FUNCTION : tcore[PHI1] - PHI1 is Bijection 1 of GKS.

CALLING SEQUENCE : PHI1(ptn,p)

PARAMETERS : ptn - partition
p - positive integer

GLOBAL VARIABLES : NONE

SYNOPSIS : PHI1(ptn,p) = [pcore, pquotient]
where pcore is the p-core of partition ptn
and pquotient is the p-quotient.

EXAMPLES :

```
> with(tcore):

> testptn:=[1, 1, 2, 4, 4, 5, 6, 6, 6, 7, 7, 7, 7, 7, 8, 9, 9, 10, 10, 13, 16, 17];
  testptn := [1, 1, 2, 4, 4, 5, 6, 6, 6, 7, 7, 7, 7, 7, 8, 9, 9, 10, 10, 13, 16, 17]

> PHI1(testptn,5);
      [[2, 2, 2, 4, 4, 7, 11], [[], [1, 2, 5], [3], [2, 3, 3], [1, 1, 1, 2, 2]]]
```

DISCUSSION : We see that 5-core of testptn is [2, 2, 2, 4, 4, 7, 11]
and the 5-quotient is
[[], [1, 2, 5], [3], [2, 3, 3], [1, 1, 1, 2, 2]]

SEE ALSO : tcoreofptn, tquot and invphi1

```
FUNCTION:  tcore[addrimcell] - change cell to * on rim

CALLING SEQUENCE:  addrimcell(L)

PARAMETERS:   L - [diagarray,[i,j]]

GLOBAL VARIABLES:
SYNOPSIS:
   [i,j] is position of current cell
   We put * in next cell on the rim
   OUTPUT = [diagarray, [ni,nj]]
   Diagarray is new darray with * in position [ni,nj] (next position on rim)
   NOTE: This function is used by markrimhookV2

EXAMPLES:

> with(tcore):
> ptn1:=[1,1,2,4,4,5,6,6,6];
                      ptn1 := [1, 1, 2, 4, 4, 5, 6, 6, 6]

> darray1:=tcore[tresdiag2array](ptn1,5):
> op(darray1);
[[0, 1, 2, 3, 4, 0], [4, 0, 1, 2, 3, 4], [3, 4, 0, 1, 2, 3], [2, 3, 4, 0, 1],
    [1, 2, 3, 4], [0, 1, 2, 3], [4, 0], [3], [2]]

> printdarray(darray1);
0 1 2 3 4 0
4 0 1 2 3 4
3 4 0 1 2 3
2 3 4 0 1
1 2 3 4
0 1 2 3
4 0
3
2
> darray2:=addrimcell([darray1,[4,4]]):
> op(darray2);
                               diagarray, [4, 5]

> op(darray2[1]);
[[0, 1, 2, 3, 4, 0], [4, 0, 1, 2, 3, 4], [3, 4, 0, 1, 2, 3], [2, 3, 4, 0, "*"],
    [1, 2, 3, 4], [0, 1, 2, 3], [4, 0], [3], [2]]

> printdarray(darray2[1]);
0 1 2 3 4 0
4 0 1 2 3 4
3 4 0 1 2 3
2 3 4 0 *
1 2 3 4
0 1 2 3
4 0
3
2

DISCUSSION:  * marks next cell on rim in position [4,4]
             * is at position [4,5]

SEE ALSO:  markrimhookV2, printdarray
```

```
FUNCTION :  tcore[addrimthook] - add rim t-hook to partition


CALLING SEQUENCE : addrimthook(ptn,j,L,t)


PARAMETERS :   ptn -  partition
               j   -  positive integer
               L   -  positive integer
               t   -  positive integer

GLOBAL VARIABLES : none

SYNOPSIS :  addrimthook(ptn,j,L,t) adds rim-hook of length L*t
            to the partition ptn starting at part j.


EXAMPLES :
> with(tcore):
> t := 5;
                                    t := 5

> BV:= [[4, 8, 12], [[1], [4, 4, 4], [1], [1], [1]]];
              BV := [[4, 8, 12], [[1], [4, 4, 4], [1], [1], [1]]]

>      tc:=BV[1]:
>      ntc:=nops(tc):
>      ptn:=[seq(tc[ntc-j+1],j=1..ntc)]:
>      tq:=BV[2]:
>      for j from t by -1 to 1 do
>          tqj:=tq[j];
>          r:=j-1:  #residue
>       ntqj:=nops(tqj):
>       if ntqj>0 then
>         for k from ntqj by -1 to 1 do
>           p:=tqj[k]:
>           posi:=tcore[findhookinpos](ptn,t,r,p);
>           print("j=",j,"p=",p,"posi=",posi);
>            newptn:=tcore[addrimthook](ptn,posi,p,t);
>           print("newptn=",newptn);
>           ptn:=newptn:
>         od:
>       fi:
>      od:
                        "j=", 5, "p=", 1, "posi=", 6

