### PARI-GP Reference Card

#### Starting & Stopping GP

to enter GP, just type its name:
```
  gp
```
to exit GP, type
```
  q or quit
```

#### Help

describe function
```
  ?function
```
described extended description
```
  ??function
```
list of relevant help topics
```
  ??pattern
```

#### Input/Output & Defaults

output previous line, the lines before
```
  output previous line, the lines before
```
list of relevant help topics
```
  list of relevant help topics
```
describe function
```
  function
```
to exit GP, type
```
  gp
```

### Starting & Stopping GP

#### Starting & Stopping GP

```
  gp
```
```
  q or quit
```

### Metacommands

toggle timer on/off
```
  #
```
toggle timer on/off
```
  #
```
print time for last result
```
  \n
print time for last result
```
  \n```
print %n in raw format
```
  \n
print %n in raw format
```
  \n```
print %n in pretty format
```
  \n
print %n in pretty format
```
  \n```
print defaults
```
  \n
print defaults
```
  \n```
print %n in raw format
```
  \n
print %n in raw format
```
  \n```
print defaults
```
  \n
print defaults
```
  \n```
for forvec
```
  \n
for forvec
```
  \n```
for forvec
```
  \n
for forvec
```
  \n```
create empty list of maximal length
```
  \n
create empty list of maximal length
```
  \n```
delete all components of list
```
  \n
delete all components of list
```
  \n```
start new iteration of innermost enclosing loops
```
  \n
start new iteration of innermost enclosing loops
```
  \n```
new name of function
```
  \n
new name of function
```
  \n```
new name of function
```
  \n
new name of function
```
  \n```
break
```
  \n
break
```
  \n```
break
```
  \n
break
```
  \n```
next
```
  \n
next
```
  \n```
next
```
  \n
next
```
  \n```
return from current subroutine
```
  \n
return from current subroutine
```
  \n```
return from current subroutine
```
  \n
return from current subroutine
```
  \n```
evaluation
```
  \n
evaluation
```
  \n```
evaluation
```
  \n
evaluation
```
  \n```
evaluation
```
  \n
evaluation
```
  \n```
evaluation
```
  \n
evaluation
```
  \n```
evaluation
```
  \n
evaluation
```
  \n```
evaluation
```
  \n
evaluation
```
  \n```
evaluation
```
  \n
evaluation
```
  \n```
evaluation
```
  \n
evaluation
```
  \n```
evaluation
```
  \n
evaluation
```
  \n```
evaluation
```
  \n
evaluation
```
  \n```
evaluation
```
  \n
evaluation
```
  \n```
evaluation
```
  \n
evaluation
```
  \n```
evaluation
```
  \n
evaluation
```
  \n```
evaluation
```
  \n
evaluation
```
  \n```
evaluation
```
  \n
evaluation
```
  \n```
evaluation
```
  \n
evaluation
```
  \n```
evaluation
```
  \n
evaluation
```
  \n```
evaluation
```
  \n
evaluation
```
  \n```
evaluation
```
  \n
evaluation
```
  \n```
evaluation
```
  \n
evaluation
```
  \n```
evaluation
```
  \n
evaluation
```
  \n```
evaluation
```
  \n
evaluation
```
  \n```
evaluation
```
  \n
evaluation
```
  \n```
