1 Orderings

Prefixes:

- `o` = order
- `r` = removed
- `zk` = ring of integers of `nf`
- *nothing* = finite set

1.1 Regular $\mathfrak{p}$-orderings

- `GEN pord(GEN nf, GEN pr, GEN S, long first = 1, long trunc = -1, GEN* ex = NULL, GEN* inv = NULL)`

  Return in a vector beginning by `S[first]` the first `trunc` elements of a `pr`-ordering of the set `S`. If `trunc = -1`, it is set to `#S`. Set to `*ex` the extraction small vector and to `*inv` the invariant vector as it would be returned by `pord_e(nf, pr, S, trunc)`

- `GEN pord_e(GEN nf, GEN pr, GEN S, long trunc = -1)`

  Return in a vector the first `trunc-1` invariant exponents of any `pr`-ordering of the set `S`. If `trunc = -1`, it is set to `#S`.

- `int ispord(GEN nf, GEN pr, GEN S, long trunc = -1, GEN *i = NULL)`

  Return 1 if the sequence of the first `trunc` elements of `S` is the beginning of a `pr`-ordering of the set `S`, 0 otherwise. If `trunc = -1`, it is set to `#S`. If 0 is returned, set to `*i` the index of the first element of `S` responsible for failure.

- `GEN pord_get_e(GEN nf, GEN pr, GEN po, long trunc = -1)`

  Return in a vector the first `trunc - 1` invariant exponents of the `pr`-ordering `po`. If `trunc = -1`, it is set to `#po`.

- `GEN zkpord(GEN nf, GEN pr, long n)`

  Return a `pr`-ordering of length `n` of `nf`.

- `GEN iszkpord(GEN nf GEN pr, GEN S)`

  Return 1 if the sequence `S` is a `pr`-ordering of length `#S` of `nf`, 0 otherwise.

- `GEN zkpord_e(GEN nf, GEN pr, long n)`

  Return the `n` first invariant exponents of any `pr`-ordering of `nf`.

1.2 r-removed $\mathfrak{p}$-orderings

- `GEN rpord(GEN nf, GEN pr, GEN S, long r, long trunc = -1, GEN* ex = NULL, GEN* inv = NULL)`
  Return in a vector the first `trunc` elements of a `r`-removed `pr`-ordering of the set `S`. If `trunc = -1`, it is set to `#S*(r+1)`. Set to `*ex` the extraction small vector and to `*inv` the invariant vector as it would be returned by `rpord_e(nf, pr, S, r, trunc)`.

- `GEN rpord_e(GEN nf, GEN pr, GEN S, long r, long trunc = -1)`

  Return in a vector the first `trunc-1` invariant exponents of any `r`-removed `pr`-ordering of the set `S`. If `trunc = -1`, it is set to `#S*(r+1)`.

- `GEN isrpord(GEN nf, GEN pr, GEN S, long r, long trunc = -1)`

  Return 1 if the sequence of the first `trunc` elements of `S` is the beginning of a `r`-removed `pr`-ordering of the set `S`, 0 otherwise. If `trunc = -1`, it is set to `#S*(r+1)`. Set to `*i` the index of the first element of `S` responsible for failure.

- `GEN rpord_get_e(GEN nf, GEN pr, GEN rpo, long r, long trunc = -1)`

  Return in a vector the first `trunc - 1` invariant exponents of the `r`-removed `pr`-ordering `rpo`. If `trunc = -1`, it is set to `#rpo`.

- `GEN zkrpord(GEN nf, GEN pr, long r, long n)`

  Return a `r`-removed `pr`-ordering of length `n` of `nf`.

- `GEN iszkrpord(GEN nf, GEN pr, long rpo, long r)`

  Return 1 if the sequence `S` is a `r`-removed `pr`-ordering of length `#S` of `nf`, 0 otherwise.

- `GEN zkrpord_e(GEN nf, GEN pr, long r, long n)`

  Return the `n` first invariant exponents of any `r`-removed `pr`-ordering of `nf`.

