

# Algebraic Number Theory

(PARI-GP version 2.15.3)

## Binary Quadratic Forms

create  $ax^2 + bxy + cy^2$  **Qfb**( $a, b, c$ ) or **Qfb**( $[a, b, c]$ )  
reduce  $x$  ( $s = \sqrt{D}$ ,  $l = \lfloor s \rfloor$ ) **qfbred**( $x, \{flag\}, \{D\}, \{l\}, \{s\}$ )  
return  $[y, g]$ ,  $g \in \text{SL}_2(\mathbf{Z})$ ,  $y = g \cdot x$  reduced **qfbreds12**( $x$ )  
composition of forms  $x*y$  or **qfbnucomp**( $x, y, l$ )  
 $n$ -th power of form  $x^n$  or **qfbnupow**( $x, n$ )  
composition **qfbcomp**( $x, y$ )  
... without reduction **qfbcomppraw**( $x, y$ )  
 $n$ -th power **qfbpow**( $x, n$ )  
... without reduction **qfbpowraw**( $x, n$ )  
prime form of disc.  $x$  above prime  $p$  **qfbprimeform**( $x, p$ )  
class number of disc.  $x$  **qfbclassno**( $x$ )  
Hurwitz class number of disc.  $x$  **qfbhclassno**( $x$ )  
solve  $Q(x, y) = n$  in integers **qfbsolve**( $Q, n$ )  
solve  $x^2 + Dy^2 = p$ ,  $p$  prime **qfbcornacchia**( $D, p$ )  
...  $x^2 + Dy^2 = 4p$ ,  $p$  prime **qfbcornacchia**( $D, 4 * p$ )

## Quadratic Fields

quadratic number  $\omega = \sqrt{x}$  or  $(1 + \sqrt{x})/2$  **quadgen**( $x$ )  
minimal polynomial of  $\omega$  **quadpoly**( $x$ )  
discriminant of **Q**( $\sqrt{x}$ ) **quaddisc**( $x$ )  
regulator of real quadratic field **quadregulator**( $x$ )  
fundamental unit in  $O_D$ ,  $D > 0$  **quadunit**( $D, \{ 'w \}$ )  
norm of fundamental unit in  $O_D$  **quadunitnorm**( $D$ )  
index of  $O_{Df_2}^\times$  in  $O_D^\times$  **quadunitindex**( $D, f$ )  
class group of **Q**( $\sqrt{D}$ ) **quadclassunit**( $D, \{flag\}, \{t\}$ )  
Hilbert class field of **Q**( $\sqrt{D}$ ) **quadhilbert**( $D, \{flag\}$ )  
... using specific class invariant ( $D < 0$ ) **polclass**( $D, \{inv\}$ )  
ray class field modulo  $f$  of **Q**( $\sqrt{D}$ ) **quadrays**( $D, f, \{flag\}$ )

## General Number Fields: Initializations

The number field  $K = \mathbf{Q}[X]/(f)$  is given by irreducible  $f \in \mathbf{Q}[X]$ .  
We denote  $\theta = \bar{X}$  the canonical root of  $f$  in  $K$ . A  $nf$  structure contains a maximal order and allows operations on elements and ideals. A  $bnf$  adds class group and units. A  $bnr$  is attached to ray class groups and class field theory. A  $rnf$  is attached to relative extensions  $L/K$ .

init number field structure  $nf$  **nfinit**( $f, \{flag\}$ )  
  known integer basis  $B$  **nfinit**( $[f, B]$ )  
  order maximal at  $vp = [p_1, \dots, p_k]$  **nfinit**( $[f, vp]$ )  
  order maximal at all  $p \leq P$  **nfinit**( $[f, P]$ )  
  certify maximal order **nfcertify**( $nf$ )

### nf members:

a monic  $F \in \mathbf{Z}[X]$  defining  $K$  **nf.pol**  
number of real/complex places **nf.r1/r2/sign**  
discriminant of  $nf$  **nf.disc**  
primes ramified in  $nf$  **nf.p**  
 $T_2$  matrix **nf.t2**  
complex roots of  $F$  **nf.roots**  
integral basis of  $\mathbf{Z}_K$  as powers of  $\theta$  **nf.zk**  
different/codifferent **nf.diff**, **nf.codiff**  
index  $[\mathbf{Z}_K : \mathbf{Z}[X]/(F)]$  **nf.index**  
recompute  $nf$  using current precision **nfnewprec**( $nf$ )  
init relative  $rnf$   $L = K[Y]/(g)$  **rnfinit**( $nf, g$ )  
init  $bnf$  structure **bnfinit**( $f, 1$ )

**bnf members:** same as  $nf$ , plus  
  underlying  $nf$  **bnf.nf**  
  class group, regulator **bnf.clgp**, **bnf.reg**  
  fundamental/torsion units **bnf.fu**, **bnf.tu**  
  add  $S$ -class group and units, yield  $bnfS$  **bnfsunit**( $bnf, S$ )  
  init class field structure  $bnr$  **bnrinit**( $bnf, m, \{flag\}$ )  
**bnr members:** same as  $bnf$ , plus  
  underlying  $bnf$  **bnr.bnf**  
  big ideal structure **bnr.bid**  
  modulus  $m$  **bnr.mod**  
  structure of  $(\mathbf{Z}_K/m)^*$  **bnr.zkst**

