FUNCTION : misc[briefmischelp] - brief help for a misc function

CALLING SEQUENCE : briefmischelp(funcname)

PARAMETERS : funcname - name of misc function

SYNOPSIS :
Help for a function in the misc package.
It only includes FUNCTION, CALLING SEQUENCE and PARAMETERS

EXAMPLES :

```
> with(qseries):
> briefmischelp(sieveqcheck);
```

FUNCTION : misc[dilly] - dilate q-series

CALLING SEQUENCE : dilly(func,d)

PARAMETERS : func - series or expression in q
d - positive integer

GLOBAL VARIABLES : none

SYNOPSIS : Performs substitution q -> q^d

EXAMPLES :

```
> with(misc):
> with(qseries):
> E1:=etaq(q,1,100):
> E2:=dilly(E1,2):
> etamake(E1,q,100);
```

$$\frac{\text{eta}(\tau)}{q^{1/24}}$$

```
> etamake(E2,q,100);
```

$$\frac{\text{eta}(2\,\tau)}{q^{1/12}}$$

DISCUSSION :

SEE ALSO :

```
FUNCTION :  misc[EISENq] - Eisenstein series


CALLING SEQUENCE :  EISENq(d,q,T)


PARAMETERS :   d  -  positive even integer
               T  -  positive integer

GLOBAL VARIABLES : none

SYNOPSIS :
      Eisenstein series
      1 - 2*d/bernoulli(d)*add( sigma[d-1](n)*q^n, n=1 .. T)


EXAMPLES :
> with(misc):
> with(qseries):
> series(EISENq(2,q,10),q,10);
memory used=5.6MB, alloc=41.3MB, time=0.14
                2       3        4        5        6        7        8        9
1 - 24 q - 72 q  - 96 q  - 168 q  - 144 q  - 288 q  - 192 q  - 360 q  - 312 q

          10
     + O(q  )

> D12:=series(EISENq(4,q,100)^3-EISENq(6,q,100)^2,q,100):
> etamake(D12,q,20);
                                              24
                                1728 eta(tau)

DISCUSSION :

SEE ALSO :  Phiq
```

```
FUNCTION :  misc[findleg] - find a pattern in a list primes related to
                            quadratic residues or nonresidues


CALLING SEQUENCE :  findleg()
                    findleg(L,plist,a1,a2,sg)


PARAMETERS :  L     -  list [[p1,r1],[p2,r2],...]
              plist - list of primes
              a1,a2 - positive integers a1 < a2
              sg    - +1 or -1

GLOBAL VARIABLES : none

SYNOPSIS :
    Searches from a = a1 to a2  and returns the pair [L1,R]
    where L1 =[p1,p2, ...] and R is a list of values of a such
    L1 coincides with primes p in plist that satisfy LegendreSymbol(a,p)=sg


EXAMPLES :
> with(misc):
> with(NumberTheory):
> L:=[[11, 4], [23, 11], [31, 17], [41, 26], [43, 28], [47, 4]];
        L := [[11, 4], [23, 11], [31, 17], [41, 26], [43, 28], [47, 4]]

> plist2:=[seq(ithprime(j),j=2..15)];
        plist2 := [3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47]

> findleg(L,plist2,2,100,-1);
                          [11, 23, 31, 41, 43, 47], [30]

> seq([p,LegendreSymbol(30,p)],p in plist2);
[3, 0], [5, 0], [7, 1], [11, -1], [13, 1], [17, 1], [19, 1], [23, -1], [29, 1],
    [31, -1], [37, 1], [41, -1], [43, -1], [47, -1]

DISCUSSION : A pattern was discovered for the primes in the list
             [11, 23, 31, 41, 43, 47]
             These are the odd primes p less than or equal to 47 that satisfy
             LegendreSymbol(30,p) = -1.

SEE ALSO :
```

```
FUNCTION :  misc[findlegV2] - find a pattern in a list primes related to
                              quadratic residues or nonresidues


CALLING SEQUENCE :  findlegV2()
                    findleg(L,r1,r2,sg)

 findlegV2(L,r1,r2,sg)
    L = list of primes
    Identifies L as list of primes satisfying
    Legendre symbol condition
    (where m<= m2) if possible.

PARAMETERS :  L     -  list if primes
              r1,r2 - positive integers a1 < a2
              sg    - +1 or -1

GLOBAL VARIABLES : none

SYNOPSIS :
    Searches from r = r1 to r2  and returns the pair [r,sg]
    where L coincides with the odd primes (up to largest prime in L )
    satisfy LegendreSymbol(r,p)=sg
    It returns the smallest such r in the range that works if it exists.
    This is an IMPROVED version of findleg

EXAMPLES :
> with(misc):
> with(NumberTheory):
> LPL:=[7,11,13,17,31,37,41,59,61,79];
                 LPL := [7, 11, 13, 17, 31, 37, 41, 59, 61, 79]

> findlegV2(LPL,2,12,-1);
                                    [6, -1]

> plist:=[seq(ithprime(j),j=2..22)];
plist := [3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67,
    71, 73, 79]

> seq([p,LegendreSymbol(6,p)],p in plist);
[3, 0], [5, 1], [7, -1], [11, -1], [13, -1], [17, -1], [19, 1], [23, 1],
    [29, 1], [31, -1], [37, -1], [41, -1], [43, 1], [47, 1], [53, 1], [59, -1],
    [61, -1], [67, 1], [71, 1], [73, 1], [79, -1]

DISCUSSION : A pattern was discovered for the primes in the list
             [7, 11, 13, 17, 31, 37, 41, 59, 61, 79]
             These are the odd primes p less than or equal to 79 that satisfy
             LegendreSymbol(6,p) = -1.

SEE ALSO :  findleg
```