                      "newptn=", [12, 8, 4, 3, 1, 1]

                        "j=", 4, "p=", 1, "posi=", 7

                    "newptn=", [12, 8, 4, 3, 3, 2, 2]

                        "j=", 3, "p=", 1, "posi=", 8

                   "newptn=", [12, 8, 4, 3, 3, 3, 3, 3]

                        "j=", 2, "p=", 4, "posi=", 7

                  "newptn=", [17, 13, 9, 5, 4, 4, 4, 3]

                        "j=", 2, "p=", 4, "posi=", 4

                 "newptn=", [22, 18, 14, 10, 4, 4, 4, 3]

                        "j=", 2, "p=", 4, "posi=", 4

                 "newptn=", [27, 23, 19, 15, 4, 4, 4, 3]

                        "j=", 1, "p=", 1, "posi=", 8
```

```
                    "newptn=", [27, 23, 19, 15, 5, 5, 5, 5]

>       nn:=tcore[np](ptn):
>       optn:=[seq(ptn[nn-j+1],j=1..nn)]:
>       print(optn);
                            [5, 5, 5, 5, 15, 19, 23, 27]

> PHI1(optn,5);
                  [[4, 8, 12], [[1], [4, 4, 4], [1], [1], [1]]]
DISCUSSION :
  In the example we illustrated how the inverse map invphi1 works.
  We started with
   BV:= [[4, 8, 12], [[1], [4, 4, 4], [1], [1], [1]]]
  which is a given 5-core and 5-quotient.
  We looped through the steps that adds 5-rim-hooks
  to get corresponding partition
                    optn = [5, 5, 5, 5, 15, 19, 23, 27]
  We double check that PHI1(optn, 5) = BV

SEE ALSO :
```

```
FUNCTION:  tcore[avec2nvec] - convert alpha-vector to n-vector

CALLING SEQUENCE:  avec2nvec(avec)

PARAMETERS:   avec -  alpha-vector

SYNOPSIS:
    The alpha-vector of a t-core converted to a an n-vector
    NOTE: Only t=5 version implemented

EXAMPLES:

> with(tcore):
> ptn:=[1, 1, 1, 1, 3, 3, 3, 5, 6];
                        ptn := [1, 1, 1, 1, 3, 3, 3, 5, 6]

> ptnnorm(ptn);
                                      24

> istcore(ptn,5);
                                     true

> nv:=ptn2nvec(ptn,5);
                             nv := [2, -2, -1, 1, 0]

> av:=nvec2alphavec(nv);
                              av := [2, 0, -1, 0, 0]

> avec2nvec(av);
                                [2, -2, -1, 1, 0]

DISCUSSION:

SEE ALSO:   nvec2alphavec
```

```
FUNCTION:  tcore[aveccyc] - cyclic shift of alpha-vector

CALLING SEQUENCE:   aveccyc(av)

PARAMETERS:     av -  alpha-vector

GLOBAL VARIABLES:
SYNOPSIS:
    Cyclically permute components of av (alpha-vector)

EXAMPLES:

> with(tcore):
> ptn:=[1,2,6]:
> istcore(ptn,5);
                                      true

> nv:=ptn2nvec(ptn,5);
                            nv := [2, 0, -1, 0, -1]

> av:=nvec2alphavec(nv);
                             av := [1, 0, -1, 0, 1]

> for j from 1 to 5 do
>   newav:=aveccyc(av):
>   newnv:=avec2nvec(newav):
>   newptn:=nvec2ptn(newnv):
>   print(newptn,newnv,newav):
>   av:=newav:
> od:
                 [2, 2, 5], [1, 0, -1, -1, 1], [1, 1, 0, -1, 0]

              [1, 1, 1, 1, 5], [-1, 0, 0, 0, 1], [0, 1, 1, 0, -1]

              [1, 1, 1, 3, 3], [-1, 1, 1, 0, -1], [-1, 0, 1, 1, 0]

             [1, 1, 1, 1, 2, 3], [1, 0, 1, 0, -2], [0, -1, 0, 1, 1]

                 [1, 2, 6], [2, 0, -1, 0, -1], [1, 0, -1, 0, 1]

DISCUSSION:  [2,2,5] is the 5-core 5+2+2. We have calculated the orbit
             of the 5-cycle on this partition.

SEE ALSO:  nvec2alphavec, avec2nvec
```

