>zest2min*1.02 then goto novt;
if lkod2 then goto mloc else goto novt;
end;
convio(nofvio,iv,jv,dis,2e-7);
if (nofvio<>0)and(dis>2e-7) then
convio(nofvio,iv,jv,dis,2e-7);
formact(rmax);
dutunlin;
convio(nofvio,iv,jv,dis,2e-7);
if (nofvio<>0)and(dis>2e-7) then
convio(nofvio,iv,jv,dis,2e-7);
zest:=0;
for i:=1 to n do
if r[i]>rin[i] then zest:=zest+sqr(rin[i])*(rin[i]) else
zest:=zest+sqr(r[i])*(r[i]);
zestddd:=zest;
for i:=1 to n do
begin
xbddd[i]:=x[i]; ybddd[i]:=y[i]; zbddd[i]:=z[i]; rbddd[i]:=r[i];
end;
for k:=1 to n do
lrhhddd[k]:=lrhhh[k];
ilid:=0;
while (true) do
begin
for idut:=2 to n do

```

```

begin
for i:=1 to n do
  begin
    x[i]:=xbddd[i]; y[i]:=ybddd[i]; z[i]:=zbddd[i]; r[i]:=rbddd[i];
  end;
{for k:=1 to n do
lrhhh[k]:=lrhhhddd[k];
}
for jdut:=1 to n do lrhhh1[jdut]:=true;
lidut1::
  begin
    delmax:=0; idelmax:=0;
    for jdut:=1 to n-1 do
      if not(lrhhh[mprin[jdut+1]])and(lrhhh1[mprin[jdut+1]]) then
        begin
          {lrhhh[mprin[jdut]]:=true;}
          if (r[mprin[jdut+1]]>rin[mprin[jdut+1]])and
            ({sqr} 2*(r[mprin[jdut+1]]-{sqr}r{in}[mprin[jdut]])+(r[mprin[jdut+1]]-
{sqr}rin[mprin[jdut+1]])>delmax)and(random(3)>0) then
            begin
              idelmax:=mprin[jdut+1];
(* delmax:={sqr}(r[mprin[jdut+1]]-{sqr}(r{in}[mprin[jdut]]);*)
delmax:=2*(r[mprin[jdut+1]]-{sqr}r{in}[mprin[jdut]])+(r[mprin[jdut+1]]-{sqr}rin[mprin[jdut+1]]);
              end;
(* if (r[mprin[jdut+1]]>rin[mprin[jdut+1]])and
            ({sqr}(r[mprin[jdut+1]]-{sqr}(rin[mprin[jdut+1]]>delmax)and(random(3)=1) then
              begin
                idelmax:=mprin[jdut+1];
                delmax:={sqr}(r[mprin[jdut+1]]-{sqr}(rin[mprin[jdut+1]]);
              end;*)
            end;
if ilid>n*2 then idelmax:=0;
if idelmax=0 then
  goto enddut;
r[idelmax]:=rin[idelmax];
lrhhh[idelmax]:=true;
{ dut;}
formact(rmax);
dutunlin;
{
for i:=1 to n do
  begin
    xbddd1[i]:=x[i]; ybddd1[i]:=y[i]; rbddd1[i]:=r[i];
  end;
for i:=1 to n do
  if r[i]>rin2[i] then r[i]:=rin2[i];
dutlin;
if lkod then goto mloc;}
zest:=0;
for i:=1 to n do
  if r[i]>rin[i] then zest:=zest+sqr(rin[i])*rin[i] else
    zest:=zest+sqr(r[i])*r[i];
  if zest>zestddd then
    begin
      convio(nofvio,iv,jv,dis,2e-7);
      zestddd:=zest;
    end;
for i:=1 to n do
  begin
    xbddd[i]:=x[i]; ybddd[i]:=y[i]; zbddd[i]:=z[i]; rbddd[i]:=r[i];
  end;
for k:=1 to n do
  lrhhhddd[k]:=lrhhh[k];

```