FUNCTION : misc[findpcoffs] - find prime coefficients in q-series

CALLING SEQUENCE : findpcoffs(P)

PARAMETERS : P - q-series

GLOBAL VARIABLES : none

SYNOPSIS :
For a q-series F = sum f(n)*q^n returns a list
[[p1,[n11,n12,....], [p2,[n21,n22,....] .....]
where p1, p2, .. are the prime coefficients in the q-series.
Here f(n) = p[j], for n=n[j,1], n[j,2], ...

EXAMPLES :

```
> with(misc):
> with(qseries):
> P:=series(1/etaq(q,1,100),q,101):
> LP:=findpcoffs(P);
LP := [[2, [2]], [3, [3]], [5, [4]], [7, [5]], [11, [6]], [101, [13]],
    [17977, [36]], [10619863, [77]]]

> seq(k[2], k in LP);
                   [2], [3], [4], [5], [6], [13], [36], [77]

> RR:=add(q^(n*(n+1)/2)/aqprod(-q,q,n),n=0..100):
> RRS:=series(RR,q,100):
> findpcoffs(%);
[[2, [3, 4, 8, 10, 13, 14, 17, 18, 19, 24, 25, 28, 32, 39, 42, 43, 47, 48, 50,
    52, 54, 55, 62, 67, 69, 73, 74, 75, 76, 78, 83, 84, 87, 88, 89, 90, 95, 99]
    ], [3, [22, 92]]]
```

DISCUSSION : We see p(n) is prime for n=2, 3, 4, 5, 6, 13, 36, ...
We see r(n)=2 for n=3,4,8,10,13,... where r(n) is coeff of q^n
in R = sum(q^(n*(n+1)/2)/aqprod(-q,q,n), n)