## Fields, subfields, embeddings

**Defining polynomials, embeddings**  
(some) number fields with Galois group  $G$  **nflist**( $G$ )  
... and  $|\text{disc}(K)| = N$  and  $s$  complex places **nflist**( $G, N, \{s\}$ )  
... and  $a \leq |\text{disc}(K)| \leq b$  **nflist**( $G, [a, b], \{s\}$ )  
smallest poly defining  $f = 0$  (slow) **polredabs**( $f, \{flag\}$ )  
small poly defining  $f = 0$  (fast) **polredbest**( $f, \{flag\}$ )  
monic integral  $g = Cf(x/L)$  **poltomonic**( $f, \{\&L\}$ )  
random Tschirnhausen transform of  $f$  **poltschirnhaus**( $f$ )  
**Q**[ $t$ ]/( $f$ )  $\subset$  **Q**[ $t$ ]/( $g$ ) ? Isomorphic? **nfisincl**( $f, g$ ), **nfisisom**  
reverse polmod  $a = A(t) \bmod T(t)$  **modreverse**( $a$ )  
compositum of **Q**[ $t$ ]/( $f$ ), **Q**[ $t$ ]/( $g$ ) **polcompositum**( $f, g, \{flag\}$ )  
compositum of  $K[t]/(f)$ ,  $K[t]/(g)$  **nfcompositum**( $nf, f, g, \{flag\}$ )  
splitting field of  $K$  (degree divides  $d$ ) **nfsplitting**( $nf, \{d\}$ )  
signs of real embeddings of  $x$  **nfeltsign**( $nf, x, \{pl\}$ )  
complex embeddings of  $x$  **nfeltembed**( $nf, x, \{pl\}$ )  
 $T \in K[t]$ , # of real roots of  $\sigma(T) \in R[t]$  **nfpolsturm**( $nf, T, \{pl\}$ )

### Subfields, polynomial factorization

subfields (of degree  $d$ ) of  $nf$  **nfsubfields**( $nf, \{d\}$ )  
maximal subfields of  $nf$  **nfsubfieldsmax**( $nf$ )  
maximal CM subfield of  $nf$  **nfsubfieldscm**( $nf$ )  
 $K_d \subset \mathbf{Q}(\zeta_n)$ , using Gaussian periods **polsubcyclo**( $n, d, \{v\}$ )  
... using class field theory **polsubcyclofast**( $n, d$ )  
roots of unity in  $nf$  **nfrootsof1**( $nf$ )  
roots of  $g$  belonging to  $nf$  **nfroots**( $nf, g$ )  
factor  $g$  in  $nf$  **nfactor**( $nf, g$ )

### Linear and algebraic relations

poly of degree  $\leq k$  with root  $x \in \mathbf{C}$  or  $\mathbf{Q}_p$  **algdep**( $x, k$ )  
alg. dep. with pol. coeffs for series  $s$  **seralgdep**( $s, x, y$ )  
diff. dep. with pol. coeffs for series  $s$  **serdiffdep**( $s, x, y$ )  
small linear rel. on coords of vector  $x$  **lindep**( $x$ )

## Basic Number Field Arithmetic (nf)

Number field elements are **t\_INT**, **t\_FRAC**, **t\_POL**, **t\_POLMOD**, or **t\_COL**  
(on integral basis  $nf.zk$ ).

### Basic operations

$x + y$  **nfeltadd**( $nf, x, y$ )  
 $x \times y$  **nfeltmul**( $nf, x, y$ )  
 $x^n$ ,  $n \in \mathbf{Z}$  **nfeltpow**( $nf, x, n$ )  
 $x/y$  **nfeltdiv**( $nf, x, y$ )  
 $q = x \setminus y := \text{round}(x/y)$  **nfeltdiveuc**( $nf, x, y$ )  
 $r = x \setminus y := x - (x \setminus y)y$  **nfeltmod**( $nf, x, y$ )  
...  $[q, r]$  as above **nfeltdivrem**( $nf, x, y$ )  
reduce  $x$  modulo ideal  $A$  **nfeltreduce**( $nf, x, A$ )  
absolute trace  $\text{Tr}_{K/\mathbf{Q}}(x)$  **nfelttrace**( $nf, x$ )  
absolute norm  $N_{K/\mathbf{Q}}(x)$  **nfeltnorm**( $nf, x$ )

is  $x$  a square? **nfeltissquare**( $nf, x, \{\&y\}$ )  
... an  $n$ -th power? **nfeltispower**( $nf, x, n, \{\&y\}$ )

**Multiplicative structure of  $K^*$ ;  $K^*/(K^*)^n$**   
valuation  $v_{\mathfrak{p}}(x)$  **nfeltval**( $nf, x, \mathfrak{p}$ )  
... write  $x = \pi^{v_{\mathfrak{p}}(x)}y$  **nfeltval**( $nf, x, \mathfrak{p}, \&y$ )  
quadratic Hilbert symbol (at  $\mathfrak{p}$ ) **nfhilbert**( $nf, a, b, \{\mathfrak{p}\}$ )  
 $b$  such that  $xb^n = v$  is small **idealredmodpower**( $nf, x, n$ )

