Finite_Fields

```
# RECHNEN IN ENDLICHEN KOERPER MIT SAGE
# %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
R.<X> = PolynomialRing(Integers(2))
print R
```

```
    Univariate Polynomial Ring in X over Ring of integers modulo 2
    (using NTL)
```

```
# R ist der Polynomring F_2[X]
```

```
# in R koennen wir rechnen:
```

```
print (X^2+X+1)^2
print (X^2+2)*(X+1)^7
```

```
    X^4 + X^2 + 1
    X^9 + X^8 + X^7 + X^6 + X^5 + X^4 + X^3 + X^2
```

```
# ein Polynom in R kann faktorisiert werden:
```

```
for n in range(13,23):
    Fakt=factor(X^23+X^n+1)
    print n,": ",Fakt
    print
```

```
    13 :  (X^2 + X + 1) * (X^3 + X + 1) * (X^18 + X^17 + X^16 + X^15 +
    X^14 + X^12 + X^10 + X^8 + X^5 + X^4 + 1)

    14 :  X^23 + X^14 + 1

    15 :  (X^6 + X + 1) * (X^7 + X^6 + X^5 + X^2 + 1) * (X^10 + X^9 +
    X^7 + X^6 + X^4 + X + 1)

    16 :  (X^2 + X + 1) * (X^10 + X^9 + X^8 + X^7 + X^5 + X^4 + 1) *
    (X^11 + X^9 + X^8 + X^7 + X^3 + X + 1)

    17 :  (X^3 + X^2 + 1) * (X^4 + X + 1) * (X^16 + X^15 + X^14 + X^13
    X^12 + X^10 + X^8 + X^4 + X^3 + X + 1)

    18 :  X^23 + X^18 + 1

    19 :  (X^2 + X + 1) * (X^8 + X^7 + X^3 + X^2 + 1) * (X^13 + X^10 +
    X^9 + X^8 + X^7 + X^6 + X^4 + X^3 + X^2 + X + 1)

    20 :  (X^3 + X + 1) * (X^5 + X^4 + X^2 + X + 1) * (X^15 + X^14 +
    X^12 + X^10 + X^8 + X^7 + X^6 + X^5 + 1)

    21 :  (X^4 + X^3 + 1) * (X^19 + X^18 + X^15 + X^11 + X^10 + X^9 +
```

```
X^8 + X^6 + X^4 + X^3 + 1)

22 :  (X^2 + X + 1) * (X^8 + X^6 + X^5 + X^2 + 1) * (X^13 + X^8 +
X^6 + X^5 + X^2 + X + 1)
```

```
# wir sehen, dass die Polynome X^23 + X^14 + 1
# und X^23 + X^18 + 1 irreduzibel sind
```

```
# nun suchen wir weitere irreduzible Polynome
# der Form X^23 + X^n + 1 fuer 0<n<23:
```

```
for n in range(1,23):
    Fakt=factor(X^23+X^n+1)
    if len(Fakt)==1:
        print n,": ",Fakt
```

```
5 :  X^23 + X^5 + 1
9 :  X^23 + X^9 + 1
14 :  X^23 + X^14 + 1
18 :  X^23 + X^18 + 1
```

```
# nun faktorisieren wir X^(2^n) - X
# fuer 0<n<8:
```

```
for n in range(1,8):
    Fakt=factor(X^(2^n)-X)
    print n,": ",Fakt
    print
```

```
1 :  X * (X + 1)

2 :  X * (X + 1) * (X^2 + X + 1)

3 :  X * (X + 1) * (X^3 + X + 1) * (X^3 + X^2 + 1)

4 :  X * (X + 1) * (X^2 + X + 1) * (X^4 + X + 1) * (X^4 + X^3 + 1)