FUNCTION: tcore[darray2ptn] - convert darray to partition

CALLING SEQUENCE: darray2ptn(L)

PARAMETERS: L - diagarray (array form of t-residue diagram)

GLOBAL VARIABLES:

SYNOPSIS:
Converts array L to partition

EXAMPLES:

```
> with(tcore):
> ptn := [1, 1, 2, 4, 4, 5, 6, 6, 6];
                        ptn := [1, 1, 2, 4, 4, 5, 6, 6, 6]

> darray:=tcore[tresdiag2array](ptn,5):
> printdarray(darray);
0 1 2 3 4 0
4 0 1 2 3 4
3 4 0 1 2 3
2 3 4 0 1
1 2 3 4
0 1 2 3
4 0
3
2
> darray2ptn(darray);
                           [1, 1, 2, 4, 4, 5, 6, 6, 6]
```

DISCUSSION:

SEE ALSO: tresdiag2array, printdarray

```
FUNCTION:  tcore[findcell] - find cell to remove rim hook

CALLING SEQUENCE:    findcell(darray,t,r,k)

PARAMETERS:  darray - array of t-residue diagram
                 t  - positive integer
                 r  - positive integer (region number)
                 k  - element of {0,1,...,t-1}
GLOBAL VARIABLES:
SYNOPSIS:
  Find row and column number where to remove rim hook
  corresponding to an N in W[k] in region r

EXAMPLES:

> with(tcore):
> ptn1:=[1,1,2,4,4,5,6,6,6];
                      ptn1 := [1, 1, 2, 4, 4, 5, 6, 6, 6]

> darray1:=tcore[tresdiag2array](ptn1,5):
> tcore[printdarray](darray1);
0 1 2 3 4 0
4 0 1 2 3 4
3 4 0 1 2 3
2 3 4 0 1
1 2 3 4
0 1 2 3
4 0
3
2
> tcore[makebiw](ptn1,5,3);
   -3-2-1 0 1 2 3
W0  E E E E N E N
W1  E E N N E N N
W2  E E E N N N N
W3  E E E E E N N
W4  E E N E E N N
> findcell(darray1,5,1,0);
                                      [4, 4]

> newone1:=markrimhookV2([darray1,[4,4]],5):
> tcore[printdarray](newone1);
0 1 2 3 4 *
4 0 1 2 3 *
3 4 0 1 * *
2 3 4 0 *
1 2 3 4
0 1 2 3
4 0
3
2
DISCUSSION:  In W0 there is an N (preceeding E) in region 1
             Findcell gave [4,4] so in cell in next of rim
             is where rim hook can be removed (marked by *)

SEE ALSO:  markrimhookV2
```

FUNCTION : tcore[findhookinpos] - find where to insert rim-hook

CALLING SEQUENCE : findhookinpos(ptn,t,w,p)