### Maximal order and discriminant

integral basis of field **Q**[ $x$ ]/( $f$ ) **nfbasis**( $f$ )  
field discriminant of **Q**[ $x$ ]/( $f$ ) **nfdisc**( $f$ )  
... and factorization **nfdiscfactors**( $f$ )  
express  $x$  on integer basis **nfalgtobasis**( $nf, x$ )  
express element  $x$  as a polmod **nfbasistoalg**( $nf, x$ )

### Hecke Grossencharacters

Let  $K$  be a number field and  $m$  a modulus. A **gchar** structure describes the group of Hecke Grossencharacters of  $K$  of modulus  $m$  and allows computations with these characters. A character  $\chi$  is described by its components modulo  $gc.cyc$ .

init **gchar** structure  $gc$  for modulus  $m$  **gcharinit**( $bnf, m, \{cm\}$ )  
**gc members:**  
  underlying  $bnf$  **gc.bnf**  
  modulus **gc.mod**  
  elementary divisors (including 0s) **gc.cyc**  
recompute  $gc$  using current precision **gcharnewprec**( $gc$ )  
evaluate Hecke character  $chi$  at ideal  $id$  **gchareval**( $gc, chi, id$ )  
exponent column of  $id$  in  $\mathbf{R}^n$  **gcharideallog**( $gc, id$ )  
log representation of ideal  $id$  **gcharlog**( $gc, id$ )  
... of character  $\chi$  **gcharduallog**( $gc, chi$ )  
exponent vector of  $\chi$  in  $\mathbf{R}^n$  **gcharparameters**( $gc, chi$ )  
conductor of  $\chi$  **gcharconductor**( $gc, chi$ )  
L-function of  $\chi$  **lfuncreate**( $[gc, chi]$ )  
local component  $\chi_v$  of  $\chi$  **gcharlocal**( $gc, chi, v$ )  
 $\chi$  s.t.  $\chi_v \approx Lchiv[i]$  for  $v = Lv[i]$  **gcharidentify**( $gc, Lv, Lchiv$ )  
basis of group of algebraic characters **gcharalgebraic**( $gc$ )  
is  $\chi$  algebraic? **gcharisalgebraic**( $gc, chi$ )

### Dedekind Zeta Function $\zeta_K$ , Hecke $L$ series

$R = [c, w, h]$  in initialization means we restrict  $s \in \mathbf{C}$  to domain  $|\Re(s) - c| < w$ ,  $|\Im(s)| < h$ ;  $R = [w, h]$  encodes  $[1/2, w, h]$  and  $[h]$  encodes  $R = [1/2, 0, h]$  (critical line up to height  $h$ ).

$\zeta_K$  as Dirichlet series,  $N(I) \leq b$  **dirzetak**( $nf, b$ )  
init  $\zeta_K^{(k)}(s)$  for  $k \leq n$  **L = lfunitinit**( $bnf, R, \{n = 0\}$ )  
compute  $\zeta_K(s)$  ( $n$ -th derivative) **lfun**( $L, s, \{n = 0\}$ )  
compute  $\Lambda_K(s)$  ( $n$ -th derivative) **lfunlambda**( $L, s, \{n = 0\}$ )

init  $L_K^{(k)}(s, \chi)$  for  $k \leq n$  **L = lfunitinit**( $[bnr, chi], R, \{n = 0\}$ )  
compute  $L_K(s, \chi)$  ( $n$ -th derivative) **lfun**( $L, s, \{n\}$ )  
Artin root number of  $K$  **bnrrootnumber**( $bnr, chi, \{flag\}$ )  
 $L(1, \chi)$ , for all  $\chi$  trivial on  $H$  **bnrL1**( $bnr, \{H\}, \{flag\}$ )

## Class Groups & Units (bnf, bnr)

Class field theory data  $a_1, \{a_2\}$  is usually  $bnr$  (ray class field),  $bnr, H$  (congruence subgroup) or  $bnr, \chi$  (character on **bnr.clgp**). Any of these define a unique abelian extension of  $K$ .  
units /  $S$ -units **bnfunits**( $bnf, \{S\}$ )  
remove GRH assumption from  $bnf$  **bnfcertify**( $bnf$ )

expo. of ideal $x$ on class gp	<code>bnfisprincipal(<i>bnf</i>, <math>x</math>, {<i>flag</i>})</code>
... on ray class gp	<code>bnrisprincipal(<i>bnr</i>, <math>x</math>, {<i>flag</i>})</code>
expo. of $x$ on fund. units	<code>bnfisunit(<i>bnf</i>, <math>x</math>)</code>
... on $S$ -units, $U$ is <code>bnfunits(<i>bnf</i>, <math>S</math>)</code>	<code>bnfisunit(<i>bnfs</i>, <math>x</math>, <math>U</math>)</code>
signs of real embeddings of <i>bnf</i> .fu	<code>bnfsignunit(<i>bnf</i>)</code>
narrow class group	<code>bnfnarrow(<i>bnf</i>)</code>