1.3 $\mathfrak{p}$-orderings of order h

- `GEN opord(GEN nf, GEN pr, GEN S, long h, long first, long trunc = -1, GEN* ex = NULL, GEN* inv = NULL)`

  Return in a vector beginning by `S[first]` the first `trunc` elements of a `pr`-ordering of order `h` of the set `S`. If `trunc = -1`, it is set to `#S`. Set to `*ex` the extraction small vector and to `*inv` the vector of invariants as it would be returned by `opord_e(nf,npr, S, r, trunc)`.

- `GEN opord_e(GEN nf, GEN pr, GEN S, long h, long trunc = -1)`

  Return in a vector the first `trunc-1` invariant exponents of any `pr`-ordering of order `h` of the set `S`. If `trunc = -1`, it is set to `#S`.

- `GEN isopord(GEN nf, GEN pr, GEN S, long h, long trunc = -1)`

  Return 1 if the sequence of the first `trunc` elements of `S` is the beginning of a `pr`-ordering of order `h` of the set `S`, 0 otherwise. If `trunc = -1`, it is set to `#S`. Set to `*i` the index of the first element of `S` responsible for failure.

- `GEN opord_get_e(GEN nf, GEN pr, GEN opo, long h, long trunc = -1)`

  Return in a vector the first `trunc - 1` invariant exponents of the `pr`-ordering of order `h` `opo`. If `trunc = -1`, it is set to `#opo`.

- `GEN zkopord(GEN nf, GEN pr, long h, long n)`

  Return a `pr`-ordering of order `h` of length `n` of `nf`.

- `GEN iszkopord(GEN nf, GEN pr, long opo, long h)`

  Return 1 if the sequence `S` is a `pr`-ordering of order `h` of length `#S` of `nf`, 0 otherwise.

- `GEN zkopord_e(GEN nf, GEN pr, long h, long n)`

  Return the `n` first invariant exponents of any `pr`-ordering of order `h` of `nf`.

2 Factorial ideals

Prefixes:

- `s` = finite set
- `zk` = the whole ring of integers of `nf`
- `q` = the whole ring of integers of the quadratic field `nf`
- `rem` = removed
- `mod` = modulus

2.1 Regular factorial ideals

- `GEN sfact(GEN nf, GEN S, long k)`

  Return the `k`-th factorial ideal of the set `S`.

- `GEN sfact_vec(GEN nf, GEN S, long n = -1)`

  Return in a vector the first `n` factorial ideals of the set of algebraic integers `S`. If `n=-1`, it is set to `#S-1`. Faster than building the vector by incremental calls to `sfact`.

- `GEN sfactnorm(GEN nf, GEN S, long k)`

  Return the norm of the `k`-th factorial ideal of the set `S`.

- `GEN sfactnorm_vec(GEN nf, GEN S, long n = -1)`

  Return in a vector the norms of the first `n` factorial ideals of the set `S`. If `n=-1`, it is set to `#S-1`. Faster than building the vector by incremental calls to `sfactnorm`.

- `GEN zkfact(GEN nf, long k)`

  Return the `k`-th factorial ideal of `nf`.

- `GEN zkfact_vec(GEN nf, long n)`

  Return in a vector the `n` first factorial ideals of `nf`. Faster than building the vector by incremental calls to `zkfact`.

- `GEN zkfactnorm(GEN nf, long k)`

  Return the norm of the `k`-th factorial ideal of `nf`. Faster than calling `idealnorm(nf, zkfact(nf, k))`.

- `GEN zkfactnorm_vec(GEN nf, long n)`

  Return in a vector the norm of the `n` first factorial ideals of `nf`. Faster than building the vector by incremental calls to `zkfactnorm`.

- `GEN qfact(GEN nf, long k)`

  Return the `k`-th factorial ideal of the quadratic number field `nf`. Faster than calling `zkfact`.

- `GEN qfact_vec(GEN nf, long n)`

  Return in a vector the first `n` factorial ideals of the quadratic number field `nf`.