PARAMETERS : ptn - partition
t,p - positive integers
w - w=0,1,...,t-1

GLOBAL VARIABLES : NONE

SYNOPSIS : findhookinpos(ptn,t,w,p) = f
where f is the part number of the partition ptn where
can insert a rim pt-hook from word W[w]

EXAMPLES :

```
> with(tcore):
```

DISCUSSION :

SEE ALSO : _

FUNCTION : tcore[invphi1] - inverse of the PHI1 map

CALLING SEQUENCE : invphi1(bigvec,t)

PARAMETERS : bigvec - [tcore, tquoient]
t - positive integer

GLOBAL VARIABLES : NONE

SYNOPSIS : nvphi1(bigvec,t) = ptn
where PHI1(ptn) = bigvec

EXAMPLES :

```
> with(tcore):
```

DISCUSSION :

SEE ALSO : _

FUNCTION: tcore[istcore] - Determine whether a p-core

CALLING SEQUENCE: istcore(ptn,p)

PARAMETERS: ptn - partition
p - positive integer

GLOBAL VARIABLES:
SYNOPSIS:
Ddetermines if ptn is a p-core

EXAMPLES:

```
> with(tcore):
> ptn9:=combinat[partition](9):
> tc9:=select(ptn->istcore(ptn,5),ptn9);
tc9 := [

    [1, 1, 1, 1, 2, 3], [1, 1, 1, 3, 3], [1, 1, 1, 1, 5], [2, 2, 5], [1, 2, 6]]

> nops(ptn9),nops(tc9);
                                    30, 5
```

DISCUSSION: There are only five of the 30 partitions of 9 are 5-cores.