### Class Field Theory

ray class number for modulus $m$	<code>bnrclassno(<i>bnf</i>, <math>m</math>)</code>
discriminant of class field	<code>bnrdisc(<math>a_1</math>, {<math>a_2</math>})</code>
ray class numbers, $l$ list of moduli	<code>bnrclassnolist(<i>bnf</i>, <math>l</math>)</code>
discriminants of class fields	<code>bnrdisclist(<i>bnf</i>, <math>l</math>, {<i>arch</i>}, {<i>flag</i>})</code>
decode output from <code>bnrdisclist</code>	<code>bnfdecodemodule(<i>nf</i>, <math>fa</math>)</code>
is modulus the conductor?	<code>bnrisconductor(<math>a_1</math>, {<math>a_2</math>})</code>
is class field ( <i>bnr</i> , $H$ ) Galois over $K^G$	<code>bnrisgalois(<i>bnr</i>, <math>G</math>, <math>H</math>)</code>
action of automorphism on <code>bnr.gen</code>	<code>bnrgaloismatrix(<i>bnr</i>, <math>aut</math>)</code>
apply <code>bnrgaloismatrix</code> $M$ to $H$	<code>bnrgaloisapply(<i>bnr</i>, <math>M</math>, <math>H</math>)</code>
characters on <code>bnr.clgp</code> s.t. $\chi(g_i) = e(v_i)$	<code>bnrchar(<i>bnr</i>, <math>g</math>, {<math>v</math>})</code>
conductor of character $\chi$	<code>bnrconductor(<i>bnr</i>, <math>chi</math>)</code>
conductor of extension	<code>bnrconductor(<math>a_1</math>, {<math>a_2</math>}, {<i>flag</i>})</code>
conductor of extension $K[Y]/(g)$	<code>rnfconductor(<i>bnf</i>, <math>g</math>)</code>
canonical projection $\text{Cl}_F \rightarrow \text{Cl}_f$ , $f \mid F$	<code>bnrmap</code>
Artin group of extension $K[Y]/(g)$	<code>rnfnormgroup(<i>bnr</i>, <math>g</math>)</code>
subgroups of <i>bnr</i> , index $\leq b$	<code>subgrouplist(<i>bnr</i>, <math>b</math>, {<i>flag</i>})</code>
compositum as [ <code>bnr</code> , <code>H</code> ]	<code>bnrcompositum([<i>bnr</i>1, <math>H</math>1], [<i>bnr</i>2, <math>H</math>2])</code>
class field defined by $H \subset \text{Cl}_f$	<code>bnrclassfield(<i>bnr</i>, <math>H</math>)</code>
... low level equivalent, prime degree	<code>rnfkummer(<i>bnr</i>, <math>H</math>)</code>
same, using Stark units (real field)	<code>bnrstark(<i>bnr</i>, <math>sub</math>, {<i>flag</i>})</code>
is $a$ an $n$ -th power in $K_v$ ?	<code>nfislocalpower(<i>nf</i>, <math>v</math>, <math>a</math>, <math>n</math>)</code>
cyclic $L/K$ satisf. local conditions	<code>nfgrunwaldwang(<i>nf</i>, <math>P</math>, <math>D</math>, <math>pl</math>)</code>

### Cyclotomic and Abelian fields theory

An Abelian field  $F$  given by a subgroup  $H \subset (Z/fZ)^*$  is described by an argument  $F$ , e.g.  $f$  (for  $H = 1$ , i.e.  $Q(\zeta_f)$ ) or  $[G, H]$ , where  $G$  is `idealstar( $f$ , 1)`, or a minimal polynomial.

minus class number $h^-(F)$	<code>subcyclohminus(<math>F</math>)</code>
... $p$ -part	<code>subcyclohminus(<math>F</math>, <math>p</math>)</code>
minus part of Iwasawa polynomials	<code>subcycloiwasawa(<math>F</math>, <math>p</math>)</code>
$p$ -Sylow of $\text{Cl}(F)$	<code>subcyclopclgp(<math>F</math>, <math>p</math>)</code>

### Logarithmic class group

logarithmic $\ell$ -class group	<code>bnflog(<i>bnf</i>, <math>\ell</math>)</code>
$[\tilde{e}(F_v/Q_p), \tilde{f}(F_v/Q_p)]$	<code>bnflogef(<i>bnf</i>, <math>pr</math>)</code>
$\exp \deg_F(A)$	<code>bnflogdegree(<i>bnf</i>, <math>A</math>, <math>\ell</math>)</code>
is $\ell$ -extension $L/K$ locally cyclotomic	<code>rnfislocalcyclo(<i>rnf</i>)</code>

