```
// Real roots finding

Use R::=QQ[x];

F:=x^20 + x^19 - 8x^18 - 9x^17 + 21x^16 + 30x^15 - 37x^14 - 67x^13
+ 176x^12 + 243x^11 - 519x^10 - 762x^9 + 286x^8 + 1048x^7 +
1048x^6 - 1296x^4 - 1296x^3 - 432x^2 + 864x + 864;

RealRoots(F);
L:=RealRoots(F, 10^(-5));

L:=RealRoots(F);

L[1].Inf;
L[1].Sup;


DecimalStr(L[1].Inf);
DecimalStr(L[1].Sup);

DecimalStr(L[1].Inf,10);
DecimalStr(L[1].Sup,10);

DecimalStr(L[2].Inf,10);
DecimalStr(L[2].Sup,10);

G:=x^20 + x^19 - 8x^18 - 9x^17 + 21x^16 + 30x^15 - 37x^14 - 67x^13
+ 176x^12 + 243x^11 - 519x^10 - 762x^9 + 286x^8 + 1048x^7 +
1048x^6 - 1292x^4 - 1296x^3 - 432x^2 + 864x + 864;


L:=RealRoots(G,10^(-5));


DecimalStr(L[2].Inf,10);
DecimalStr(L[2].Sup,10);


/*
Returns linear form in the current vars, random coeffs in the range [-N,N]
Example
Use R::=QQ[x,y,z],Lex;
N:=100;
RandomForm(N);
*/
Define RandomForm(N)
```

```
  L:=[Rand(-N,N)|I In 1..NumIndets()];
  Return ScalarProduct(L,Indets());
EndDefine;



/////////////// Maps and pretty printing - Tagging

Use R ::= QQ[x,y,z], Lex;
S ::= QQ[x,y,z,t];

Phi:=RMap([RandomForm(100)|I In 1..NumIndets()]);

Phi;

Untagged(Phi);


Define Print_RMap(Phi)
  For I:=1 To Len(Phi) Do
    PrintLn I,"-th var -->", Phi[I];
  EndFor;
EndDefine;

Print_RMap(Phi);




Define Print_RMap(Phi)
  For I:=1 To Len(Phi) Do
    If I=1 Then
      PrintLn "First var  --> ", Phi[I];
    Elif I=2 Then
      PrintLn "Second var --> ", Phi[I];
    Elif I=3 Then
      PrintLn "Third var  --> ", Phi[I];
    Else
      PrintLn I,"-th var  --> ", Phi[I];
    EndIf;
  EndFor;
EndDefine;

Print_RMap(Phi);

A:=3;
```

```
B:=Tagged(3,"silly_tag");


Define Print_silly_tag(X)
  PrintLn X+1;
EndDefine;
B;
4


Untagged(B);
3


////////////////////////////////////////////////////////////


Use R ::= QQ[x,y,z,t], Lex;
I:=Ideal(y^2-xz, xz-yz,xt-z^2);

Phi:=RMap([RandomForm(2)|I In 1..NumIndets()]);
I1:=Image(I,Phi);
LT(I1);

Phi:=RMap([RandomForm(100)|I In 1..NumIndets()]);
I1:=Image(I,Phi);
LT(I1);


/*
Example
Use R ::= QQ[x,y,z,t], Lex;
I:=Ideal(y^2-xz+t^2, xz-yz+z^2,xt-z^2);
N:=100;
MyGin(I,N);
Ideal(x^2, xy, xz, y^3, xt^2, y^2z, y^2t^2, yz^3, yz^2t^2, yzt^4, z^7)
*/
Define MyGin(I,N)
  Phi:=RMap([RandomForm(N)|I In 1..NumIndets()]);
  Generic_I:=Image(I,Phi);
  OldGin:=LT(Generic_I);
  Tentatives:=0;
  While Tentatives<5 Do
    Phi:=RMap([RandomForm(N)|I In 1..NumIndets()]);
    Generic_I:=Image(I,Phi);
    MyGin:=LT(Generic_I);
    If MyGin<>OldGin Then
      OldGin:=MyGin;
      Tentatives:=0;
      PrintLn "MyGin Failure - trying again";
   Else
      Tentatives:=Tentatives+1;
    EndIf;
  EndWhile;
  Return MyGin;
EndDefine;

MyGin(I,2);

MyGin(I,5);

MyGin(I,1000000000);



K:=Gin(I,100);K;




// Use CoCoAServer
J:=Gin5(I,100);J;




  Use R ::= QQ[x,y,z], DegRevLex;
  Gin5(Ideal(y^2-xz, x^2z-yz^2));


-------------------------------
  Use R ::= QQ[x,y,z], Lex;
  Gin5(Ideal(y^2-xz, x^2z-yz^2), 10);  -- coeffs in [-10, 10]


-------------------------------
  Use R ::= QQ[x,y,z], Lex;
  Gin5(Ideal(y^2-xz, x^2z-yz^2), 2); -- coeffs in [-2,2], dangerously small:

-------------------------------
  Use R ::= QQ[x,y,z];
  Gin5(Ideal(y^2-xz, x^2z-yz^2), 100000);
```

```
Use R ::= QQ[x,y,z,t];

I:=Ideal(x^2, x^4y^3, xyz^2, xz^14, t^5y^6);
Max([Deg(P)|P In MinGens(Gin(I))]);
Reg(I);

Time Res(R/I);



Use R ::= QQ[x,y,z,t];

I:=Ideal(x^12, xy^2, xyz^2, xz^4, t^5y^6);
Max([Deg(P)|P In MinGens(Gin5(I))]);
Reg(I);

Time Res(R/I);

Use R ::= QQ[x,y,z,t],Lex;

I:=Ideal(x^12, xy^2, xyz^2, xz^4, t^5y^6);
Max([Deg(P)|P In MinGens(Gin5(I))]);
Reg(I);