SEE ALSO: tcores

```
FUNCTION:  tcore[makebiw] - make bi-infinite word

CALLING SEQUENCE:  makebiw(ptn,t,mj)

PARAMETERS:   ptn  -  partition
               t   -  positive integer
               mj  -  positive integer
GLOBAL VARIABLES:
SYNOPSIS:
   Make the bi-infinite words W[0],W[1],...W[t-1]
   with j from -mj to mj

EXAMPLES:

> with(tcore):
> ptn1:=[1,1,2,4,4,5,6,6,6];
                      ptn1 := [1, 1, 2, 4, 4, 5, 6, 6, 6]

> darray1:=tcore[tresdiag2array](ptn1,5):
> tcore[printdarray](darray1);
0 1 2 3 4 0
4 0 1 2 3 4
3 4 0 1 2 3
2 3 4 0 1
1 2 3 4
0 1 2 3
4 0
3
2
> tcore[makebiw](ptn1,5,3);
   -3-2-1 0 1 2 3
W0  E E E E N E N
W1  E E N N E N N
W2  E E E N N N N
W3  E E E E E N N
W4  E E N E E N N

DISCUSSION:  Only shows columns j=-3 .. 3

SEE ALSO:
```

```
FUNCTION:  tcore[markrimhookV2] - mark rim-hook to be removed

CALLING SEQUENCE:    markrimhookV2(L,N)

PARAMETERS:    L   -  [diagarray, [i,j]]
               N   -  positive integer

GLOBAL VARIABLES:
SYNOPSIS:
  diagarray is array of the t-residue diagram of a partition
  [i,j] is the cell position of N cell indicating where
  a rim hook of length N can be removed

EXAMPLES:

> with(tcore):
> ptn1:=[1,1,2,4,4,5,6,6,6];
                      ptn1 := [1, 1, 2, 4, 4, 5, 6, 6, 6]

> darray1:=tcore[tresdiag2array](ptn1,5):
> tcore[printdarray](darray1);
0 1 2 3 4 0
4 0 1 2 3 4
3 4 0 1 2 3
2 3 4 0 1
1 2 3 4
0 1 2 3
4 0
3
2
> tcore[makebiw](ptn1,5,3);
   -3-2-1 0 1 2 3
W0  E E E E N E N
W1  E E N N E N N
W2  E E E N N N N
W3  E E E E E N N
W4  E E N E E N N
> findcell(darray1,5,1,0);
                                      [4, 4]

> newone1:=markrimhookV2([darray1,[4,4]],5):
> tcore[printdarray](newone1);
0 1 2 3 4 *
4 0 1 2 3 *
3 4 0 1 * *
2 3 4 0 *
1 2 3 4
0 1 2 3
4 0
3
2
DISCUSSION: Rim hook of length 5 is marked by *.
            [4,4] is position of N cell next cell in
            the rim marks the first cell in the tail
            of the rim hook to be removed

SEE ALSO: findcell
```

```
FUNCTION:  tcore[nvec2alphavec] - Convert n-vector to alpha-vector

CALLING SEQUENCE: nvec2alphavec(nvec)

PARAMETERS:  nvec  -  n-vector (of length of t=5, 7, or 11)
                      and corresponding to a t-core of tn+delta.

GLOBAL VARIABLES:
SYNOPSIS:
    alpha-vector of t-core with given n-vector

EXAMPLES:

> with(tcore):
> f:=0:
> while f = 0 do
> ptn:=randpcore(5,12);
> n:=ptnnorm(ptn);
> if modp(n,5)=4 and n<30 then
>  f:=1:
> fi:
> end:
> istcore(ptn,5);
                                    true

> darray:=tresdiag2array(ptn,5):
> printdarray(darray);
0 1 2 3 4 0
4 0 1 2 3
3 4 0
2 3 4
1 2 3
0
4
3
2
> makebiw(ptn,5,3);
   -3-2-1 0 1 2 3
W0  E E E E E E N
W1  E E N N N N N
W2  E E E N N N N
W3  E E E E E N N
W4  E E E E N N N
> print("ptn=",ptn);
                    "ptn=", [1, 1, 1, 1, 3, 3, 3, 5, 6]

> ptnnorm(ptn);
                                    24

> nv:=ptn2nvec(ptn,5);
                           nv := [2, -2, -1, 1, 0]

> nvec2alphavec(nv);
                               [2, 0, -1, 0, 0]

> tc4:=tcores(5,4);
            tc4 := [[1, 1, 1, 1], [1, 1, 2], [2, 2], [1, 3], [4]]

> tc2avec:=ptn->nvec2alphavec(ptn2nvec(ptn,5));
              tc2avec := ptn -> nvec2alphavec(ptn2nvec(ptn, 5))

> [seq(tc2avec(tc), tc in tc4)];
[[1, 0, 0, 0, 0], [0, 1, 0, 0, 0], [0, 0, 0, 0, 1], [0, 0, 1, 0, 0],
    [0, 0, 0, 1, 0]]

DISCUSSION:  Found a randon p-core of 5n+4  (n<6) and its n-vector
             and then found its alpha-vector.
```