**Ideals:** elements, primes, or matrix of generators in HNF

is $id$ an ideal in $nf$ ?	<code>nfisideal(<i>nf</i>, <math>id</math>)</code>
is $x$ principal in <i>bnf</i> ?	<code>bnfisprincipal(<i>bnf</i>, <math>x</math>)</code>
give $[a, b]$ , s.t. $a\mathbf{Z}_K + b\mathbf{Z}_K = x$	<code>idealtwoelt(<i>nf</i>, <math>x</math>, {<math>a</math>})</code>
put ideal $a$ ( $a\mathbf{Z}_K + b\mathbf{Z}_K$ ) in HNF form	<code>idealhnf(<i>nf</i>, <math>a</math>, {<math>b</math>})</code>
norm of ideal $x$	<code>idealnrm(<i>nf</i>, <math>x</math>)</code>
minimum of ideal $x$ (direction $v$ )	<code>idealmin(<i>nf</i>, <math>x</math>, <math>v</math>)</code>
LLL-reduce the ideal $x$ (direction $v$ )	<code>idealred(<i>nf</i>, <math>x</math>, {<math>v</math>})</code>

### Ideal Operations

add ideals $x$ and $y$	<code>idealadd(<i>nf</i>, <math>x</math>, <math>y</math>)</code>
multiply ideals $x$ and $y$	<code>idealmul(<i>nf</i>, <math>x</math>, <math>y</math>, {<i>flag</i>})</code>
intersection of ideal $x$ with $Q$	<code>idealdown(<i>nf</i>, <math>x</math>)</code>
intersection of ideals $x$ and $y$	<code>idealintersect(<i>nf</i>, <math>x</math>, <math>y</math>, {<i>flag</i>})</code>
$n$ -th power of ideal $x$	<code>idealpow(<i>nf</i>, <math>x</math>, <math>n</math>, {<i>flag</i>})</code>
inverse of ideal $x$	<code>idealinv(<i>nf</i>, <math>x</math>)</code>
divide ideal $x$ by $y$	<code>idealdiv(<i>nf</i>, <math>x</math>, <math>y</math>, {<i>flag</i>})</code>

# Algebraic Number Theory

(PARI-GP version 2.15.3)

Find $(a, b) \in x \times y$ , $a + b = 1$	<code>idealaddtoone(<i>nf</i>, <math>x</math>, {<math>y</math>})</code>
coprime integral $A, B$ such that $x = A/B$	<code>idealnumden(<i>nf</i>, <math>x</math>)</code>

### Primes and Multiplicative Structure

check whether $x$ is a maximal ideal	<code>idealismaximal(<i>nf</i>, <math>x</math>)</code>
factor ideal $x$ in $\mathbf{Z}_K$	<code>idealfactor(<i>nf</i>, <math>x</math>)</code>
expand ideal factorization in $K$	<code>idealfactorback(<i>nf</i>, <math>f</math>, {<math>e</math>})</code>
is ideal $A$ an $n$ -th power ?	<code>idealispower(<i>nf</i>, <math>A</math>, <math>n</math>)</code>
expand elt factorization in $K$	<code>nffactorback(<i>nf</i>, <math>f</math>, {<math>e</math>})</code>
decomposition of prime $p$ in $\mathbf{Z}_K$	<code>idealprimedec(<i>nf</i>, <math>p</math>)</code>
valuation of $x$ at prime ideal $pr$	<code>idealval(<i>nf</i>, <math>x</math>, <math>pr</math>)</code>
weak approximation theorem in $nf$	<code>idealchinese(<i>nf</i>, <math>x</math>, <math>y</math>)</code>
$a \in K$ , s.t. $v_{\mathfrak{p}}(a) = v_{\mathfrak{p}}(x)$ if $v_{\mathfrak{p}}(x) \neq 0$	<code>idealappr(<i>nf</i>, <math>x</math>)</code>
$a \in K$ such that $(a \cdot x, y) = 1$	<code>idealcoprime(<i>nf</i>, <math>x</math>, <math>y</math>)</code>
give $bid$ =structure of $(\mathbf{Z}_K/id)^*$	<code>idealstar(<i>nf</i>, <math>id</math>, {<i>flag</i>})</code>
structure of $(1 + \mathfrak{p})/(1 + \mathfrak{p}^k)$	<code>idealprincipalunits(<i>nf</i>, <math>pr</math>, <math>k</math>)</code>
discrete log of $x$ in $(\mathbf{Z}_K/bid)^*$	<code>ideallog(<i>nf</i>, <math>x</math>, <math>bid</math>)</code>
idealstar of all ideals of norm $\leq b$	<code>ideallist(<i>nf</i>, <math>b</math>, {<i>flag</i>})</code>
add Archimedean places	<code>ideallistarch(<i>nf</i>, <math>b</math>, {<math>ar</math>}, {<i>flag</i>})</code>
init <code>modpr</code> structure	<code>nfmodprinit(<i>nf</i>, <math>pr</math>, {<math>v</math>})</code>
project $t$ to $\mathbf{Z}_K/pr$	<code>nfmodpr(<i>nf</i>, <math>t</math>, <i>modpr</i>)</code>
lift from $\mathbf{Z}_K/pr$	<code>nfmodprlift(<i>nf</i>, <math>t</math>, <i>modpr</i>)</code>