```

goto lidut;
end else
begin
for i:=1 to n do
begin
x[i]:=xbddd[i]; y[i]:=ybddd[i]; z[i]:=zbddd[i]; r[i]:=rbddd[i];
end;
//MyPaintBox.Refresh;
//patt(b0,a0);
//for i:=1 to n do
//begin
//showcir(x[i],y[i],r[i],rin[i],i,i,1);
//end;
//patt(b0,a0);
ilid:=ilid+1;
lrhhh1[idelmax]:=false;lrhhh[idelmax]:=false; goto lidut1;
end;
{ for i:=1 to n do
begin
lrhhh[i]:=false;
lrlll[i]:=false;
end;}
end;
for i:=1 to n do
begin
xmemdel[mprin[i]]:=x[mpr[i]];
ymemdel[mprin[i]]:=y[mpr[i]];
zmemdel[mprin[i]]:=z[mpr[i]];
if r[mpr[i]]>rin[mprin[i]] then
rmemdel[mprin[i]]:=rin[mprin[i]] else rmemdel[mprin[i]]:=r[mpr[i]];
end;
for i:=1 to n do
begin
r[i]:=rmemdel[i]; x[i]:=xmemdel[i]; y[i]:=ymemdel[i]; z[i]:=zmemdel[i];
end;
proddut;
{for i:=1 to n do lr[i]:=true; lleng:=false;}
lprob:=false;
{dut;}
formact(rmax);
dutlin;
if lkod then
begin
for i:=1 to n do ;
goto mloc;
end;
zest:=0;
for i:=1 to n do
zest:=zest+sqr(r[i])*r[i];
if zest>zestbloc+1e-3 then
begin
zestbloc:=zest;
nprob:=0;
n:=n;
{patt(a0,b0);}
//patt(b0,a0);
//for i:=1 to n do
//begin
//showcir(x[i],y[i],r[i],rin[i],i,i,1);
//end;
//patt(b0,a0);
nbegin:=nbeg;
i_vstavka;

```

```

n:=nbegin;
zestbdut:=zest;
for i:=1 to n do
  begin
    xbdut[i]:=x[i]; ybdut[i]:=y[i]; zbdut[i]:=z[i]; rbdut[i]:=r[i];
  end;
convio(nofvio,iv,jv,dis,2e-7);
if (nofvio<>0)and(dis>2e-7) then
convio(nofvio,iv,jv,dis,2e-7);
{goto proddut;}
end;
until lkod;
parami:=0;
convio(nofvio,iv,jv,dis,2e-7);
if (nofvio<>0)and(dis>2e-7) then
convio(nofvio,iv,jv,dis,2e-7);
if lengkon<lengmin then lengmin:=lengkon;
if lengkon>lengmin*1.003 then goto novt;
sqrin:=0;
for i:=1 to n do
sqrin:=sqrin+sqr(rin[i])*rin[i];
nparami:=0;
for ipa:=2 to n do
for jpa:=ipa-1 downto 1 do
if (-sqr(rin[mprin[ipa]])*rin[mprin[ipa]]+sqr(rin[mprin[jpa]])*rin[mprin[jpa]]<delpar)and (abs(rin[mprin[ipa]]-
rin[mprin[jpa]]))>1e-3) then
begin
nparami:=nparami+1;
ipar[nparami]:=ipa; jpar[nparami]:=jpa; rpar[nparami]:=0;
end;
parami1:=0;
lparami:=true;
leng:=lengkon{-1e-4};
for i:=1 to n do
begin
xbdut[i]:=xbest[i]; ybdut[i]:=ybest[i]; zbdut[i]:=zbest[i]; rbdut[i]:=rbest[i];
end;
for i:=1 to n do lr[i]:=false; lleng:=true;
for k:=1 to n do
begin
x[k]:=xbdut[k]; y[k]:=ybdut[k]; z[k]:=zbdut[k]; r[k]:=rbdut[k];
end;
for i:=1 to n do
begin
xbdut[i]:=x[i]; ybdut[i]:=y[i]; zbdut[i]:=z[i]; r[i]:=rbest[i];
end;
for k:=1 to n do
begin
x[k]:=xbdut[k]; y[k]:=ybdut[k]; z[k]:=zbdut[k]; r[k]:=rbdut[k];
end;
for k:=1 to n do
begin
xbdut[k]:=x[k]; ybdut[k]:=y[k]; zbdut[k]:=z[k]; rbdut[k]:=r[k];
end;
n:=nbegin;
for iparami:=1 to nparami do
begin
begin
ipa:=ipar[iparami]; jpa:=jpar[iparami];
begin
for k:=1 to n do
begin
x[k]:=xbdut[k]; y[k]:=ybdut[k]; z[k]:=zbdut[k]; r[k]:=rbdut[k];

```

