```
//CTNT -- A virtual tour of Magma

//Online calculator: http://magma.maths.usyd.edu.au/calc/

//Commands need to end in ";" !!

/*
This is a multi lined comment
This is how you batch comment things out!
*/


ZZ := Integers();
QQ := Rationals();
RR := RealField();  //can change the precision
CC := ComplexField(); //can change the precision

RR:=RealField(100);
Pi(RR);
Exp(1);
Log(RR!Exp(1));
Log(Exp(RR!1));


?RealField
?1

F:=GF(25);
R:=Integers(25);
F!(1/6);
R!(1/6);

EulerPhi(25);
173^20 mod 25;


P<x>:=PolynomialRing(Rationals());
Factorization(x^6-1);

P<x>:=PolynomialRing(GF(13));
Factorization(x^6-1);

G := SL(2,Integers());
A := G![0,1,-1,0];
B := G![1,1,0,1];
A;
A*B*A^(-1)*B^(-1);

[1..5];
[i : i in [1..5]];
```

```
PrimesUpTo(97);  //closed intervals
PrimesInInterval(11, 100);  //closed intervals
RandomPrime(100); //random prime less than 2^100

N:=10^2;
List1 := [p : p in PrimesUpTo(N) | Integers(4)!p eq 1];
List1;

Divisors(12);
NumberOfDivisors(12);
MoebiusMu(12);
MoebiusMu(11);
MoebiusMu(10);

sum:=0;
for i:=1 to 10 do
        sum:=sum+NumberOfDivisors(i);
end for;
sum;



sumdiv := function(n)
    sum:=0;
    for i:=1 to n do
        sum:=sum+NumberOfDivisors(i);
    end for;
    return sum;
end function;

sumdiv(10);



LegendreSymbol(2,7);
LegendreSymbol(5,7);

KroneckerSymbol(7,2);


IsSquarefree(10);
IsSquarefree(12);

// NUMBER FIELDS

P<x>:=PolynomialRing(Rationals());
K<i>:=NumberField(x^2+1);
i^2;
(3+4*i)*(3-4*i);
```

```
OK:=MaximalOrder(K);
OK;

OK.1 eq 1;
OK.2 eq i;
OK.2^2;

Discriminant(K);

F<f>:=NumberField(x^2+3);
Discriminant(F);
Discriminant(MaximalOrder(F));

L<a>:=NumberField(x^3+x+1);
OL:=MaximalOrder(L);
Factorization(2*OL);
Factorization(3*OL);
Factorization(31*OL);
Factorization(47*OL);

IsNormal(L);
GaloisGroup(L);

SplittingField(L);
SplittingField(x^3+x+1);

A:=SplittingField(L);
B:=SplittingField(x^3+x+1);

IsIsomorphic(A,B);

G:=GaloisGroup(A);
G;
GroupName(G);

S:=Subgroups(G); //output is a list "records"
S;

[H`subgroup : H in S];
list:=[H`subgroup : H in S];

FixedField(A,list[1]);
FixedField(A,list[2]);
FixedField(A,list[3]);



//CLASS GROUPS
```

```
P<x> := PolynomialRing(Rationals());P;
f := x^2+5;
K<a> := NumberField(f);
O:=MaximalOrder(K);
O.1 eq 1;
O.2 eq a;
G, map := ClassGroup(K);G;
#G;
Generators(G);
G.1;
map(G.1);
Norm($1);

I1 := ideal<O|[1+a,3]>;I1;
I2 := ideal<O|[1+a,2]>;I2;
IsPrime(I2);

Inverse(map)(I1);
Inverse(map)(I2);

Inverse(map)(I1*I2);

F:=NumberField(x^2-2*3*5*7*11*13);
G:=ClassGroup(F);
IsAbelian(G);
ElementaryDivisors(G);
GroupName(G);



// p-ADIC INTEGERS

Z7 := pAdicRing(7,20);
Q7 := pAdicField(7,20);
P7<x> := PolynomialRing(Q7);

Z7!(1/3);

Factorization(x^2-5);
Factorization(x^2-2);

Factorization(x^3+x+5);

HenselLift(x^3+x+5,Q7!1);


L<a> := ext<Q7|x^2-5>;
L;
L!(1/3+2*a);
```

```
//CURVES

A<x,y> := AffineSpace(GF(37),2);
C:=Curve(A,x^3+y^3-2);
Genus(C);
IsSmooth(C);
RationalPoints(C);
ProjectiveClosure(C);

P:=C![1,1];
E:=EllipticCurve(C,P);

A<X,Y,Z> := ProjectiveSpace(Rationals(),2);
C:=Curve(A,X^3+Y^3-2*Z^3);
P:=C![1,1,1];
E:=EllipticCurve(C,P);
E;



//======================
// 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
//=========================

A<X,Y,Z> := ProjectiveSpace(Rationals(),2);
C:=Curve(A,X^3+Y^3-2*Z^3);
P:=C![1,1,1];
E:=EllipticCurve(C,P);
E;
MinimalModel(E);
E:=$1;
jInvariant(E);
Discriminant(E);
BadPrimes(E);
ReductionType(E,3);


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;

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;




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

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