SEE ALSO :

```
FUNCTION :  misc[findres] - find a residue pattern


CALLING SEQUENCE :  findres()
                    findres(L,r1,r2)


PARAMETERS :   L  -  list of pairs [p,R] (p prime, R residue mod p)
               r1,r2 - positive integers r1 < r2

GLOBAL VARIABLES : none

SYNOPSIS :
    Find a rational number s/r (where r1 <= r <= r2) that satisifies
    R == s/r mod p  for each [p,R] in the list


EXAMPLES :
> with(misc):
> RLPL := [[7, 4], [11, 3], [17, 14], [19, 9]];
                   RLPL := [[7, 4], [11, 3], [17, 14], [19, 9]]

> findres(RLPL,1,40);
                                        -1
                                        --
                                        40
> seq( modp(40*K[2]+1,K[1]),K in RLPL);
                                  0, 0, 0, 0

DISCUSSION :  40*r == -1 mod p for
              (p,r) = (7,4), (11,3), (17,14), (19,9)
SEE ALSO :
```

```
FUNCTION :  misc[heckerodgerssearch] - search for Hecke-Rogers identities
                                       among mock-theta functions


CALLING SEQUENCE :  heckerodgerssearch(symfunc,func,flist,plist,d,T)

REQUIRES PACKAGES: qseries, etatheta


PARAMETERS :   symfunc - symbolic name for mock theta function
                  func - q-series of mock theta function
                 flist - q-series list of mock theta functions
                 plist - list of primes
                     d - positive integer
                     T - positive integer

GLOBAL VARIABLES : none

SYNOPSIS :
    This proc returns the list [symfunc,RESULTS1,"*",RESULTS3]):
    RESULTS1 is a list of terms [k, V1] where  k in [1,2,3,4,5,6]
    and V1 is a list of terms [p,r].
    Each such k,p,r corresponds to an alleged identity
       a(pn + r) = c1*b( (n-r2)/p)
   where  sum a(n)*q^n = (func(q) * etalistw1[k](q^d))
   and    sum b(n)*q^n = (other.func.in.flist(q) * etalistw1[k](q^d))
    RESULTS3 same as RESULTS1 except etalistw1 is replaced by etalistw3.
   NOTE: A lot of other stuff gets printed out in the process.

EXAMPLES :
> with(qseries):
> with(misc):
> with(etatheta):
> heckerodgerssearch();
> read "FF05func.m":
> read "FF15func.m":
> plist:=[seq(ithprime(j),j=2..10)];
                   plist := [3, 5, 7, 11, 13, 17, 19, 23, 29]

> heckerodgerssearch(_F05,F05Aq,[F05Aq,F15Aq],plist,1,10000);
[_F05, [[1, [[7, 3], [13, 3], [17, 13], [23, 13]]], [2, [[17, 13], [23, 13]]],
    [4, [[13, 4], [23, 11]]], [5, [[23, 14]]]], "*",
    [[1, [[23, 11]]], [3, [[17, 12]]], [5, [[17, 13], [23, 13]]]]]

> V0:=series(etalistw1[1]*F05Aq,q,5000):
> V1:=convert(series(etalistw1[1]*F15Aq,q,5000),polynom):
> series(sift(V0,q,7,3,5000)-subs(q=q^7,V1)*q^4,q,700);
                                     704
                                  O(q   )

> a:=n->coeff(V0,q,n):
> b:=n->if n>=0 and type(n,integer) then coeff(V1,q,n) else 0 fi:
> {seq(a(7*n+3)-b((n-4)/7),n=0..700)};
                                     {0}

DISCUSSION : _F05 and _F15 are the fifth order mock theta functions
     _F05 = sum q^(2*n^2)/(q;q^2)[n], n>=0),
     _F15 = sum q^(2*n^2+2*n)/(q;q^2)[n+1], n>=0),
     We discuss the item:
     [[1, [[7, 3], [13, 3], [17, 13], [23, 13]]]
     Let sum a(n)*q^n =  _F05*(q;q)inf
     and sum b(n)*q^n =  _F15*(q;q)inf
     Then it seems that
           a(7n + 3) = b( (n-4)/7 ).

SEE ALSO :
```

```
FUNCTION :  misc[hrmatch] - determine whether qseries  has a certain
                            Hecke type identity

CALLING SEQUENCE :  hrmatch()
                    hrmatch(func,p,r,T,funclist)

REQUIRES PACKAGE: qseries

PARAMETERS :    func  -  q-series
                   p  -  prime
                   r  - residue mod p
                   T  - positive integer
             funclist  - list of q-series

GLOBAL VARIABLES : none

SYNOPSIS :
   Returns [k,c,s] if
   sift(func,q,p,r) = c*q^s*funclist[k](q^p)


EXAMPLES :
> with(qseries):
> with(misc):
> with(etatheta):
> read "FF05func.m":
> read "FF15func.m":
> V0:=series(etalistw1[1]*F05Aq,q,5000):
> V1:=convert(series(etalistw1[1]*F15Aq,q,5000),polynom):
> hrmatch(V0,7,3,1000,[V0,V1]);
                                    [2, 1, 4]

DISCUSSION :
      This means (up to q^5000) that
      sift(V0,q,7,3) = 1*q^4*V1(q^7)

SEE ALSO :
```

```
FUNCTION :  misc[hrs] - Search for sift identities related to possible
                        Hecke-Rogers identities


CALLING SEQUENCE :  hrs()
                    hrs(func,symfunc,flist,symflist,plist,T)

REQUIRES PACKAGE: qseries

PARAMETERS :   func - qseries
            symfunc - not used
              flist - list of qseries
           symflist - not used
              plist - list of primes
                  T - positive integer

GLOBAL VARIABLES : none

SYNOPSIS :
  Looks for identities of the form
  sift(func,q,p,r) = c*q^s*funclist[k](q^p)
  Returns a list of probable identities encoded as [p,r,k,c,s]

EXAMPLES :
> with(qseries):
> with(misc):
> with(etatheta):
> read "FF05func.m":
> read "FF15func.m":
> V0:=convert(series(subs(q=q,etalistw3[1])*(F05Aq-1),q,20001),polynom):
> V1:=convert(series(subs(q=q,etalistw3[1])*F15Aq,q,20001),polynom):
> plist:=[seq(ithprime(j),j=2..25)]:
> flist:=[V0,V1]:
> hrs(V0,symfunc,flist,symlist,plist,20000);
[[23, 11, 2, 23, 16], [29, 11, 1, 29, 3], [41, 32, 1, -41, 4],
    [47, 32, 2, -47, 33], [83, 40, 1, 83, -107], [89, 34, "ONLY.ONE.TERM"]]

> series(sift(V0,q,23,11,20000)-23*q^16*subs(q=q^23,V1),q,trunc(20000/23));
                                      913
                                   O(q   )

> series(sift(V0,q,83,40,20000)-83*subs(q=q^83,V0)/q^107,q,240+107);
                                      308
                                   O(q   )

DISCUSSION :
     [23, 11, 2, 23, 16] <-> sift(V0,q,23,11) = 23*q^16*V1(q^23)
     [83, 40, 1, 83, -107]  <-> sift(V0,q,23,11) = 23*q^16*V0(q^23)

SEE ALSO : hrmatch
```

```
FUNCTION :  misc[hrsanalyze] -


CALLING SEQUENCE :  hrsanalyze(HRSL,T,M)


PARAMETERS :   HRSL - list output of hrs function
                  T - positive integer

GLOBAL VARIABLES : none

SYNOPSIS :
   analyze output of hrs function using the functions
   idres, findlegV2 and findres

EXAMPLES :
> with(qseries):
> with(etatheta):
> with(misc):
> read "f05func.m":
> read "f15func.m":
> qdegree(f05Aq);
                                    100000

> qdegree(f15Aq);
                                    100000

> plist:=[seq(ithprime(j),j=2..25)];
plist := [3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67,
          71, 73, 79, 83, 89, 97]

> FL11:=Array(1..2):
> FL11[1]:=series(f05Aq*etalistw1[1],q,50001):
> FL11[2]:=series(f15Aq*etalistw1[1],q,50001):
> HRS11:=hrs(FL11[1],symfunc,FL11,symflist,plist,20000);
HRS11 := [[7, 4, 2, 1, 1], [11, 3, 1, 1, 0], [17, 14, 2, 1, 3],
    [19, 9, 1, -1, 0], [23, 4, 2, -1, 5], [29, 21, 1, -1, 0],
    [47, 27, 2, 1, 10], [59, 28, 1, -1, 1], [61, 32, 1, 1, 1],
    [73, 31, 2, -1, 16], [97, 80, 1, 1, 21]]

> hrsanalyze(HRS11,120,40);
[[[7, 4, 2, 1, 1], [11, 3, 1, 1, 0], [17, 14, 2, 1, 3], [19, 9, 1, -1, 0],
    [23, 4, 2, -1, 5], [29, 21, 1, -1, 0], [47, 27, 2, 1, 10],
    [59, 28, 1, -1, 1], [61, 32, 1, 1, 1], [73, 31, 2, -1, 16],
    [97, 80, 1, 1, 21]], " ",
    ["Residue of primes:", 40, {7, 11, 17, 19, 21, 23, 29, 33}],
                                                              -1
    [10, "is QNR for these primes"], ["Residue is probably", --, "mod p"]]
                                                              40
> squareit( n*(5*n+1)/2-j^2,n,j);
                                  2
                          5 (n + 1/10)     2
                          ------------ - j  - 1/40
                                2
DISCUSSION :
    This produces some output to give a clue for the shape of
    a Hecke-Rogers identity for  (q;q)inf * f0
    where f0 is the 5th order mock theta function
    f0 = sum q^(n^2)/(-q,q)[n]
    It turns out (by Andrews[1986]) that
    (q)inf*f0 = add(add( (-1)^j*q^(n*(5*n+1)/2-j^2)*(1-q^(4*n+2)), n>=|j|),j)
    which is the Hecke-Rodgers identity that is leading to this

SEE ALSO :
```

```
FUNCTION :  misc[hrsanalyzebat] - batch version of hrsanalyzebat


CALLING SEQUENCE :  hrsanalyzebat(HRSL,T,M)


PARAMETERS :   HRSL - list output of hrs function
                  T - positive integer

GLOBAL VARIABLES : none

SYNOPSIS :
   analyze output of hrs function using the functions
   idres, findlegV2 and findres

EXAMPLES :
> with(qseries):
> with(etatheta):
> with(misc):
> read "f05func.m":
> read "f15func.m":
> qdegree(f05Aq);
                                    100000

> qdegree(f15Aq);
                                    100000

> plist:=[seq(ithprime(j),j=2..25)];
plist := [3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67,
         71, 73, 79, 83, 89, 97]

> FL11:=Array(1..2):
> FL11[1]:=series(f05Aq*etalistw1[1],q,50001):
> FL11[2]:=series(f15Aq*etalistw1[1],q,50001):
> HRS11:=hrs(FL11[1],symfunc,FL11,symflist,plist,20000);
HRS11 := [[7, 4, 2, 1, 1], [11, 3, 1, 1, 0], [17, 14, 2, 1, 3],
    [19, 9, 1, -1, 0], [23, 4, 2, -1, 5], [29, 21, 1, -1, 0],
    [47, 27, 2, 1, 10], [59, 28, 1, -1, 1], [61, 32, 1, 1, 1],
    [73, 31, 2, -1, 16], [97, 80, 1, 1, 21]]

> hrsanalyzebat(HRS11,120,40);
[[[7, 4, 2, 1, 1], [11, 3, 1, 1, 0], [17, 14, 2, 1, 3], [19, 9, 1, -1, 0],
    [23, 4, 2, -1, 5], [29, 21, 1, -1, 0], [47, 27, 2, 1, 10],
    [59, 28, 1, -1, 1], [61, 32, 1, 1, 1], [73, 31, 2, -1, 16],
    [97, 80, 1, 1, 21]],
    ["Residue of primes:", 40, {7, 11, 17, 19, 21, 23, 29, 33}],
                                                          -1
    [10, "is QNR for these primes"], ["Residue is probably", --, "mod p"]]
                                                          40

> squareit( n*(5*n+1)/2-j^2,n,j);
                                  2
                          5 (n + 1/10)     2
                          ------------ - j  - 1/40
                                2

DISCUSSION :
    This produces some output to give a clue for the shape of
    a Hecke-Rogers identity for  (q;q)inf * f0
    where f0 is the 5th order mock theta function
    f0 = sum q^(n^2)/(-q,q)[n]
    It turns out (by Andrews[1986]) that
    (q)inf*f0 = add(add( (-1)^j*q^(n*(5*n+1)/2-j^2)*(1-q^(4*n+2)), n>=|j|),j)
    which is the Hecke-Rodgers identity that is leading to this

SEE ALSO :
```