evaluation
```
  \n
evaluation
```
  \n```
evaluation
```
  \n
evaluation
```
  \n```
evaluation
```
  \n
evaluation
```
  \n```
evaluation
```
  \n
evaluation
```
  \n```
evaluation
```
  \n
evaluation
```
  \n```
evaluation
```
  \n
evaluation
```
  \n```
evaluation
```
  \n
evaluation
```
  \n```
evaluation
```
  \n
evaluation
```
  \n```
evaluation
```
  \n
evaluation
```
  \n```
evaluation
```
  \n
evaluation
```
  \n```
evaluation
```
  \n
evaluation
```
  \n```
evaluation
```
  \n
evaluation
```
  \n```
evaluation
```
  \n
evaluation
```
  \n```
evaluation
```
  \n
evaluation
```
  \n```
evaluation
```
  \n
evaluation
```
  \n```
evaluation
```
  \n
evaluation
```
  \n```
evaluation
```
  \n
evaluation
```
  \n```
evaluation
```
  \n
evaluation
```
  \n```
evaluation
```
  \n
evaluation
```
  \n```
evaluation
```
  \n
evaluation
```
  \n```
evaluation
```
  \n
evaluation
```
  \n```
evaluation
```
  \n
evaluation
```
  \n```
evaluation
```
  \n
evaluation
```
  \n```
evaluation
```
  \n
evaluation
```
  \n```
evaluation
```
  \n
evaluation
```
  \n```
evaluation
```
  \n
evaluation
```
  \n```
evaluation
```
  \n
evaluation
```
  \n```
evaluation
```
  \n
evaluation
```
  \n```
evaluation
```
  \n
evaluation
```
  \n```
evaluation
```
  \n
evaluation
```
  \n```
evaluation
```
  \n
evaluation
```
  \n```
evaluation
```
  \n
evaluation
```
  \n```
evaluation
```
  \n
evaluation
```
  \n```
evaluation
```
  \n
evaluation
```
  \n```
evaluation
```
  \n
evaluation
```
  \n```
evaluation
```
  \n
evaluation
```
  \n```
evaluation
```
  \n
evaluation
```
  \n```
evaluation
```
  \n
evaluation
```
  \n```
evaluation
```
  \n
evaluation
```
  \n```
evaluation
```
  \n
evaluation
```
  \n```
evaluation
```
  \n
evaluation
```
  \n```
evaluation
```
  \n
evaluation
```
  \n```
evaluation
```
  \n
evaluation
```
  \n```
evaluation
```
  \n
evaluation
```
  \n```
evaluation
```
  \n
evaluation
```
  \n```
evaluation
```
  \n
evaluation
```
  \n```
evaluation
```
  \n
evaluation
```
  \n```
evaluation
```
  \n
evaluation
```
  \n```
evaluation
```
  \n
evaluation
```
  \n```
evaluation
```
  \n
evaluation
```
  \n```
evaluation
```
  \n
evaluation
```
  \n```
evaluation
```
  \n
evaluation
```
  \n```
evaluation
```
  \n
evaluation
```
  \n```
evaluation
```
  \n
evaluation
```
  \n```
evaluation
```
  \n
evaluation
```
  \n```
evaluation
```
  \n
evaluation
```
  \n```
evaluation
```
  \n
evaluation
```
  \n```
evaluation
```
  \n
evaluation
```
  \n```
evaluation
```
  \n
evaluation
```
  \n```
evaluation
```
  \n
evaluation
```
  \n```
evaluation
```
  \n
evaluation
```
  \n```
evaluation
```
  \n
evaluation
```
  \n```
evaluation
```
  \n
evaluation
```
  \n```
evaluation
```
  \n
evaluation
```
  \n```
evaluation
```
  \n
evaluation
```
  \n```
evaluation
```
  \n
evaluation
```
  \n```
evaluation
```
  \n
evaluation
```
  \n```
Elliptic Curves

