Fq functions	
Function signature	**Description**
Zn(n)	If n is an integer >1, this function constructs the ring Z/(n). If A=Zn(n), and k is any integer, the expression k>>A yields the value k mod n.
[B,b]= extension (A, f(T), 'a') [B,b]=extension (A, [1,c1,...,cr], 'a') base(B) prime_field(F)	If A is a ring, f(T) a monic polynomial with coefficients in A, and a is an identifier, then the first statement binds B and b to the quotient A[T] / (f(T)) and to the class of T modulo f(T), which is named a (or x if 'a' is not supplied). The second signature is equivalent to the first when f(T) = Tr + c1Tr-1 +⋯+ cr. If A is a field and f(T) is irreducible over A, then B is a field of degree r over A. The ring A can be recovered from B with the command base(B). The elements 1, a, a^2, ..., a^{r-1} form a linear free basis of B over A which is called the *natural* or *standard basis*. The minimum subfield of a finite field F can be obtained with prime_field(F) (it is Zn(p), where p is the characteristic of F).
K_(a)	If the domain of an object a is a ring (which may be a field), this function returns A :: Ring (or A :: Field). Examples: K_(5) returns Z_ :: Ring, K_(1/5) returns Q_ :: Field.
md_mult(f,g,h) md_mul = md_mult	Delivers the remainder of the euclidean division of f·g by h.
md_power(f,n,h)	Delivers the remainder of the euclidean division of f^n by h.
md_double_power(f,m,n,h)	Delivers the remainder of the euclidean division of f^{m^n} by h.
period(a) order(a) period(f) exponent = period	For a non-zero element a of a finite field, the call period(a) delivers the minimum positive integer r such that a^r = 1, which coincides with order(a). If f is a univariate polynomial with variable x with coefficients in a finite field, period(f) yields the minimum positive integer r such that x^r = 1 mod f.
legendre (x, F) QR(F) QNR(F)	If x is a non-zero element of the finite field F of characteristic ≠ 2, legendre(x,F) retuns 1 if x is quadratic residue in F and −1 otherwise. By convention, legendre(0) is 0. For x ≠ 0, legendre(x,F) coincides with (−1)^{(q−1)/2}, where q is the cardinal of F. QR(F) yields the list of the non-zero quadratic residues of F. Similary, QNR(F) yields the list of the quadratic non-residues of F.
is_irreducible(f,K) is_irreducible(f)	The first call tells us whether the polynomial f is irreducible over K. The second does the same, but taking K=K_(f).
get_irreducible_polynomial(K,r,symbol='X')	Produces a monic irreducible polynomial of degree r over K in the varible supplied by 'symbol', which is 'X' by default.
irr(q,r)	Yields the number of monic irreducible polynomials over Fq of degree t.
Tr(x,K,L) Tr(x,K) Tr(x)	Given an element x defined in an extension of K, it computes the trace of the conjugates of x with respect to a finite field extension L/K. if L is omited, it takes L as the minimum extension where x is defined. If K is not defined, it takes K as the minimum field in the chain domain of x.
prime_field(F)	Given a domain F with a chain of fields K0⊆K1⊆...⊆Kn=F, it returns the base field K0.
index_table(x)	Given an element x of order k, it return a table given by the pairs [x**j,j]

Fq functions	
Function signature	**Description**
GF(q,n,alpha) GF(q,n) GF(q,alpha) GF(q)	This function creates the finite field of q^n elements with generative element represented by alpha. By defalut alpha = 'x'. If n is not given, the function creates the finite field of q elements. n can also be a polynomial or a list of coefficients and then the functions creates the extension using that polynomial. q can also be a domain and then, the function creates an extension using a given polynomial.
order(k, n)	If gcd(k,n) = 1, the order of k in Zn*. Otherwise 'Error'
inverse(k,n)	If gcd(k,n) = 1, inverse(k,n) computes a positive integer k' such that k' < n and k'k ≡ 1 mod n. Otherwise, "Error"
mult(n,k,b) bpow(n,k,b) quot(n,k,b) power(n,k,m)	The value of these expressions is n · k mod 2^b, n^k mod 2^b and (m / k) mod 2^b, respectively. In the latter case, k has to be odd. The function bpow(n,k,b) coincides with power(n,k,2^b), as power(n,k,m) computes n^k mod m. These functions are used, for example, in the definition of the next two.
jacobi(a,n) legendre(a,n)	Computes the Jacobi symbol (a/n) of two integers a and n, which must be odd and positive. The PyM implementation is based on Algorithm 2.3.5 of Crandall-Pomerance-2005. If n is prime, it coincides with the Legendre symbol (a/n), which is 0 if a is divisible by n and otherwise it is +1 or -1 according to whether a is or is not a quadratic residue mod n. It coincides with legendre(a,Z_n)
nroot(n,k,b) nsqroot(n,b) sqroot(a,F)	If n is an odd integer and k is either odd or 2, nroot(n,k,b) computes an integer r < 2^b such that r^kn ≡ 1 mod 2^b, which is a kth root of n^´{-1} mod 2^b. The function nsqroot(n,b) is equivalent to nroot(n,2,b). These funtions are used in the definition of is_perfect_power(n). sqroot(a,F) returns an element b in the domain F, such that b*b = a in the domain F.
cyclotomic_class (k, n, q) cyclotomic_class (k, n)	Assuming that q and n are positive integers and that gcd(q,n)=1, the call cyclotomic_class(k,n,q) supplies the q-cyclotomic class of k mod n, which by definition is the list [k, q·k,q^2·k,...,q^{r−1·k}], where the operations are done mod n and r is the least positive integer such that q^r·k=1.
cycloctomic_classes (n, q) cycloctomic_classes (n)	Assuming that q and n are positive integers and that gcd(q,n)=1, the function cycloctomic_classes(n,q) furnishes the list of all the q-cyclotomic classes mod n. Finally, cycloctomic_classes(n) is defined as cycloctomic_classes(n,2).
frobenius(K)	
conjugates(x,K) conjugates(x)	List of conjugates of an element x in L over subfield K. By default K is the minimum subfield of L.
nm(x,K) nm(x)	It returns the trace of the conjugates of x over K. By default K is Zn(2)
is_primitive(f)	It returns if f is a primitive element in its domain.
primitive_root(K)	It returns a primiitve element of the finite field F.
md_trace(h,m,f)	It is a auxiliar function of equal degree splitting. It computes the trace of an element in F2: h+h**2+h**4 + h**8+ ... + h**(2**(m-1)) mod f