### Galois theory over Q

conjugates of a root $\theta$ of <i>nf</i>	<code>nfgaloisconj(<i>nf</i>, {<i>flag</i>})</code>
apply Galois automorphism $s$ to $x$	<code>nfgaloisapply(<i>nf</i>, <math>s</math>, <math>x</math>)</code>
Galois group of field $\mathbf{Q}[x]/(f)$	<code>polgalois(<math>f</math>)</code>
resolvent field of $\mathbf{Q}[x]/(f)$	<code>nfresolvent(<math>f</math>)</code>
initializes a Galois group structure $G$	<code>galoisinit(<i>pol</i>, {<i>den</i>})</code>
... for the splitting field of <i>pol</i>	<code>galoisplittinginit(<i>pol</i>, {<math>d</math>})</code>
character table of $G$	<code>galoischartable(<math>G</math>)</code>
conjugacy classes of $G$	<code>galoisconjclasses(<math>G</math>)</code>
$\det(1 - \rho(g)T)$ , $\chi$ character of $\rho$	<code>galoischarpoly(<math>G</math>, <math>\chi</math>, {<math>o</math>})</code>
$\det(\rho(g))$ , $\chi$ character of $\rho$	<code>galoischarDET(<math>G</math>, <math>\chi</math>, {<math>o</math>})</code>
action of $p$ in <code>nfgaloisconj</code> form	<code>galoispermtpol(<math>G</math>, {<math>p</math>})</code>
identify as abstract group	<code>galoisidentify(<math>G</math>)</code>
export a group for GAP/MAGMA	<code>galoisexport(<math>G</math>, {<i>flag</i>})</code>
subgroups of the Galois group $G$	<code>galoissubgroups(<math>G</math>)</code>
is subgroup $H$ normal?	<code>galoisisnormal(<math>G</math>, <math>H</math>)</code>
subfields from subgroups	<code>galoissubfields(<math>G</math>, {<i>flag</i>}, {<math>v</math>})</code>
fixed field	<code>galoisfixedfield(<math>G</math>, <i>perm</i>, {<i>flag</i>}, {<math>v</math>})</code>
Frobenius at maximal ideal $P$	<code>idealfrobenius(<i>nf</i>, <math>G</math>, <math>P</math>)</code>
ramification groups at $P$	<code>idealramgroups(<i>nf</i>, <math>G</math>, <math>P</math>)</code>
is $G$ abelian?	<code>galoisisabelian(<math>G</math>, {<i>flag</i>})</code>
abelian number fields/ $\mathbf{Q}$	<code>galoissubcyclo(<math>N</math>, <math>H</math>, {<i>flag</i>}, {<math>v</math>})</code>

### The galpol package

query the package: polynomial	<code>galoisgetpol(a,b,{s})</code>
...: permutation group	<code>galoisgetgroup(a,b)</code>
...: group description	<code>galoisgetname(a,b)</code>

### Relative Number Fields (rnf)

Extension  $L/K$  is defined by  $T \in K[x]$ .

absolute equation of $L$	<code>rnfequation(<i>nf</i>, <math>T</math>, {<i>flag</i>})</code>
is $L/K$ abelian?	<code>rnfisabelian(<i>nf</i>, <math>T</math>)</code>
relative <code>nfalgtobasis</code>	<code>rnfalgtobasis(<i>rnf</i>, <math>x</math>)</code>
relative <code>nfbasistoalg</code>	<code>rnfbasistoalg(<i>rnf</i>, <math>x</math>)</code>
relative <code>idealhnf</code>	<code>rnfidealhnf(<i>rnf</i>, <math>x</math>)</code>
relative <code>idealmul</code>	<code>rnfidealmul(<i>rnf</i>, <math>x</math>, <math>y</math>)</code>
relative <code>idealtwoelt</code>	<code>rnfidealtwoelt(<i>rnf</i>, <math>x</math>)</code>

### Lifts and Push-downs

absolute $\rightarrow$ relative representation for $x$	<code>rnfelstabstorel(<i>rnf</i>, <math>x</math>)</code>
relative $\rightarrow$ absolute representation for $x$	<code>rnfeltrretoabs(<i>rnf</i>, <math>x</math>)</code>
lift $x$ to the relative field	<code>rnfeltup(<i>rnf</i>, <math>x</math>)</code>
push $x$ down to the base field	<code>rnfeltdown(<i>rnf</i>, <math>x</math>)</code>
idem for $x$ ideal: ( <code>rnfideal</code> ) <code>reltoabs</code> , <code>abstorel</code> , <code>up</code> , <code>down</code>	