(X^4 + X^3 + X^2 + X + 1)

5 :  X * (X + 1) * (X^5 + X^2 + 1) * (X^5 + X^3 + 1) * (X^5 + X^3
X^2 + X + 1) * (X^5 + X^4 + X^2 + X + 1) * (X^5 + X^4 + X^3 + X +
* (X^5 + X^4 + X^3 + X^2 + 1)

6 :  X * (X + 1) * (X^2 + X + 1) * (X^3 + X + 1) * (X^3 + X^2 + 1)
(X^6 + X + 1) * (X^6 + X^3 + 1) * (X^6 + X^4 + X^2 + X + 1) * (X^6
X^4 + X^3 + X + 1) * (X^6 + X^5 + 1) * (X^6 + X^5 + X^2 + X + 1) *
(X^6 + X^5 + X^3 + X^2 + 1) * (X^6 + X^5 + X^4 + X + 1) * (X^6 + X
+ X^4 + X^2 + 1)

7 :  X * (X + 1) * (X^7 + X + 1) * (X^7 + X^3 + 1) * (X^7 + X^3 +
X^2 + X + 1) * (X^7 + X^4 + 1) * (X^7 + X^4 + X^3 + X^2 + 1) * (X^
+ X^5 + X^2 + X + 1) * (X^7 + X^5 + X^3 + X + 1) * (X^7 + X^5 + X^
+ X^3 + 1) * (X^7 + X^5 + X^4 + X^3 + X^2 + X + 1) * (X^7 + X^6 +
* (X^7 + X^6 + X^3 + X + 1) * (X^7 + X^6 + X^4 + X + 1) * (X^7 + X
```

```
+ X^4 + X^2 + 1) * (X^7 + X^6 + X^5 + X^2 + 1) * (X^7 + X^6 + X^5
X^3 + X^2 + X + 1) * (X^7 + X^6 + X^5 + X^4 + 1) * (X^7 + X^6 + X^
+ X^4 + X^2 + X + 1) * (X^7 + X^6 + X^5 + X^4 + X^3 + X^2 + 1)
```

```
Fakt=factor(X^(2^2)-X)
print "Anzahl irreduzible Faktoren: ", len(Fakt)
print Fakt
```

```
Anzahl irreduzible Faktoren:  3
X * (X + 1) * (X^2 + X + 1)
```

```
Fakt=factor(X^(2^5)-X)
print "Anzahl irreduzible Faktoren: ", len(Fakt)
print Fakt
```

```
Anzahl irreduzible Faktoren:  8
X * (X + 1) * (X^5 + X^2 + 1) * (X^5 + X^3 + 1) * (X^5 + X^3 + X^2
X + 1) * (X^5 + X^4 + X^2 + X + 1) * (X^5 + X^4 + X^3 + X + 1) *
(X^5 + X^4 + X^3 + X^2 + 1)
```

```
Fakt=factor(X^(2^10)-X)
print "Anzahl irreduzible Faktoren: ", len(Fakt)
print Fakt
```

```
Anzahl irreduzible Faktoren:  108
X * (X + 1) * (X^2 + X + 1) * (X^5 + X^2 + 1) * (X^5 + X^3 + 1) *
(X^5 + X^3 + X^2 + X + 1) * (X^5 + X^4 + X^2 + X + 1) * (X^5 + X^4
X^3 + X + 1) * (X^5 + X^4 + X^3 + X^2 + 1) * (X^10 + X^3 + 1) *
(X^10 + X^3 + X^2 + X + 1) * (X^10 + X^4 + X^3 + X + 1) * (X^10 +
X^4 + X^3 + X^2 + 1) * (X^10 + X^5 + X^2 + X + 1) * (X^10 + X^5 +
X^3 + X^2 + 1) * (X^10 + X^5 + X^4 + X^2 + 1) * (X^10 + X^6 + X^2
X + 1) * (X^10 + X^6 + X^4 + X + 1) * (X^10 + X^6 + X^5 + X + 1) *
(X^10 + X^6 + X^5 + X^2 + 1) * (X^10 + X^6 + X^5 + X^3 + X^2 + X +
1) * (X^10 + X^7 + 1) * (X^10 + X^7 + X^3 + X + 1) * (X^10 + X^7 +