```

end;
for i:=1 to n do lr[i]:=false; lleng:=true;
lr[mprin[ipa]]:=true; lr[mprin[jpa]]:=true;
{if (dt[1]<0) and (abs(dt[1])>1e-3) then}
begin
  xrab:=x[mprin[ipa]]; yrab:=y[mprin[ipa]]; zrab:=z[mprin[ipa]]; rrab:=r[mprin[ipa]];
  x[mprin[ipa]]:=x[mprin[jpa]]; y[mprin[ipa]]:=y[mprin[jpa]]; z[mprin[ipa]]:=z[mprin[jpa]];
  x[mprin[jpa]]:=xrab; y[mprin[jpa]]:=yrab; z[mprin[jpa]]:=zrab; r[mprin[jpa]]:=rrab;
  for i:=1 to n do lr[i]:=true; lleng:=false;
  convio(nofvio,iv,jv,dis,2e-7);
  formact(rmax);
  dutlin;
  convio(nofvio,iv,jv,dis,2e-7);
sqr:=0;
for i:=1 to n do
sqr:=sqr+sqr(r[i])*r[i];
if lkod then rpar[iparami]:=-1 else
rpar[iparami]:=sqrin-sqr;
{for i:=nparami-1 downto 1 do
for j:=1 to i do
if rpar[j]>rpar[j+1] then
begin
  irab:=ipar[j]; ipar[j]:=ipar[j+1]; ipar[j+1]:=irab;
  jrab:=jpar[j]; jpar[j]:=jpar[j+1]; jpar[j+1]:=jrab;
  rrab:=rpar[j]; rpar[j]:=rpar[j+1]; rpar[j+1]:=rrab;
end;}
  if lkod then
begin
for i:=1 to n do lr[i]:=false; lleng:=true;
//dut;
davit;
{dutlin;}
if leng+1e-5<lengkon then
begin
lengkon:=leng;
for k:=1 to n do
begin
  xbest[k]:=x[k]; ybest[k]:=y[k]; zbest[k]:=z[k]; rbest[k]:=r[k];
end;
convio(nofvio,iv,jv,dis,2e-7);
if nofvio<>0 then
convio(nofvio,iv,jv,dis,2e-7);
for i:=1 to n do
r[i]:=r[i]-2e-7;
convio(nofvio,iv,jv,dis,0);
for i:=1 to n do
r[i]:=r[i]+2e-7;
assignfile(fff,'pbest.dat');
rewrite(fff);
for i:=1 to n do
writeln(fff,x[i],',',y[i],',',z[i]);
writeln(fff,lengkon);
closefile(fff);
for i:=nparami-1 downto 1 do
for j:=1 to i do
if rpar[j]>rpar[j+1] then
begin
  irab:=ipar[j]; ipar[j]:=ipar[j+1]; ipar[j+1]:=irab;
  jrab:=jpar[j]; jpar[j]:=jpar[j+1]; jpar[j+1]:=jrab;
  rrab:=rpar[j]; rpar[j]:=rpar[j+1]; rpar[j+1]:=rrab;
end;
goto parami1;
end;

```