FUNCTION : misc[idres] - identify residues of primes

CALLING SEQUENCE : idres(L,m2)

PARAMETERS : L - list of primes
m2 - positive integer

GLOBAL VARIABLES : none

SYNOPSIS :
Identifies L as list of primes satisfying a set of of congruences mod m
(where m<= m2) if possible.

EXAMPLES :

```
> with(misc):
> P := [7,11,17,19,23,29,47,59,61];
                    P := [7, 11, 17, 19, 23, 29, 47, 59, 61]

> idres(P,100);
                       [40, {7, 11, 17, 19, 21, 23, 29}]
```

DISCUSSION :

SEE ALSO :

FUNCTION : misceries[miscchanges] - print out list of recent changes to
misc package

CALLING SEQUENCE : miscchanges()

PARAMETERS : none

GLOBAL VARIABLES : none

SYNOPSIS :

miscchanges() prints out a list of changes in previous versions
of the misc package.

EXAMPLES :

```
> with(misc):
> miscchanges();
************************************************************
*
*
*  misc package version 0.1 -  Wed, Feb 13, 2019  8:51:39 PM
*  misc package version 0.2  - Fri, Mar  1, 2019  4:01:37 PM
*  misc package version 0.3  - Fri, Mar 29, 2019  6:50:44 PM
*  misc package version 0.4  - Tue Jul  9 10:08:30 EDT 2019
*  misc package version 0.5  - Wed Mar  4 10:34:43 EST 2020
*  This version tested on MAPLE 2018
*
*
*  Changes since version 0.4
*
*      *  New functions:
*         newprodmake, SaveAll
*
*  Changes since version 0.3
*
*      *  New functions:
*         EISENq, miscchanges, PHIq
*
*  Please report any problems to fgarvan@ufl.edu
*  NO Previous versions:

************************************************************
```

FUNCTION : misc[mischelp] - help for a misc function

CALLING SEQUENCE : mischelp(funcname)

PARAMETERS : funcname - name of misc function

SYNOPSIS :
Help for a function in the misc package

EXAMPLES :

```
> with(qseries):
> qshelp(sieveqcheck);
```

FUNCTION : qseries[miscpfuncs] - list functions in the misc package

CALLING SEQUENCE : miscpfuncs()

PARAMETERS : none

SYNOPSIS :
List functions in the misc package

EXAMPLES :

```
> with(misc):
> miscpfuncs();
------------------------------------------------------------
 miscpfuncs()
 FUNCTIONS in misc package:
dilly,EISENq,findleg, findlegV2, findpcoffs, findres, heckerodgerssearch,
hrmatch, hrs, hrsanalyze, idres, miscchanges, miscpversion, mocksearch,
newprodmake, Phiq, polyfind, SaveAll, sieveqcheck,
siftfind, siftfindrange, squareit
------------------------------------------------------------
```

DISCUSSION:

SEE ALSO :