X^4 + X^3 + 1) * (X^10 + X^7 + X^5 + X^3 + 1) * (X^10 + X^7 + X^5
X^3 + X^2 + X + 1) * (X^10 + X^7 + X^6 + X^2 + 1) * (X^10 + X^7 +
X^6 + X^3 + 1) * (X^10 + X^7 + X^6 + X^4 + X^2 + X + 1) * (X^10 +
X^7 + X^6 + X^5 + X^2 + X + 1) * (X^10 + X^7 + X^6 + X^5 + X^3 + X
+ 1) * (X^10 + X^7 + X^6 + X^5 + X^4 + X + 1) * (X^10 + X^7 + X^6
X^5 + X^4 + X^3 + X^2 + X + 1) * (X^10 + X^8 + X^3 + X + 1) * (X^1
+ X^8 + X^3 + X^2 + 1) * (X^10 + X^8 + X^4 + X^3 + 1) * (X^10 + X^
+ X^4 + X^3 + X^2 + X + 1) * (X^10 + X^8 + X^5 + X + 1) * (X^10 +
X^8 + X^5 + X^4 + 1) * (X^10 + X^8 + X^5 + X^4 + X^3 + X^2 + 1) *
(X^10 + X^8 + X^6 + X + 1) * (X^10 + X^8 + X^6 + X^4 + X^2 + X + 1
* (X^10 + X^8 + X^6 + X^5 + 1) * (X^10 + X^8 + X^6 + X^5 + X^2 + X
1) * (X^10 + X^8 + X^6 + X^5 + X^3 + X + 1) * (X^10 + X^8 + X^7 +
X^2 + 1) * (X^10 + X^8 + X^7 + X^3 + X^2 + X + 1) * (X^10 + X^8 +
X^7 + X^4 + X^2 + X + 1) * (X^10 + X^8 + X^7 + X^4 + X^3 + X + 1)
(X^10 + X^8 + X^7 + X^5 + 1) * (X^10 + X^8 + X^7 + X^5 + X^3 + X +
1) * (X^10 + X^8 + X^7 + X^5 + X^4 + X^3 + 1) * (X^10 + X^8 + X^7
X^6 + 1) * (X^10 + X^8 + X^7 + X^6 + X^2 + X + 1) * (X^10 + X^8 +
X^7 + X^6 + X^5 + X^2 + 1) * (X^10 + X^8 + X^7 + X^6 + X^5 + X^4 +
X^2 + X + 1) * (X^10 + X^8 + X^7 + X^6 + X^5 + X^4 + X^3 + X + 1)
(X^10 + X^9 + X^4 + X + 1) * (X^10 + X^9 + X^4 + X^2 + 1) * (X^10
X^9 + X^5 + X + 1) * (X^10 + X^9 + X^5 + X^2 + 1) * (X^10 + X^9 +
X^5 + X^4 + 1) * (X^10 + X^9 + X^5 + X^4 + X^2 + X + 1) * (X^10 +
X^9 + X^6 + X + 1) * (X^10 + X^9 + X^6 + X^3 + X^2 + X + 1) * (X^1
```

```
+ X^9 + X^6 + X^4 + 1) * (X^10 + X^9 + X^6 + X^4 + X^3 + X + 1) *
(X^10 + X^9 + X^6 + X^5 + X^4 + X^3 + 1) * (X^10 + X^9 + X^6 + X^5
X^4 + X^3 + X^2 + X + 1) * (X^10 + X^9 + X^7 + X^2 + 1) * (X^10 +
X^9 + X^7 + X^3 + 1) * (X^10 + X^9 + X^7 + X^5 + X^2 + X + 1) *
(X^10 + X^9 + X^7 + X^5 + X^3 + X^2 + 1) * (X^10 + X^9 + X^7 + X^5
X^4 + X^2 + 1) * (X^10 + X^9 + X^7 + X^5 + X^4 + X^3 + X^2 + X + 1
* (X^10 + X^9 + X^7 + X^6 + 1) * (X^10 + X^9 + X^7 + X^6 + X^3 + X
+ 1) * (X^10 + X^9 + X^7 + X^6 + X^4 + X + 1) * (X^10 + X^9 + X^7
X^6 + X^4 + X^3 + X^2 + X + 1) * (X^10 + X^9 + X^7 + X^6 + X^5 + X
+ X^2 + X + 1) * (X^10 + X^9 + X^7 + X^6 + X^5 + X^4 + X^3 + X^2 +
1) * (X^10 + X^9 + X^8 + X^3 + X^2 + X + 1) * (X^10 + X^9 + X^8 +