```

        end;
    end;
end;
end;
leng:=lengkon;
for i:=1 to n do
begin
x[i]:=xbest[i]; y[i]:=ybest[i]; z[i]:=zbest[i];
r[i]:=rin2[i];
end;
for i:=1 to n do lr[i]:=false; lleng:=true;
//dut;
convio(nofvio,iv,jv,dis,2e-7);
if nofvio<>0 then
convio(nofvio,iv,jv,dis,2e-7);
davit;
lengkon:=leng;
convio(nofvio,iv,jv,dis,2e-7);
for i:=1 to n do
r[i]:=r[i]-2e-7;
convio(nofvio,iv,jv,dis,0);
if nofvio<>0 then
convio(nofvio,iv,jv,dis,0);
for i:=1 to n do
r[i]:=r[i]+2e-7;
if nofvio=0 then
begin
assignfile(fff,'pbestgstat.dat');
append(fff);
for i:=1 to n do
writeln(fff,r[i]:5:3,' ',x[i],' ',y[i],' ',z[i]);
writeln(fff,lengkon);
closefile(fff);
end else
begin
for i:=1 to n do
r[i]:=r[i]-2e-7;
convio(nofvio,iv,jv,dis,0);
if nofvio<>0 then
convio(nofvio,iv,jv,dis,0);
for i:=1 to n do
r[i]:=r[i]+2e-7;
end;
if lengkon<lengbestg then
begin
convio(nofvio,iv,jv,dis,2e-7);
for i:=1 to n do
r[i]:=r[i]-2e-7;
convio(nofvio,iv,jv,dis,0);
if nofvio<>0 then
convio(nofvio,iv,jv,dis,0);
{for i:=1 to n do
r[i]:=r[i]+2e-7;}
if nofvio=0 then
begin
lengbestg:=lengkon;
assignfile(fff,'pbestg.dat');
rewrite(fff);
for i:=1 to n do
writeln(fff,r[i]:5:3,' ',x[i],' ',y[i],' ',z[i]);
writeln(fff,lengbestg);
closefile(fff);

```

```

end else
begin
for i:=1 to n do
r[i]:=r[i]-2e-7;
convio(nofvio,iv,jv,dis,0);
for i:=1 to n do
r[i]:=r[i]+2e-7;
end;
end;
{ goto novt;}
endbranch;;
end; {realise a tree branch}
endtree;;
assignfile(fff,'pbest10.dat');
rewrite(fff);
for k := 1 to 10 do
begin
n:=0;
for i:=1 to ntype do
n:=n+ttree10[k,i];
writeln(fff,n:3);
writeln(fff,r[0]);
for i:=1 to n do
begin
writeln(fff,rintree10[k,i],',',rtree10[k,i],',',xtree10[k,i],',',ytree10[k,i],',',ztree10[k,i]);
end;
for i:=1 to ntype do
write(fff,ttree10[k,i]:3,', ');
writeln(fff);writeln(fff);
end;
closefile(fff);
ShowMessage('Press Enter and Exit Button');
end;
procedure TForm1.Button2Click(Sender: TObject);
begin
Form1.close;
end;
end.

```

ВІДОМІСТЬ АТЕСТАЦІЙНОЇ РОБОТИ

Позначення	Найменування	Дод. відомості
	Текстові документи	
1	Пояснювальна записка	87с.
2	Презентаційний матеріал	20с.
	Інші документи	
3	Роздруківки програм	27с.
4	Рецензія	2с.
5	Відгук керівника	1с.

					Упакування куль у опуклому багатограннику із зонами заборони			
Змін	Арк.	Номер докум.	Підп.	Дата				
Розроб.	Рибачок І.А.				(Тема роботи) Відомість атестаційної роботи		Аркуш	Аркушів
Перевір.	Стоян Ю.Г.							
Н. контр.	Сидоров М.В.					ХНУРЕ		
Затв.	Гевяшев А.Д.					Кафедра ПМ		