- `GEN qfactnorm(GEN nf, long k)`

  Return the norm of the `k`-th factorial ideal of the quadratic number field `nf`. Faster than calling `idealnorm(nf,qfact(nf,k))` or `zkfactnorm`.

- `GEN qfactnorm_vec(GEN nf, long n)`

  Return in a vector the norms of the first `n` factorial ideals of the quadratic number field `nf`.

2.2 r-removed factorial ideals

- `GEN sremfact(GEN nf, GEN S, long r, long k)`

  Return the `k`-th `r`-removed factorial ideal of the set `S`.

- `GEN sremfact_vec(GEN nf, GEN S, long r, long n)`

  Return in a vector the `n` first `r`-removed factorial ideals of the set `S`. If `n = -1`, it is set to `((r+1)*#S)-1`. Faster than building the vector by incremental calls to `sremfact`.

- `GEN sremfactnorm(GEN nf, GEN S, long r, long k)`

  Return the norm of the `k`-th `r`-removed factorial ideal of the set `S`.

- `GEN sremfactnorm_vec(GEN nf, GEN S, long r, long n)`

  Return in a vector the norms of the n `first` `r`-removed factorial ideals of the set `S`. If `n = -1`, it is set to `((r+1)*#S)-1`. Faster than building the vector by incremental calls to `sremfactnorm`.

- `GEN zkremfact(GEN nf, long r, long k)`

  Return the `k`-th `r-removed` factorial ideal of `nf`.

- `GEN zkremfact_vec(GEN nf, long r, long n)`

  Return in a vector the `n` first `r`-removed factorial ideals of `nf`. Faster than building the vector by incremental calls to `zkremfact`.

- `GEN zkremfactnorm(GEN nf, long r, long k)`

  Return the norm of the `k`-th `r`-removed factorial ideal of `nf`.

- `GEN zkremfactnorm_vec(GEN nf, long r, long n)`

  Return in a vector the norms of the first `n` `r`-removed factorial ideals of `nf`. Faster than building the vector by incremental calls to `zkremfactnorm`.

2.3 Factorial ideals of modulus M

- `GEN sfactmod(GEN nf, GEN S, GEN M, long k)`

  Return the `k`-th factorial ideal of modulus `M` of the set `S`.

- `GEN sfactmod_vec(GEN nf, GEN S, GEN M, long n)`

  Return in a vector the `n` first factorial ideals of modulus `M` of the set `S`. If `n = -1`, it is set to `#S - 1`. Faster than building the vector by incremental calls to `sfactmod`.

- `GEN sfactmodnorm(GEN nf, GEN S, GEN M, long k)`

  Return the norm of the `k`-th factorial ideal of modulus `M` of the set `S`.

- `GEN sfactmodnorm_vec(GEN nf, GEN S, GEN M, long n)`

  Return in a vector the norms of the first `n` factorial ideals of modulus `M` of the set `S`. If `n = -1`, it is set to `#S - 1`. Faster than building the vector by incremental calls to `sfactmodnorm`.

- `GEN zkfactmod(GEN nf, GEN M, long k)`

  Return the `k`-th factorial ideal of modulus `M` of `nf`.

- `GEN zkfactmod_vec(GEN nf, GEN M, long n)`

  Return in a vector the `n` first factorial ideals of modulus `M` of `nf`. Faster the building the vector by incremental calls to `zkfactmod`.

- `GEN zkfactmodnorm(GEN nf, GEN M, long k)`

  Return the norm of the `k`-th factorial ideal of modulus `M` of `nf`.

- `GEN zkfactmodnorm_vec(GEN nf, GEN M, long n)`

  Return in a vector the norms of the first `n` factorial ideals of modulus `M` of `nf`. Faster than building the vector by incremental calls to `zkfactmodnorm`.

3 Regular basis

3.1 Regular basis for integer-valued polynomials

- `GEN zkfactpol(GEN nf, long k, const char *s, long cmode = 1)`

  Return a polynomial *pol* of degree `k` in $zk[X]$ such that $pol(zk)$ generates the `k`-th factorial ideal of `nf`. The variable name is set to `s`. The flag `cmode` tunes the returned polynomial coefficients : 0 for `t_POLMOD`, 1 for `t_POL`, 2 for `t_COL`.