//////////////////////////////////////////////////////
// Twin Float
Use R ::= QQ[x,y,z,t];
I:=Ideal(y^2-xz, xz-yz,xt-z^2);


J:=I+I^2;

I=J;
Gens(I)=Gens(J);



Use QQ[x[0..5]];
  L := [Randomized(DensePoly(4)) | I In 1..2];
  Time LT(Ideal(L));
  LT5x(Ideal(L), Record[FloatPrecision:=128]);
  Time  W:=ReducedGBasis(Ideal(L));
  W1:=ReducedGBasis5(Ideal(L));
  W2:=ReducedGBasis5x(Ideal(L), Record[FloatPrecision:=128]);


W=W1;
EqSet(W,W1);
EqSet(W,W2);




[LT(P)|P In W];

[LT(P)|P In W2];



P:=W[1];
P2:=W2[2];

L:=Coefficients(P);
L2:=Coefficients(P2);
FloatStr(L[2],40);
FloatStr(L2[2],40);


////////////////////////////// Operations with parameters
  // Binomial case

  M := $cocoa5.AddParamOrdMat(DegRevLexMat(4), 6); // compatible Term Ordering
  Use ZZ/(32003)[a[1..6], t,x,y,z], Ord(M);

  I := Ideal(a[1]t^11-a[2]x^11,a[3]t^5-a[4]y^5,a[5]t^7-a[6]z^7);
  ReducedGBasis5x(I, Record[NumParams:=6]);
  LT5x(I, Record[NumParams:=6]);
  LT5(I);
  LT5(Subst(LT5(I),[[Indet(Ind),Rand(-100,100)]|Ind In 1..6]));



  ///// Operations with parameters
  // Binomial case - more general case

  M := $cocoa5.AddParamOrdMat(DegRevLexMat(4), 9); // compatible Term Ordering
  Use ZZ/(32003)[a[1..9], t,x,y,z], Ord(M);
  I :=Ideal(a[1]t^11-a[2]x^11+a[7]xyt^5,
            a[3]t^5-a[4]y^5-a[8]xyzt^2,
    a[5]t^7-a[6]z^7+a[9]x^3y^4);
  ReducedGBasis5x(I, Record[NumParams:=6]);  // Very Long


  // To see what is happening
  ReducedGBasis5x(I, Record[NumParams:=6, VerbosityLevel:=4]);


  // Specializing the parameters things are easy
  LT5(Subst(I,[[Indet(Ind),Rand(-100,100)]|Ind In 1..9]));
```

```
// We want to perform some operations with the degree [1,2,3,4] on
//Z/(32003)[x,y,z,t]

Use R::=ZZ/(32003)[x,y,z,t,a];
I:=Ideal(xy^2,xz^2,yt^4,xyzt);
SyzOfGens(I);
Syz(I,1);

Module([[0, yt, 0, -z],
        [zt, 0, 0, -y],
        [z^2, -y^2, 0, 0],
[0, 0, xz, -t^3],
[t^4, 0, -xy, 0]]);

IsHomog(Gens(Syz(I,1)));

M:=Mat(SyzOfGens(I));M;


Shifts:=Mat([
  [a^3, 0, 0, 0],
  [0, a^3, 0, 0],
  [0, 0, a^5, 0],
  [0, 0, 0, a^4]
]);

M*Shifts;
```

```
S:=Module(M*Shifts);

IsHomog(Gens(S));

  Use S ::= QQ[a,b,c], Weights(1,2,3);
  Deg(b);
  W := Mat([[1,2,3],[4,5,6]]);
  Use S ::= QQ[a,b,c], Weights(W);
  Deg(b);
  MDeg(b);  -- the multi-degree of b
  Deg(b^3+a^2c);
  MDeg(b^3+a^2c);
  WeightsMatrix();

  Use R ::= QQ[x,y,z];
  IsHomog([z-xy, x-y^2]);
  Use R ::= QQ[x,y,z], Weights(Mat([[1,1,2]]));
  IsHomog([z-xy, x^2-y^2]);

///////////////////////////////////////////////////////////////////////////////
// Exercise:
// There exist weights L1,..,L4 such that the polynomial
// x^2y-zt+yz
// is homogeneous?


  ///// Operations with parameters

  M := CoCoA5.AddParamOrdMat(DegRevLexMat(3), 2); -- 2 parameters
  Use ZZ/(32003)[a,b, x,y,z], Ord(M);
  II := Ideal(x-y);
  I := (a-b+1) * x * II;
  J := (a+1) * y * II;
  Intersection5x(I, J, Record[NumParams:=2]); -- 2 parameters
  Intersection5(I, J);
  Intersection(I, J);







/////// Some cocoa5
```

```
WNeg := StdDegRevLexMat(3);  WNeg;

WFine := LexMat(3);  WFine;

P := NewPolyRing(QQ, ["x","y","z"], WNeg, 1);

P := NewPolyRing(QQ, ["x","y","z"], WFine, 3);

W2St:=RowMat([1,1,1]);

P := NewPolyRing(QQ, ["x","y","z"], CompleteToOrd(W2St), NumRows(W2St));

Use P ::= QQ[x,y,z];
I := ***Ideal(x^3, xy, yz^2)***;
HilbertSeries(P/I);


Qa_i ::= QQ[a,b,c,d,e,f,g,h,i];
Qparam := NewFractionField(Qa_i);
P := NewPolyRing(Qparam, ["t","x","y","z"]);
use P;


  I :=***Ideal(at^11-bx^11+cxyt^5,
           ft^5-ey^5-dxyzt^2,
    gt^7-hz^7+ix^3y^4)***;
  ReducedGBasis(I);
  // ~30m
```