### Norms and Trace

relative norm of element $x \in L$	<code>rnfeltnrm(<i>rnf</i>, <math>x</math>)</code>
relative trace of element $x \in L$	<code>rnfeltrtrace(<i>rnf</i>, <math>x</math>)</code>
absolute norm of ideal $x$	<code>rnfidealnrmabs(<i>rnf</i>, <math>x</math>)</code>
relative norm of ideal $x$	<code>rnfidealnrmrel(<i>rnf</i>, <math>x</math>)</code>
solutions of $N_{K/\mathbf{Q}}(y) = x \in \mathbf{Z}$	<code>bnfisintnrm(<i>bnf</i>, <math>x</math>)</code>
is $x \in \mathbf{Q}$ a norm from $K$ ?	<code>bnfisnrm(<i>bnf</i>, <math>x</math>, {<i>flag</i>})</code>
initialize $T$ for norm eq. solver	<code>rnfisnorminit(<math>K</math>, <i>pol</i>, {<i>flag</i>})</code>
is $a \in K$ a norm from $L$ ?	<code>rnfisnrm(<math>T</math>, <math>a</math>, {<i>flag</i>})</code>
initialize $t$ for Thue equation solver	<code>thueinit(<math>f</math>)</code>
solve Thue equation $f(x, y) = a$	<code>thue(<math>t</math>, <math>a</math>, {<i>sol</i>})</code>
characteristic poly. of $a \bmod T$	<code>rnfcharpoly(<i>nf</i>, <math>T</math>, <math>a</math>, {<math>v</math>})</code>

### Factorization

factor ideal $x$ in $L$	<code>rnfidealfactor(<i>rnf</i>, <math>x</math>)</code>
$[S, T]: T_{i,j} \mid S_i$ ; $S$ primes of $K$ above $p$	<code>rnfidealprimedec(<i>rnf</i>, <math>p</math>)</code>

### Maximal order $\mathbf{Z}_L$ as a $\mathbf{Z}_K$ -module

relative <code>polredbest</code>	<code>rnfpolredbest(<i>nf</i>, <math>T</math>)</code>
relative <code>polredabs</code>	<code>rnfpolredabs(<i>nf</i>, <math>T</math>)</code>
relative Dedekind criterion, prime $pr$	<code>rnfdedekind(<i>nf</i>, <math>T</math>, <math>pr</math>)</code>
discriminant of relative extension	<code>rnfdisc(<i>nf</i>, <math>T</math>)</code>
pseudo-basis of $\mathbf{Z}_L$	<code>rnfpseudobasis(<i>nf</i>, <math>T</math>)</code>

**General  $\mathbf{Z}_K$ -modules:**  $M = [\text{matrix, vec. of ideals}] \subset L$

relative HNF / SNF	<code>nfhnf(<i>nf</i>, <math>M</math>), nfsnf</code>
multiple of det $M$	<code>nfDETint(<i>nf</i>, <math>M</math>)</code>
HNF of $M$ where $d = nfDETint(M)$	<code>nfhnfmod(<math>x</math>, <math>d</math>)</code>
reduced basis for $M$	<code>rnfilllgram(<i>nf</i>, <math>T</math>, <math>M</math>)</code>
determinant of pseudo-matrix $M$	<code>rnfdet(<i>nf</i>, <math>M</math>)</code>
Steinitz class of $M$	<code>rnfstEinitz(<i>nf</i>, <math>M</math>)</code>
$\mathbf{Z}_K$ -basis of $M$ if $\mathbf{Z}_K$ -free, or 0	<code>rnfhnfBasis(<i>bnf</i>, <math>M</math>)</code>
$n$ -basis of $M$ , or $(n + 1)$ -generating set	<code>rnfbasis(<i>bnf</i>, <math>M</math>)</code>
is $M$ a free $\mathbf{Z}_K$ -module?	<code>rnfisfree(<i>bnf</i>, <math>M</math>)</code>