FUNCTION : qseries[miscpversion] - package version

CALLING SEQUENCE : miscpversion()

PARAMETERS : none

SYNOPSIS :

Prints version and date of misc package

EXAMPLE :

```
> with(misc):
> miscpversion();
************************************************
*
*CURRENT VERSION:
*  misc  package version 0.5
*  Wed Mar  4 10:34:43 EST 2020
*  This version tested on MAPLE 2018
*
*PREVIOUS VERSIONS:
*  misc  package version 0.4
*  Tue Jul  9 10:08:30 EDT 2019
*  This version tested on MAPLE 2018
*
*  misc  package version 0.3
*  Fri, Mar 29, 2019  6:50:44 PM
*  This version tested on MAPLE 2018
*
*  misc  package version 0.2
*  Fri, Mar  1, 2019  4:01:37 PM
*  This version tested on MAPLE 2018
*
*  misc  package version 0.1
*  Wed, Feb 13, 2019  8:51:39 PM
*  This version tested on MAPLE 2018
*
*  Please report any problems to fgarvan@ufl.edu
*  NO Previous versions:

************************************************
```

```
FUNCTION :  misc[mocksearch] - search for Hecke-Rogers behaviour for a
                               list of mock (or similar) functions

CALLING SEQUENCE :  mocksearch()
                    mocksearch(j,mockfuncLIST,klist,plist,T,w,d)

REQUIRES PACKAGES: qseries, etatheta

PARAMETERS:
 j - integer
 mockfuncLIST - list of qseries (usually mock theta functions)
 klist - list of integers
 plist - list of primes
 T - large integer
 w - 1/2 or 3/2 (corresponds to weight)
 d - positive integer

GLOBAL VARIABLES : none

SYNOPSIS :
  Utilizes hrs and hrsanalyze to search for Hecke-Rogers behaviour
  for a list mock (or similar) functions.
  If w=1/2 it studies the function mockfuncLIST[j]*etalistw1[k](q^d)
  If w=3/2 it studies the function mockfuncLIST[j]*etalistw3[k](q^d)
  for k in klist.


EXAMPLES :
> with(NumberTheory):
> with(qseries):
> with(etatheta):
> with(misc):
> read "f05func.m":
> read "f15func.m":
> MFL5f:=[f05Aq,f15Aq]:
> plist:=[seq(ithprime(j),j=2..25)]:
> mocksearch(1,MFL5f,[1,2,3],plist,10000,1/2,1);
"k=", 1, " *** ", [[[7, 4, 2, 1, 1], [11, 3, 1, 1, 0], [17, 14, 2, 1, 3],
    [19, 9, 1, -1, 0], [23, 4, 2, -1, 5], [29, 21, 1, -1, 0],
    [47, 27, 2, 1, 10], [59, 28, 1, -1, 1], [61, 32, 1, 1, 1],
    [73, 31, 1, -1, 16]],
    ["Residue of primes:", 40, {7, 11, 17, 19, 21, 23, 29, 33}],
                                                          -1
    [10, "is QNR for these primes"], ["Residue is probably", --, "mod p"]]
                                                          40

"k=", 2, " *** ", [[[11, 3, 1, -1, 0], [17, 14, 2, 1, 3], [19, 9, 1, 1, 0],
    [29, 21, 1, 1, 0], [59, 28, 1, 1, 1], [61, 32, 1, -1, 1],
    [73, 31, 1, -1, 16]], ["Residue of primes:", 44, {11, 15, 17, 19, 29}],
                                                           -1
    ["QNR UNKNOWN for these primes"], ["Residue is probably", --, "mod p"]]
                                                           40

"k=", 3, " *** ", [[[11, 2, 1, 1, 1], [23, 2, 2, -1, 7], [31, 11, 1, -1, 3],
    [41, 18, 1, -1, 4], [43, 11, 2, 1, 13], [47, 23, 2, 1, 14],
    [53, 18, 2, 1, 16], [59, 23, 1, -1, 6], [61, 37, 1, 1, 6],
    [67, 44, 1, -1, 20], [73, 37, 1, -1, 22], [79, 44, 1, 1, 8],
    [89, 57, 1, 1, 9], [97, 88, "ONLY.ONE.TERM"]], ["Residue of primes:", 49,
    {4, 10, 11, 12, 18, 23, 24, 30, 31, 40, 41, 43, 47, 48}],
                                                          -13
    [30, "is QNR for these primes"], ["Residue is probably", ---, "mod p"]]

DISCUSSION :
    See the DISCUSSION for the hrsanalyze function

SEE ALSO :  hrs, hrsanalyze
```

FUNCTION : misc[newprodmake] - convert a q-series to an infinite product

CALLING SEQUENCE : newprodmake(f,q,T)
newprodmake(f,q,T,list)

PARAMETERS : f - qseries
T - positive integer

GLOBAL VARIABLES : none

SYNOPSIS :
Returns a formal product for the series f (up to q^T)
Returns the result as a list of exponents is list option used.
NOTE: This is a new version of prodmake that does not assume that
f = 1 + c1*q^1 + ...