- `GEN zkfactpol_vec(GEN nf, long n, const char *s, long cmode = 1)`

  Return a vector v of length `n` + 1 such that $v[i]$ = `zkfactpol(nf,i-1,s,cmode)`.

- `GEN zkregbasis(GEN nf, long n, const char *s, long cmode = 1)`

  Return in a vector v of length `n`+1 a regular basis for the zk-module Int(n,X). For such a basis to exist, it is **mandatory** that all factorial ideals up to `n` are principal and this can be checked with the function `ispolyaupto`.

3.2 Regular basis for integer-valued polynomials with integer-valued r-divided differences

- `GEN zkremfactpol(GEN nf, long r, long k, const char *s, long cmode = 1)`

  Return a polynomial *pol* of degree `k` in $zk[X]$ such that $pol(zk)$ generates the `k`-th `r`-removed factorial ideal of `nf`. The variable name is set to `s`. The flag `cmode` tunes the returned polynomial coefficients : 0 for `t_POLMOD`, 1 for `t_POL`, 2 for `t_COL`.

- `GEN zkremfactpol_vec(GEN nf, long r, long n, const char *s, long cmode = 1)`

  Return a vector v of length `n+1` such that $v[i]$ = `zkremfactpol(nf,r,i-1,s,cmode)`

- `GEN zkremregbasis(GEN bnf, long r, long n, const char *s, long cmode = 1)`

  Return in a vector v of length `n+1` a regular basis for the zk-module Int(n, r, X).

  It is the module of all integer-valued polynomials of `bnf`[X] of degree at most `n` such that their `r` first divided differences are also integer-valued.

  Being a regular basis means that $\deg(v[i]) = i - 1$ for $1 \leq i \leq n + 1$.

  For such a basis to exist, it is **mandatory** that the `n` first `r`-removed factorial ideals of `bnf` are principal and this can be checked with the function `ispolyaupto_rem`.

  If the later condition is not met, the behavior is undefined. The flag `cmode` tunes the returned polynomial coefficients : 0 for `t_POLMOD`, 1 for `t_POL`, 2 for `t_COL`.

3.3 Regular basis for integer-valued polynomials of modulus M

- `GEN zkfactmodpol(GEN nf, GEN M, long k, const char *s, long cmode = 1)`

  Return a polynomial *pol* of degree `k` in $zk[X]$ such that $pol(zk)$ generates the `k`-th factorial ideal of modulus `M` of `nf`. The variable name is set to `s`. The flag `cmode` tunes the returned polynomial coefficients : 0 for `t_POLMOD`, 1 for `t_POL`, 2 for `t_COL`.

- `GEN zkfactmodpol_vec(GEN nf, GEN M, long n, const char *s, long cmode = 1)`

  Return a vector v of length `n+1` such that $v[i]$ = `zkfactmodpol(nf,M,i-1,s,cmode)`.

- `GEN zkmodregbasis(GEN nf, long h, long n, const char *s, long cmode = 1)`

  Return in a vector v of length `n+1` a regular basis for the zk-module Int(n, M, X).

  It is the module of all integer-valued polynomials *pol* of `bnf`[X] of degree at most `n` such that if I_M is the ideal represented by the modulus `M` and $m \in I_M$, then $pol(mX + s) \in zk[X]$ for all $s \in zk$.

Being a regular basis means that $\deg(v[i]) = i - 1$ for $1 \leq i \leq n + 1$.

For such a basis to exist, it is **mandatory** that the `n` first factorial ideals of modulus `M` of `bnf` are principal and this can be checked with the function `ispolyaupto_mod`.

If the later condition is not met, the behavior is undefined. The flag `cmode` tunes the returned polynomial coefficients : 0 for `t_POLMOD`, 1 for `t_POL`, 2 for `t_COL`.