X^4 + 1) * (X^10 + X^9 + X^8 + X^4 + X^2 + X + 1) * (X^10 + X^9 +
X^8 + X^4 + X^3 + X^2 + 1) * (X^10 + X^9 + X^8 + X^5 + 1) * (X^10
X^9 + X^8 + X^5 + X^3 + X + 1) * (X^10 + X^9 + X^8 + X^5 + X^4 + X
+ 1) * (X^10 + X^9 + X^8 + X^5 + X^4 + X^3 + 1) * (X^10 + X^9 + X^
+ X^6 + X^2 + X + 1) * (X^10 + X^9 + X^8 + X^6 + X^3 + X^2 + 1) *
(X^10 + X^9 + X^8 + X^6 + X^4 + X^2 + 1) * (X^10 + X^9 + X^8 + X^6
X^4 + X^3 + 1) * (X^10 + X^9 + X^8 + X^6 + X^5 + X + 1) * (X^10 +
X^9 + X^8 + X^6 + X^5 + X^4 + X^3 + X + 1) * (X^10 + X^9 + X^8 + X
+ X^5 + X^4 + X^3 + X^2 + 1) * (X^10 + X^9 + X^8 + X^7 + 1) * (X^1
+ X^9 + X^8 + X^7 + X^2 + X + 1) * (X^10 + X^9 + X^8 + X^7 + X^3 +
X^2 + 1) * (X^10 + X^9 + X^8 + X^7 + X^4 + X + 1) * (X^10 + X^9 +
X^8 + X^7 + X^5 + X^3 + 1) * (X^10 + X^9 + X^8 + X^7 + X^5 + X^4 +
1) * (X^10 + X^9 + X^8 + X^7 + X^6 + X^2 + 1) * (X^10 + X^9 + X^8
X^7 + X^6 + X^4 + X^3 + X + 1) * (X^10 + X^9 + X^8 + X^7 + X^6 + X
+ X^3 + X + 1) * (X^10 + X^9 + X^8 + X^7 + X^6 + X^5 + X^4 + X + 1
* (X^10 + X^9 + X^8 + X^7 + X^6 + X^5 + X^4 + X^3 + 1) * (X^10 + X
    + X^8 + X^7 + X^6 + X^5 + X^4 + X^3 + X^2 + X + 1)
```

```
2*1+1*2+6*5+99*10
```

```
    1024
```

```
# das war ein Beispiel fuer Korollar 16.9
```

```
# wenn h ueber F_2 irreduzibel ist, so ist F_2[X]/(h) ein Koerper
# wir suchen nun irreduzible Polynome der Form X^9 + X^n + 1
```

```
R.<X> = PolynomialRing(Integers(2))
reset('F.<X>')
```

```
for n in range(1,9):
    Fakt=factor(X^9+X^n+1)
    if len(Fakt)==1:
        print n,": ",Fakt
```

```
    1 :  X^9 + X + 1
    4 :  X^9 + X^4 + 1
    5 :  X^9 + X^5 + 1
    8 :  X^9 + X^8 + 1
```

```
# als Koerper waehlen wir zuerst K5 := F_2[X]/(X^9+X^5+1):
```

```
F.<X>=R.quotient([X^9 + X^5 + 1])
```

```
# die Ordnung von X in K5 ist 511,
# d.h. X generiert K5*
```

```
for n in range(512):
    if X^n==1:
        print "X ^",n," = ",X^n
```

```
X ^ 0   =  1
X ^ 511   =  1
```

```
# die folgenden Polynome sind die 9 Nullstellen
# des Polynoms h = X^9 + X^5 + 1:
```

```
for n in range(10):
   print "X ^",2^n," = ",X^(2^n)
   print "eingesetz in h: ",(X^(2^n))^9+(X^(2^n))^5+1
```

```
X ^ 1   =  X
eingesetz in h:  0
X ^ 2   =  X^2
eingesetz in h:  0
X ^ 4   =  X^4
eingesetz in h:  0
X ^ 8   =  X^8
