```
//=====================
// ZETA FUNCTIONS
//=====================

A<X,Y,Z> := ProjectiveSpace(GF(37),2);
C:=Curve(A,X^3+Y^3-2*Z^3);

Zeta:= ZetaFunction(C);
Zeta;

A<X,Y,Z> := ProjectiveSpace(GF(13),2);
C:=Curve(A,X^3+Y^3-2*Z^3);

Zeta:= ZetaFunction(C);
Zeta;

// Verify Riemann Hypothesis!

P<t>:=PolynomialRing(Integers());
P!Denominator(Zeta);
Factorization($1);

f:=P!Numerator(Zeta);
K<k>:=NumberField(f);
fK<y> := ChangeRing(f,K);fK;
Factorization(fK);
roo:=Roots(fK);
Norm(roo[1][1]),Norm(roo[2][1]);


//========================
//  ELLIPTIC CURVES OVER Q
//========================

E:=EllipticCurve("11a3");
E;

E:=EllipticCurve([0,-1,1,0,0]);
E;

TorsionSubgroup(E);

T,mappy:=TorsionSubgroup(E);
mappy;

T.1;
mappy(T.1);
```

```
P:=mappy(T.1);
2*P;
3*P;
4*P;
5*P;

t:=11;
Et := EllipticCurveFromjInvariant((t+27)*(t+243)^3/t^3);
IsogenousCurves(Et);

Rank(E);

QuadraticTwist(E,2);
E2:=$1;
Rank(E2);

for i:=2 to 20 do
  E2:=QuadraticTwist(E,i);
  print Rank(E2);
end for;

max:=0;
wintwist:=0;
for i:=2 to 100 do
  E2:=QuadraticTwist(E,i);
  r:=Rank(E2);
  if r gt max then
     max:=r;
     wintwist:=i;
  end if;
end for;
max, wintwist;



E:=EllipticCurve([0,0,1,-7,6]);
TorsionSubgroup(E);
Rank(E);


MordellWeilShaInformation(E);

time rank, gens, sha :=MordellWeilShaInformation(E : ShaInfo);
rank;
gens;
sha;


E := EllipticCurve("571a1");
time rank, gens, sha :=MordellWeilShaInformation(E);
time rank, gens, sha :=MordellWeilShaInformation(E : ShaInfo);


// Code from LMFDB

// Magma code for working with elliptic curve 5077.a1


// Define the curve:
E := EllipticCurve([0, 0, 1, -7, 6]); // or
E := EllipticCurve("5077a1");

// Torsion subgroup:
TorsionSubgroup(E);

// Integral points:
IntegralPoints(E);

// Conductor:
Conductor(E);

// Discriminant:
Discriminant(E);

// j-invariant:
jInvariant(E);

// Rank:
Rank(E);

// Regulator:
Regulator(E);

// Real Period:
RealPeriod(E);

// Tamagawa numbers:
TamagawaNumbers(E);

// Torsion order:
Order(TorsionSubgroup(E));

// Order of Sha:
MordellWeilShaInformation(E);

// q-expansion of modular form:
ModularForm(E);

// Modular degree:
ModularDegree(E);
```

```
// Special L-value:
Lr1 where r,Lr1 := AnalyticRank(E: Precision:=12);

// Local data:
[LocalInformation(E,p) : p in BadPrimes(E)];

// mod p Galois image:
[GaloisRepresentation(E,p): p in PrimesUpTo(20)];


// VERIFY BSD

r,Lr1 := AnalyticRank(E: Precision:=12);
L:=LSeries(E);
Evaluate(L,1);
Evaluate(L,1 : Derivative:=3);
$1/Factorial(3);

rank,gens,sha:=MordellWeilShaInformation(E);

shaorder:=#sha;
shaorder;
omega:=RealPeriod(E);
omega;
reg:=Regulator(E);
reg;
TamagawaNumbers(E);
tam:=1;
TorsionSubgroup(E);
tors:=1;
shaorder*omega*reg*tam/tors^2;
Evaluate(L,1 : Derivative:=3)/6;

// FINDING POINTS ON HIGHER GENUS CURVES-- AN EXAMPLE

PP<t,s,z>:=ProjectiveSpace(Rationals(),2);
f0:=(t^4+t^2*z^2+z^4)^3*s^3 - (s+27*z)*(s+243*z)^3*t^4*(t^2+z^2)^2*z^3;
C0:=Curve(PP,f0);
C0


Genus(C0);
f:=(t^2+t*z+z^2)^3*s^3 - (s+27*z)*(s+243*z)^3*t^2*(t+z)^2*z;
C:=Curve(PP,f);
Genus(C);


Aut:=AutomorphismGroup(C); // TAKES A LONG TIME TO COMPUTE THIS STEP!
G2:=AutomorphismGroup(C,[Aut.2]);
```

```
CG2,phi:=CurveQuotient(G2);


Genus(CG2);
CG2;
CG2b,mappy:=SimplifiedModel(CG2);
J:=Jacobian(CG2b);
RankBounds(J);
S:=Chabauty0(J);
invmappy:=Inverse(mappy);
P:=invmappy(S[1]); Q:=invmappy(S[2]);

schP:=P@@phi;
schQ:=Q@@phi;
RationalPoints(schP);
RationalPoints(schQ);
```

```
// Omega = 2*(real period) if E(R) is disconnected!!

omega:=2*omega;
sharoder*omega*reg*tam/tors^2;
```