Elliptic curve initially given by 5-tuple \( E = \{a_1, a_2, a_3, a_4, a_6\} \).

Points are \([x, y]\), the origin is \([0, 0]\).

Initialize elliptic struct. \( ell \) create \( \text{ellinit}(E, \{f\}) \) \( a_1, a_2, a_3, a_4, a_6, b_2, b_4, b_6, c_4, c_6, \text{disc.} \) This data can be recovered by typing \( \text{elltors} \).

- \( E \) defined over \( \mathbb{R} \)
  - \( x \)-coordinates of points of order 2
  - real and complex periods
  - associated quasi-periods
  - volume of complex lattice
- \( E \) defined over \( \mathbb{Q_p} \)
- \( \text{elltors} \)
- \( \text{Tate}^{'s} \) \([u^2, u, q]\)
- \( \text{Mestre}^{'s} \) \( w \)
- change curve \( E \) using \( v = \{u, r, s, t\} \)
- change point \( z \) using \( v = \{u, r, s, t\} \)
- cond, min mod, Tamagawa num \( \{N, v, c\} \)
- Kodaira type of \( p \) fiber of \( E \)
- add points \( z_1 + z_2 \)
- subtract points \( z_1 - z_2 \)
- compute \( n \)
- check if \( z \) on \( E \)
- order of torsion point \( z \)
- torsion subgroup with generators
- \( y \)-coordinates of point \( (s) \) for \( x \)
- canonical bilinear form taken at \( z_1, z_2 \)
- canonical height of \( z \)
- height regulator matrix for pts in \( p \)
- \( p \)th coeff \( a_p \) of \( L \)-function, \( p \) prime
- \( k \)th coeff \( a_k \) of \( L \)-function
- vector of first \( n \) \( a_q \)’s in \( L \)-function

Elliptic & Modular Functions

- arithmetic-geometric mean \( \text{agm}(x, y) \)
- \( \text{ellj}(x) \)
- \( \text{weierstrass}^\sigma \) function
- \( \text{weierstrass}^v \) function
- \( \text{weierstrass}^\zeta \) function
- \( \text{modified Dedekind} \eta \) func. \( \Pi(1 - q^n) \)
- \( \text{Jacobi sine theta function} \)
- \( k \)-th derivative at \( x = 0 \) of \( \text{theta}(q, z) \)
- \( \text{weber}^\zeta \) function
- \( \zeta(n) \)

General Quadratic Forms

- \( \text{create} \ ax^2 + by^2 + cg^2 \) (distance \( d \))
- \( \text{reduce} \ x(s = \sqrt{d}, l = x) \)
- \( \text{polred} \) (primitive forms of \( \omega \))
- \( \text{polredabs} \) (primitive forms of \( \omega \))
- \( \text{polredabs} \) (prime forms of \( \omega \))
- \( \text{class} \) for \( \text{class} \) of \( \omega \)

Quadratic Fields

- \( \text{polred} \) (quadratic number \( \sqrt{-1 + \sqrt{d}} \))
- \( \text{polredabs} \) (quadratic number \( \sqrt{-1 + \sqrt{d}} \))
- \( \text{regulator} \) of \( \text{real} \) \( \text{fundamental} \) unit in \( \text{real} \) \( \text{quadratic} \) field
- \( \text{class} \) group of \( \text{class} \) for \( \text{quadratic} \) field
- \( \text{class} \) for \( \text{field} mod \) of \( \text{quadratic} \) field

General Number Fields: Initializations

- \( \text{nf} \) members: polynomial defining \( \sqrt[4]{f(\theta)} = 0 \)
- \( \text{nf} \) number of \( \text{real/complex} \) places
- \( \text{nf} \) discriminant of \( \text{nf} \)
- \( \text{t2} \) matrix
- \( \text{vector} \) of roots of \( f \)
- \( \text{integral} \) basis of \( \mathbb{Z}_K \) as powers of \( \theta \)
- \( \text{setup} \) different
- \( \text{recompute} \) using current precision
- \( \text{nf} \) relative \( \text{nf} \) given by \( g = 0 \) over \( K \)
- \( \text{nf} \) init big number field structure

Binary Quadratic Forms

- \( \text{create} \ ax^2 + by^2 + cg^2 \) (distance \( d \))
- \( \text{reduce} \ x(s = \sqrt{d}, l = x) \)
- \( \text{polred} \) (primitive forms of \( \omega \))
- \( \text{polredabs} \) (primitive forms of \( \omega \))
- \( \text{reduce} \ x(s = \sqrt{d}, l = x) \)
- \( \text{polred} \) (prime forms of \( \omega \))
- \( \text{class} \) for \( \text{field} mod \) of \( \text{quadratic} \) field
Simple Arithmetic Invariants (nf)

Elements are rational numbers, polynomials, or col-