Associative Algebras

*A* is a general associative algebra given by a multiplication table *mt* (over **Q** or **F<sub>p</sub>**); represented by *al* from `algtableinit`.  
create *al* from *mt* (over **F<sub>p</sub>**)                    `algtableinit(mt, {p = 0})`  
group algebra **Q**[*G*] (or **F<sub>p</sub>**[*G*])                    `alggroup(G, {p = 0})`  
center of group algebra                    `alggrouppcenter(G, {p = 0})`  
**Properties**  
is (*mt*, *p*) OK for `algtableinit`?                    `algisassociative(mt, {p = 0})`  
multiplication table *mt*                    `algmultable(al)`  
dimension of *A* over prime subfield                    `algdim(al)`  
characteristic of *A*                    `algchar(al)`  
is *A* commutative?                    `algiscommutative(al)`  
is *A* simple?                    `algissimple(al)`  
is *A* semi-simple?                    `algissemisimple(al)`  
center of *A*                    `algcenter(al)`  
Jacobson radical of *A*                    `algradical(al)`  
radical *J* and simple factors of *A*/*J*                    `algsimpledec(al)`  
**Operations on algebras**  
create *A*/*I*, *I* two-sided ideal                    `algquotient(al, I)`  
create *A*<sub>1</sub> ⊗ *A*<sub>2</sub>                    `algtensor(al1, al2)`  
create subalgebra from basis *B*                    `algsubalg(al, B)`  
quotients by ortho. central idempotents *e*                    `algcentralproj(al, e)`  
isomorphic alg. with integral mult. table                    `algmakeintegral(mt)`  
prime subalgebra of semi-simple *A* over **F<sub>p</sub>**                    `algprimesubalg(al)`  
find isomorphism *A* ≅ *M<sub>d</sub>*(**F<sub>q</sub>**)                    `algsplit(al)`  
**Operations on lattices in algebras**  
lattice generated by cols. of *M*                    `alglathnf(al, M)`  
... by the products *xy*, *x* ∈ *lat1*, *y* ∈ *lat2*                    `alglatmul(al, lat1, lat2)`  
sum *lat1* + *lat2* of the lattices                    `alglatadd(al, lat1, lat2)`  
intersection *lat1* ∩ *lat2*                    `alglatinter(al, lat1, lat2)`  
test *lat1* ⊂ *lat2*                    `alglatsubset(al, lat1, lat2)`  
generalized index (*lat2* : *lat1*)                    `alglatindex(al, lat1, lat2)`  
{*x* ∈ *al* | *x* · *lat1* ⊂ *lat2*}                    `alglatlefttransporter(al, lat1, lat2)`  
{*x* ∈ *al* | *lat1* · *x* ⊂ *lat2*}                    `alglatrighttransporter(al, lat1, lat2)`  
test *x* ∈ *lat* (set *c* = coord. of *x*)                    `alglatcontains(al, lat, x, {&c})`  
element of *lat* with coordinates *c*                    `alglatelement(al, lat, c)`  
**Operations on elements**  
*a* + *b*, *a* − *b*, −*a*                    `algadd(al, a, b), algsub, algneg`  
*a* × *b*, *a*<sup>2</sup>                    `algmul(al, a, b), algsq`  
*a<sup>n</sup>*, *a*<sup>−1</sup>                    `algpow(al, a, n), alginv`  
is *x* invertible ? (then set *z* = *x*<sup>−1</sup>)                    `alginv(al, x, {&z})`  
find *z* such that *x* × *z* = *y*                    `algdivl(al, x, y)`  
find *z* such that *z* × *x* = *y*                    `algdivr(al, x, y)`  
does *z* s.t. *x* × *z* = *y* exist? (set it)                    `algisdivl(al, x, y, {&z})`  
matrix of *v* ↦ *x* · *v*                    `algtomatrix(al, x)`  
absolute norm                    `algnorm(al, x)`  
absolute trace                    `algtrace(al, x)`  
absolute char. polynomial                    `algcharpoly(al, x)`  
given *a* ∈ *A* and polynomial *T*, return *T*(*a*)                    `algpoleval(al, T, a)`  
random element in a box                    `algrandom(al, b)`

Central Simple Algebras

*A* is a central simple algebra over a number field *K*; represented by *al* from `algininit`; *K* is given by a *nf* structure.  
create CSA from data                    `algininit(B, C, {v}, {maxord = 1})`  
multiplication table over *K*                    *B* = *K*, *C* = *mt*  
cyclic algebra (*L*/*K*, *σ*, *b*)                    *B* = *rmf*, *C* = [*sigma*, *b*]  
quaternion algebra (*a*, *b*)<sub>*K*</sub>                    *B* = *K*, *C* = [*a*, *b*]  
matrix algebra *M<sub>d</sub>*(*K*)                    *B* = *K*, *C* = *d*  
local Hasse invariants over *K*                    *B* = *K*, *C* = [*d*, [*PR*, *HF*], *HI*]

Properties

type of *al* (*mt*, CSA)                    `algtype(al)`  
dimension of *A* over **Q**                    `algdim(al, 1)`  
dimension of *al* over its center *K*                    `algdim(al)`  
degree of *A* (= √dim<sub>*K*</sub> *A*)                    `algdegree(al)`  
*al* a cyclic algebra (*L*/*K*, *σ*, *b*); return *σ*                    `algaut(al)`  
...return *b*                    `algb(al)`  
...return *L*/*K*, as an *rmf*                    `algsplittingfield(al)`  
split *A* over an extension of *K*                    `algsplittingdata(al)`  
splitting field of *A* as an *rmf* over center                    `algsplittingfield(al)`  
multiplication table over center                    `algrelmultable(al)`  
places of *K* at which *A* ramifies                    `algramifiedplaces(al)`  
Hasse invariants at finite places of *K*                    `alghassef(al)`  
Hasse invariants at infinite places of *K*                    `alghassei(al)`  
Hasse invariant at place *v*                    `alghasse(al, v)`  
index of *A* over *K* (at place *v*)                    `algindex(al, {v})`  
is *al* a division algebra? (at place *v*)                    `algisdivision(al, {v})`  
is *A* ramified? (at place *v*)                    `algisramified(al, {v})`  
is *A* split? (at place *v*)                    `algisplit(al, {v})`

Operations on elements

reduced norm                    `algnorm(al, x)`  
reduced trace                    `algtrace(al, x)`  
reduced char. polynomial                    `algcharpoly(al, x)`  
express *x* on integral basis                    `algalgtobasis(al, x)`  
convert *x* to algebraic form                    `algbasistoalg(al, x)`  
map *x* ∈ *A* to *M<sub>d</sub>*(*L*), *L* split. field                    `algtomatrix(al, x)`

Orders

**Z**-basis of order *O*<sub>0</sub>                    `algbasis(al)`  
discriminant of order *O*<sub>0</sub>                    `algdisc(al)`  
**Z**-basis of natural order in terms *O*<sub>0</sub>'s basis                    `alginvbasis(al)`