SEE ALSO: ptn2nvec, avec2nvec

```
FUNCTION:  tcore[nvec2ptn] - Convert n-vector to partition

CALLING SEQUENCE:  nvec2ptn(nvec)

PARAMETERS:  nvec  -  n-vector (list of integers)

GLOBAL VARIABLES:
SYNOPSIS:
      Returns partition (t-core) with given n-vector

EXAMPLES:

> with(tcore):
> ptn:=randpcore(5,6);
                     ptn := [2, 3, 5, 9, 13, 17, 21, 25]

> istcore(ptn,5);
                                       true

> darray:=tresdiag2array(ptn,5):
> printdarray(darray);
0 1 2 3 4 0 1 2 3 4 0 1 2 3 4 0 1 2 3 4 0 1 2 3 4
4 0 1 2 3 4 0 1 2 3 4 0 1 2 3 4 0 1 2 3 4
3 4 0 1 2 3 4 0 1 2 3 4 0 1 2 3 4
2 3 4 0 1 2 3 4 0 1 2 3 4
1 2 3 4 0 1 2 3 4
0 1 2 3 4
4 0 1
3 4
> makebiw(ptn,5,3);
   -3-2-1 0 1 2 3
W0  E E E N N N N
W1  E E E E N N N
W2  E E N N N N N
W3  E E N N N N N
W4  E E E E E E E
> nv:=ptn2nvec(ptn,5);
                            nv := [-1, 0, -2, -2, 5]

> nvec2ptn(nv);
                          [2, 3, 5, 9, 13, 17, 21, 25]

DISCUSSION:

SEE ALSO:  ptn2nvec
```

```
FUNCTION:  tcore[ptn2nvec] - n-vector of t-core

CALLING SEQUENCE:  ptn2nvec(ptn,p)

PARAMETERS:    ptn -  partition
                 p -  positice integer

GLOBAL VARIABLES:
SYNOPSIS:
     Computes n-vector of given p-core

EXAMPLES:

> with(tcore):
> ptn:=randpcore(5,6);
                      ptn := [2, 3, 5, 9, 13, 17, 21, 25]
> istcore(ptn,5);
                                          true
> darray:=tresdiag2array(ptn,5):
> printdarray(darray);
0 1 2 3 4 0 1 2 3 4 0 1 2 3 4 0 1 2 3 4 0 1 2 3 4
4 0 1 2 3 4 0 1 2 3 4 0 1 2 3 4 0 1 2 3 4
3 4 0 1 2 3 4 0 1 2 3 4 0 1 2 3 4
2 3 4 0 1 2 3 4 0 1 2 3 4
1 2 3 4 0 1 2 3 4
0 1 2 3 4
4 0 1
3 4
> makebiw(ptn,5,3);
   -3-2-1 0 1 2 3
W0  E E E N N N N
W1  E E E E N N N
W2  E E N N N N N
W3  E E N N N N N
W4  E E E E E E E
> ptn2nvec(ptn,5);
                                [-1, 0, -2, -2, 5]

DISCUSSION:

SEE ALSO:  randpcore, tresdiag2array, makebiw
```

```
FUNCTION:  tcore[ptn2rvec] - r-vector of partition

CALLING SEQUENCE:  ptn2rvec(ptn,p)

PARAMETERS:   ptn  -  partition
                p  -  positive integer

GLOBAL VARIABLES:
SYNOPSIS:
    compute r-vector of partition (mod p)

EXAMPLES:

> with(tcore):
> with(combinat):
> ptn:=randpart(36);
               ptn := [1, 1, 1, 1, 1, 1, 1, 3, 3, 3, 4, 4, 4, 8]

> istcore(ptn,5);
                                     false

> darray:=tresdiag2array(ptn,5):
> printdarray(darray);
0 1 2 3 4 0 1 2
4 0 1 2
3 4 0 1
2 3 4 0
1 2 3
0 1 2
4 0 1
3
2
1
0
4
3
2
> makebiw(ptn,5,3);
   -3-2-1 0 1 2 3
W0  E E E N E N N
W1  E N E E E N N
W2  E E E E E E N
W3  E E E E N N N
W4  E E N N N N N
> ptn2nvec(ptn,5);
                                [0, 0, 2, 0, -2]

> ptn2rvec(ptn,5);
                                [8, 8, 8, 6, 6]

DISCUSSION:  r-vector of the random partition mod 5 is [8, 8, 8, 6, 6]


SEE ALSO: ptn2nvec
```

FUNCTION: tcore[ptnnorm] - norm (sum of parts) of partition

CALLING SEQUENCE: ptnnorm(ptn)

PARAMETERS: ptn - partition

GLOBAL VARIABLES:

SYNOPSIS:
sum of parts of given partition

EXAMPLES:

```
> with(tcore):
> ptn1:=[1,1,2,4,4,5,6,6,6];
                      ptn1 := [1, 1, 2, 4, 4, 5, 6, 6, 6]

> ptnnorm(ptn1);
                                      35
```

DISCUSSION:

SEE ALSO:

```
FUNCTION:  tcore[randpcore] - random p-core

CALLING SEQUENCE:  randpcore(p,num)

PARAMETERS:    p   -  positive integer
               num -  integer > 1

GLOBAL VARIABLES:
SYNOPSIS:
    Generates a random p-core using random n-vector
    with entries between -num/2 to num/2

EXAMPLES:

> with(tcore):
> ptn:=randpcore(5,6);
                      ptn := [2, 3, 5, 9, 13, 17, 21, 25]

> istcore(ptn,5);
                                      true

> darray:=tresdiag2array(ptn,5):
> printdarray(darray);
0 1 2 3 4 0 1 2 3 4 0 1 2 3 4 0 1 2 3 4 0 1 2 3 4
4 0 1 2 3 4 0 1 2 3 4 0 1 2 3 4 0 1 2 3 4
3 4 0 1 2 3 4 0 1 2 3 4 0 1 2 3 4
2 3 4 0 1 2 3 4 0 1 2 3 4
1 2 3 4 0 1 2 3 4
0 1 2 3 4
4 0 1
3 4
> makebiw(ptn,5,3);
   -3-2-1 0 1 2 3
W0  E E E N N N N
W1  E E E E N N N
W2  E E N N N N N
W3  E E N N N N N
W4  E E E E E E E
> ptn2nvec(ptn,5);
                                [-1, 0, -2, -2, 5]

DISCUSSION: ptn is a random 5-core

SEE ALSO:  ptn2nvec, nvec2ptn
```

```
FUNCTION:  tcore[removerimhook] - remove rim hook

CALLING SEQUENCE:    removerimhook(darraymarked)

PARAMETERS: darraymarked - array (t-residue diagram)
                  -

GLOBAL VARIABLES:
SYNOPSIS:
      Remove hook corresponding to marks in darraymarked
      A result is another array

EXAMPLES:

> with(tcore):
> ptn1:=[1,1,2,4,4,5,6,6,6];
                      ptn1 := [1, 1, 2, 4, 4, 5, 6, 6, 6]

> darray1:=tcore[tresdiag2array](ptn1,5):
> tcore[printdarray](darray1);
0 1 2 3 4 0
4 0 1 2 3 4
3 4 0 1 2 3
2 3 4 0 1
1 2 3 4
0 1 2 3
4 0
3
2
> tcore[makebiw](ptn1,5,3);
   -3-2-1 0 1 2 3
W0  E E E E N E N
W1  E E N N E N N
W2  E E E N N N N
W3  E E E E E N N
W4  E E N E E N N
> findcell(darray1,5,1,0);
                                      [4, 4]

> newone1:=markrimhookV2([darray1,[4,4]],5):
> tcore[printdarray](newone1);
0 1 2 3 4 *
4 0 1 2 3 *
3 4 0 1 * *
2 3 4 0 *
1 2 3 4
0 1 2 3
4 0
3
2
> darray2:=removerimhook(newone1);

> tcore[printdarray](darray2);
0 1 2 3 4
4 0 1 2 3
3 4 0 1
2 3 4 0
1 2 3 4
0 1 2 3
4 0
3
2
DISCUSSION:  5-rim hook has been removed

SEE ALSO:  markrimhookV2, printdarray
```

```
FUNCTION:  tcore[rvec] - component of r-vector

CALLING SEQUENCE:  rvec(ptn,p,k)

PARAMETERS:    ptn -  partition
                 p -  positive integer
                 k -  residue mod p

GLOBAL VARIABLES:
SYNOPSIS:
 Number of nodes in p-residue diagram of ptn colored k.
 Used in the ptn2rvec function

EXAMPLES:

> with(tcore):
> with(combinat):
> ptn:=randpart(28);
                            ptn := [1, 1, 1, 8, 17]

> ptn2rvec(ptn,5);
                                [6, 7, 5, 5, 5]

> rvec(ptn,5,0);
                                       6

> seq(rvec(ptn,5,k),k=0..4);