EXAMPLES :

```
> with(misc):
> with(qseries):
> Mao1a:=add(normal((-1)^n*q^(n*(3*n+1)/2)/(1+q^(5*n))),n=-30..30):
> Mao1d:=etaq(q,1,1000)/etaq(q,25,1000)
  *add( (-1)^n*q^(75*n*(n+1)/2+5)/(1+q^(25*n+5)),n=-10..10):
> M1ad:=series(Mao1a-Mao1d,q,1000):
> newprodmake(sift(M1ad,q,5,2,900),q,50,list);
[-1, [3, 0, 2, 2, 2, 2, 2, 0, 3, 2, 3, 0, 2, 2, 2, 2, 2, 0, 3, 2, 3, 0, 2, 2, 2,
    2, 2, 0, 3, 2, 3, 0, 2, 2, 2, 2, 2, 0, 3, 2, 3, 0, 2, 2, 2, 2, 2, 0, 3]]

> jacprodmake(sift(M1ad,q,5,2,900),q,50);
                     3                     2                     2
- JAC(1, 10, infinity)  JAC(3, 10, infinity)  JAC(4, 10, infinity)

                           /                     6
    JAC(5, 10, infinity)  /  JAC(0, 10, infinity)
                         /
```

DISCUSSION :

SEE ALSO :

FUNCTION : misc[Phiq] -

CALLING SEQUENCE : Phiq(j,q,T)

PARAMETERS : j,T - positive integers

GLOBAL VARIABLES : none

SYNOPSIS :
Returns sum sigma[j](n)*q^n, n=1..T
sigma[j](n) = sum d^j, d divides n

EXAMPLES :

```
> with(misc):
> with(NumberTheory):
> add(sigma[3](n)*q^n,n=1..10);
      10        9        8        7        6        5       4       3      2
1134 q   + 757 q  + 585 q  + 344 q  + 252 q  + 126 q  + 73 q  + 28 q  + 9 q

     + q

> series(Phiq(3,q,10),q,11);
       2       3       4        5        6        7        8        9
q + 9 q  + 28 q  + 73 q  + 126 q  + 252 q  + 344 q  + 585 q  + 757 q  + 1134

     10      11
    q   + O(q  )

> series(Phiq(3,q,100)-add(sigma[3](n)*q^n,n=1..100),q,101);
                                        101
                                    O(q   )
```

DISCUSSION :

SEE ALSO : EISENq

```
FUNCTION :  misc[polyfind] - find quadratic polynomial to a sequence


CALLING SEQUENCE :  polyfind(n0,n1,n2,T)


PARAMETERS :    n0,n1,n2,T - integers

GLOBAL VARIABLES : none

SYNOPSIS :
  Find a polynomial p(x) = a*x^2+b*x+c such that
    p(T)=n0, p(T+1)=n1, p(T+2)=n2


EXAMPLES :

> with(qseries):
> with(misc):
> HR3:=convert(normal(series(tripleprod(z*q,q^3,200)*add(q^(n^2)/aqprod(q*z,q,n)/aqprod(q
/z,q,n),n=0..20),q,200)),polynom):
> HR30:=coeff(HR3,z,0);
         196    188    165    130    111    105    88    63    50    46    35
HR30 := q    - q    - q    + q    + q    - q    - q   + q   + q   - q   - q

        20    13    11    6
     + q   + q   - q   - q  + q + 1

> polyfind(0,13,50,0);
                              " a b c", [12, 1, 0]
              [0, 13, 50, 111, 196, 305, 438, 595, 776, 981, 1210]
                                      2
                                  12 x  + x

> polyfind(1,20,63,0);
                              " a b c", [12, 7, 1]
             [1, 20, 63, 130, 221, 336, 475, 638, 825, 1036, 1271]
                                2
                            12 x  + 7 x + 1

> polyfind(6,35,88,1);
                             " a b c", [12, -7, 1]
               [1, 6, 35, 88, 165, 266, 391, 540, 713, 910, 1131]
                                2
                            12 x  - 7 x + 1

> polyfind(11,46,105,1);
                             " a b c", [12, -1, 0]
              [0, 11, 46, 105, 188, 295, 426, 581, 760, 963, 1190]
                                      2
                                  12 x  - x
DISCUSSION :

SEE ALSO :
```

```
FUNCTION :  misc[quadresL] - quadratic residues or non-residues mod p

CALLING SEQUENCE :  quadresL(p, epv)

PARAMETERS :   p - odd prime
             epv - 1 or -1

SYNOPSIS :
   Returns residues r such that LegendreSymbol(r,p) = epv

EXAMPLES :

> with(misc):
> quadresL(7,1);
                                    [1, 2, 4]

> quadresL(7,-1);
                                    [3, 5, 6]
```