eingesetz in h:  0
X ^ 16   =  X^8 + X^7 + X^3
eingesetz in h:  0
X ^ 32   =  X^8 + X^7 + X^5 + X^3 + X
eingesetz in h:  0
X ^ 64   =  X^8 + X^7 + X^6 + X^5 + X^3 + X^2
eingesetz in h:  0
X ^ 128   =  X^7 + X^6 + X^5 + X^4
eingesetz in h:  0
X ^ 256   =  X^5 + X^3
eingesetz in h:  0
X ^ 512   =  X
eingesetz in h:  0
```

```
# als Koerper waehlen wir nun K8 := F_2[X]/(X^9+X^8+1)
# d.h. h = X^9+X^8+1
# die Ordnung von X in K8 ist 73:
```

```
reset('F.<X>')
R.<X> = PolynomialRing(Integers(2))
F.<X>=R.quotient([X^9 + X^8 + 1])
for n in range(512):
    if X^n==1:
        print "X ^",n," = ",X^n
```

```
X ^ 0   =  1
X ^ 73   =  1
```

```
X ^ 146  =  1
X ^ 219  =  1
X ^ 292  =  1
X ^ 365  =  1
X ^ 438  =  1
X ^ 511  =  1
```

```
for n in range(10):
   print "X ^",2^n," = ",X^(2^n)
   print "eingesetz in h: ",(X^(2^n))^9+(X^(2^n))^8+1
```

```
X ^ 1  =  X
eingesetz in h:  0
X ^ 2  =  X^2
eingesetz in h:  0
X ^ 4  =  X^4
eingesetz in h:  0
X ^ 8  =  X^8
eingesetz in h:  0
X ^ 16  =  X^8 + X^7 + X^6 + X^5 + X^4 + X^3 + X^2 + X + 1
eingesetz in h:  0
X ^ 32  =  X^8 + X^7 + X^4 + X^3 + 1
eingesetz in h:  0
X ^ 64  =  X^8 + X^7 + 1
eingesetz in h:  0
X ^ 128  =  X^7 + X^6 + 1
eingesetz in h:  0
X ^ 256  =  X^5 + X^4 + 1
eingesetz in h:  0
X ^ 512  =  X
eingesetz in h:  0
```

```
# Formale adjunktion von 3-ten Wurzeln von W
# %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
S.<X,Y,Z,W> = PolynomialRing(QQ)
print S
```

```
Multivariate Polynomial Ring in X, Y, Z, W over Rational Field
```

```
T.<X,Y,Z,W>=S.quotient([X^3-W,Y^3-W,Z^3-W,
X^2+X*Y+Y^2,X^2+X*Z+Z^2,Z^2+Z*Y+Y^2,X+Y+Z])
print T
```

```
Quotient of Multivariate Polynomial Ring in X, Y, Z, W over Ration
Field by the ideal (X^3 - W, Y^3 - W, Z^3 - W, X^2 + X*Y + Y^2, X^
+ X*Z + Z^2, Y^2 + Y*Z + Z^2, X + Y + Z)
```

```
X^2*Z
```

```
Y*Z^2
```

```
Y^2
```

```
-Y*Z - Z^2
```

```
# Verschluesseln mit elliptischen Kurven
# ueber endlichen Koerpern mit char=2
# %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
reset('F.<X>')
R.<Z> = PolynomialRing(Integers(2))
F.<Z>=R.quotient(Z^23 + Z^9 + 1)
print F
```

```
Univariate Quotient Polynomial Ring in Z over Ring of integers
modulo 2 with modulus Z^23 + Z^9 + 1
```

```
var('a,b,c,d,e');
EllipticCurve([a,b,c,d,e])