                                 6, 7, 5, 5, 5

DISCUSSION: r[0]=6 for the given random partition

SEE ALSO: ptn2rvec
```

```
FUNCTION:  tcore[tcoreofptn] - t-core of partition

CALLING SEQUENCE:  tcoreofptn(ptn,p)

PARAMETERS:    ptn -  partition
                 p -  positive integer

GLOBAL VARIABLES:
SYNOPSIS:
  Compute p-core of ptn

EXAMPLES:

> with(tcore):
> rptn100:=combinat[randpart](100);
      rptn100 := [1, 1, 1, 1, 1, 1, 3, 4, 4, 4, 4, 4, 4, 4, 5, 18, 20, 20]

> tcoreofptn(rptn100,5);
                            [1, 1, 1, 1, 2, 2, 3, 4]

DISCUSSION:  Computed the 5-core of a random partition of 100

SEE ALSO: rvec, nvec2ptn
```

```
FUNCTION:  tcore[tcores] - t-cores of n

CALLING SEQUENCE:  tcores(p,n)

PARAMETERS:     p  -  positive integer
                n  -  positive integer

GLOBAL VARIABLES:
SYNOPSIS:
   Return the p-cores of n

EXAMPLES:

> with(tcore):
> tc9:=tcores(5,9);
tc9 := [

    [1, 1, 1, 1, 2, 3], [1, 1, 1, 3, 3], [1, 1, 1, 1, 5], [2, 2, 5], [1, 2, 6]]

DISCUSSION:  Found the 5-cores of 9

SEE ALSO:
```

```
FUNCTION:  tcore[tquot] - t-quotient of a partition

CALLING SEQUENCE:  tquot(ptn,t)

PARAMETERS:   ptn  -  partition
                t  -  positive integer

GLOBAL VARIABLES:
SYNOPSIS:
 Computes t-quotient of ptn

EXAMPLES:

> with(tcore):
> testptn:=[1, 1, 2, 4, 4, 5, 6, 6, 6, 7, 7, 7, 7, 7, 8, 9, 9, 10, 10, 13, 16, 17];
testptn :=

    [1, 1, 2, 4, 4, 5, 6, 6, 6, 7, 7, 7, 7, 7, 8, 9, 9, 10, 10, 13, 16, 17]

> PHI1(testptn,5);
   [[2, 2, 2, 4, 4, 7, 11], [[], [1, 2, 5], [3], [2, 3, 3], [1, 1, 1, 2, 2]]]

> tquot(testptn,5);
               [[], [1, 2, 5], [3], [2, 3, 3], [1, 1, 1, 2, 2]]

> tcoreofptn(testptn,5);
                            [2, 2, 2, 4, 4, 7, 11]

DISCUSSION : We see that 5-core of testptn is [2, 2, 2, 4, 4, 7, 11]
             and the 5-quotient is
             [[], [1, 2, 5], [3], [2, 3, 3], [1, 1, 1, 2, 2]]

SEE ALSO : tcoreofptn and PHI1
```

FUNCTION: tcore[tresdiag2array] - t-residue diagram (array form)

CALLING SEQUENCE: tresdiag2array(ptn,t)

PARAMETERS: ptn - partition (weakly increasing list of pos integers)
t - positive integer

GLOBAL VARIABLES:
SYNOPSIS:
Make array form of t-residue diagram of a partition

EXAMPLES:

```
> with(tcore):
> ptn := [1, 1, 2, 4, 4, 5, 6, 6, 6];
                      ptn := [1, 1, 2, 4, 4, 5, 6, 6, 6]

> darray:=tcore[tresdiag2array](ptn,5):
> whattype(darray);
                                  array

> op(darray);
  [[0, 1, 2, 3, 4, 0], [4, 0, 1, 2, 3, 4], [3, 4, 0, 1, 2, 3], [2, 3, 4, 0, 1],
    [1, 2, 3, 4], [0, 1, 2, 3], [4, 0], [3], [2]]

> printdarray(darray);
0 1 2 3 4 0
4 0 1 2 3 4
3 4 0 1 2 3
2 3 4 0 1
1 2 3 4
0 1 2 3
4 0
3
2
```

DISCUSSION:

SEE ALSO: printdarray