FUNCTION : misc[SaveAll] - save all variables in current session

CALLING SEQUENCE : SaveAll(fileName)

PARAMETERS : fileName - string

GLOBAL VARIABLES : none

SYNOPSIS :
Saves all variables in current session to file fileName

EXAMPLES :

```
> with(misc):
> SaveAll("04-26-20.m"):
```

DISCUSSION :

SEE ALSO :

```
FUNCTION :  misc[sieveqcheck] -

CALLING SEQUENCE : sieveqcheck(qfunc,p)


PARAMETERS :  qfunc - qseries
                  p - positive integer

GLOBAL VARIABLES : none

SYNOPSIS :
  Determine whether all the coefficients of qseries belong to
  one residue class mod p.


EXAMPLES :
> with(misc):
> with(qseries):
> read "f05func.m":
> g05:=series(f05Aq*etaq(q,1,10000),q,10001):
> siftfindrange(g05,19,10000);
      ["minnops=", 15, "minres=", 9, "average=", 175., "maxnops=", 200]

> x2:=sift(g05,q,19,9,10000);
          494    456      437    399      380      323      285    209      190
x2 := -2 q    - q    + 2 q    + q    - 4 q    + 2 q    + 2 q    - q    + 2 q

        171      152      133    57      38
     + q    - 2 q    - 2 q    - q   + 3 q   - 1

> sieveqcheck(x2,19);
                                    true

DISCUSSION :

SEE ALSO :  siftfindrange
```

FUNCTION : misc[siftfind] - find number of terms for each residue mod p of exponents

CALLING SEQUENCE : siftfind(X,p,T)

PARAMETERS : X - qseries
p - positive integer
T - positive integer

GLOBAL VARIABLES : none

SYNOPSIS :
Returns seq([j,nops(sift(X,q,p,j,T))],j=0..p-1);

EXAMPLES :

```
> with(misc):
> with(qseries):
> read "f05func.m":
> g05:=series(f05Aq*etaq(q,1,10000),q,10001):
> siftfind(g05,19,10000);
[0, 200], [1, 185], [2, 181], [3, 176], [4, 179], [5, 186], [6, 174], [7, 180],
    [8, 188], [9, 15], [10, 180], [11, 189], [12, 185], [13, 180], [14, 185],
    [15, 184], [16, 181], [17, 187], [18, 189]

> siftfindrange(g05,19,10000);
       ["minnops=", 15, "minres=", 9, "average=", 175., "maxnops=", 200]
```

DISCUSSION :

SEE ALSO : siftfindrange

```
FUNCTION :  misc[siftfindrange] - find residue mod p of exponents
                                  with least and most terms


CALLING SEQUENCE :  siftfindrange(X,p,T)


PARAMETERS :    X -  qseries
                p -  positive integer
                T -  positive integer

GLOBAL VARIABLES : none

SYNOPSIS :
   Finds min nops(sift(X,q,p,r,T)) for r=0..p-1
   and   max nops(sift(X,q,p,r,T)) for r=0..p-1
   and the average



EXAMPLES :
> with(misc):
> with(qseries):
> read "f05func.m":
> g05:=series(f05Aq*etaq(q,1,10000),q,10001):
> siftfindrange(g05,19,10000);
       ["minnops=", 15, "minres=", 9, "average=", 175., "maxnops=", 200]

> x2:=sift(g05,q,19,9,10000);
          494    456      437    399      380      323      285    209      190
x2 := -2 q    - q    + 2 q    + q    - 4 q    + 2 q    + 2 q    - q    + 2 q

        171      152      133    57      38
     + q    - 2 q    - 2 q    - q   + 3 q   - 1

DISCUSSION :

SEE ALSO :  sieveqcheck
```

FUNCTION : misc[sqaureit] - complete square in binary quadratic form

CALLING SEQUENCE : squareit(qexp,var1,var2)

PARAMETERS : qexp - binary quadratic form in variables var1 and var2
-

GLOBAL VARIABLES : none

SYNOPSIS :
Complete square in binary quadratic form

EXAMPLES :

```
> with(misc):
> squareit( n*(5*n+1)/2-j^2,n,j);
                                  2
                          5 (n + 1/10)     2
                          ------------- - j  - 1/40
                                2
```

DISCUSSION :

SEE ALSO :