```

```
Elliptic Curve defined by y^2 + a*x*y + c*y = x^3 + b*x^2 + d*x +
over Symbolic Ring
```

```
parameters=[0,0,Z,Z^2,0]
E=EllipticCurve(F,parameters)
print E
P=E([Z^2,Z^3])
print
print "der Punkt ",P," liegt auf der elliptischen Kurve E"
```

```
Elliptic Curve defined by y^2 + Z*y = x^3 + Z^2*x over Univariate
Quotient Polynomial Ring in Z over Ring of integers modulo 2 with
modulus Z^23 + Z^9 + 1

der Punkt  (Z^2 : Z^3 : 1)  liegt auf der elliptischen Kurve E
```

```
# wir koennen nun P+P oder auch m*P berechnen:
```

```
6748674283760980942874 6*P
```

```
(Z^22 + Z^19 + Z^18 + Z^16 + Z^14 + Z^12 + Z^11 + Z^10 + Z^8 + Z^5
Z^2 + Z : Z^22 + Z^20 + Z^18 + Z^17 + Z^13 + Z^12 + Z^11 + Z^9 + Z
+ Z^6 + Z^4 + Z^3 + Z^2 : 1)
```

```
# Das Diffie-Hellman Verschluesselungsverfahren mit elliptischen
# Kurven über einem endlichen Körper der Charakteristik 2
#########################################################
# Alice und Bob waehlen eine elliptische Kurve E ueber einem
# endlichen Koerper F, sowie einen Punkt Q auf E (oeffentlich):

parameters=[0,0,Z,Z^2,0]
E=EllipticCurve(F,parameters)
print E
Q=E([Z^2,Z^3])
```

```
Elliptic Curve defined by y^2 + Z*y = x^3 + Z^2*x over Univariate
Quotient Polynomial Ring in Z over Ring of integers modulo 2 with
modulus Z^23 + Z^9 + 1
```

```
# Alice waehlt eine Zahl m und Bob eine Zahl n (geheim).
# dann sendet Alice an Bob m*Q und Bob an Alice n*Q
# der gemeinsame Schluessel ist dann z.B. die erste
# Koordinate des Punkts m*n*Q:

m=2746876583764
```

```
n=9238709873583
AB=m*Q
BA=n*Q
print "Alice -> Bob:"
print m*Q
print
print "Bob -> Alice:"
print n*Q
print
print "Gemeinsamer Schlüssel"
print "berechnet von Bob:"
print (n*AB)[0]
print "berechnet von Alice:"
print (m*BA)[0]
```

```
Alice -> Bob:
(Z^22 + Z^20 + Z^19 + Z^10 + Z^9 + Z^8 + Z^6 + Z^5 + Z^4 + Z^2 + Z
Z^22 + Z^21 + Z^18 + Z^17 + Z^16 + Z^9 + Z^8 + Z^5 + Z^3 + 1 : 1)

Bob -> Alice:
(Z^14 + Z^10 + Z^8 + Z^3 + Z + 1 : Z^22 + Z^21 + Z^18 + Z^17 + Z^1
+ Z^9 + Z^8 + Z + 1 : 1)

Gemeinsamer Schlüssel
berechnet von Bob:
Z^22 + Z^20 + Z^19 + Z^18 + Z^16 + Z^14 + Z^10 + Z^9 + Z^8 + Z^7 +
Z^5 + Z^2
berechnet von Alice:
Z^22 + Z^20 + Z^19 + Z^18 + Z^16 + Z^14 + Z^10 + Z^9 + Z^8 + Z^7 +
Z^5 + Z^2
```