FUNCTION: partitions[agcrank] - The Andrews-Garvan crank of a partition

CALLING SEQUENCE: agcrank(ptn)

PARAMETERS: ptn - partition (list of integers)

SYNOPSIS:
agcrank(ptn) returns the Andrews-Garvan crank of the partition ptn

EXAMPLES:

```
> with(combinat):
> with(partitions):
> ptn6:=partition(6):
> interface(rtablesize=11):
> PMAT:=Matrix(11,3):
> for j from 1 to 11 do ptn:=ptn6[j]:
> PMAT[j,1]:=ptn:
> PMAT[j,2]:=agcrank(ptn):
> PMAT[j,3]:=modp(agcrank(ptn),11):
> od:
> print(PMAT);
```

[1, 1, 1, 1, 1, 1]	-6	5
[1, 1, 1, 1, 2]	-4	7
[1, 1, 2, 2]	-2	9
[2, 2, 2]	2	2
[1, 1, 1, 3]	-3	8
[1, 2, 3]	1	1
[3, 3]	3	3
[1, 1, 4]	-1	10
[2, 4]	4	4
[1, 5]	0	0
[6]	6	6

DISCUSSION:
We compute the crank and crank mod 11 of each partition of 6
Observe there is exactly one partition for each reside of
the crank mod 11

SEE ALSO: drank

FUNCTION: partitions[briefptnshelp] - brief help of partitions function

CALLING SEQUENCE: briefptnshelp(func)

PARAMETERS: func - partitions function

GLOBAL VARIABLES: NONE

SYNOPSIS:
brief help for given partitions function

EXAMPLES:

```
> with(partitions):
> briefptnshelp(vpcrank);
```
FUNCTION: partitions[vpcrank] - vector partition crank

CALLING SEQUENCE: vpcrank(vptn)

PARAMETERS: vptn - vector partition [dptn, ptn, ptn]

GLOBAL VARIABLES: NONE

SYNOPSIS:
crank of vector ptn [P1,P2,P3] is #(P2)-#(P3)

SEE ALSO:

FUNCTION: partitions[drank] - Dyson rank of a partition

CALLING SEQUENCE: drank(ptn)

PARAMETERS: ptn - partition (list of integers)

SYNOPSIS:
drank(ptn) returns the Dyson rank of the partition ptn

EXAMPLES:

```
> with(combinat):
> with(partitions):
> ptn4:=partition(4):
> PMAT:=Matrix(5,3):
> for j from 1 to 5 do ptn:=ptn4[j]:
> PMAT[j,1]:=ptn:
> PMAT[j,2]:=drank(ptn):
> PMAT[j,3]:=modp(drank(ptn),5):
> od:
> print(PMAT);
```

$$\begin{bmatrix} [1,1,1,1] & -3 & 2 \\ [1,1,2] & -1 & 4 \\ [2,2] & 0 & 0 \\ [1,3] & 1 & 1 \\ [4] & 3 & 3 \end{bmatrix}$$

DISCUSSION: We calculate the rank of each partition of 4 together with the rank mod 5

SEE ALSO:

FUNCTION: partitions[lamPD] - Lambda of partition into distinct parts

CALLING SEQUENCE: lamPD(dptn)

PARAMETERS: dptn - partition into distinct parts (sequence of increasing integers)

SYNOPSIS:
lamPD(dptn) = 0 if the two largest parts of dptn are consecutive
otherwise equal to 1

NOTE: used in the computation of the overpartition crank

EXAMPLES:

```
> with(combinat):
> with(partitions):
> dptn8:=select(ptnDP,partition(8));
          dptn8 := [[1, 3, 4], [1, 2, 5], [3, 5], [2, 6], [1, 7], [8]]
> for ptn in dptn8 do
> print(ptn, "   ",lamPD(ptn));
> od:
                              [1, 3, 4], "   ", 0

                              [1, 2, 5], "   ", 1

                                [3, 5], "   ", 1

                                [2, 6], "   ", 1

                                [1, 7], "   ", 1

                                 [8], "   ", 1
```

DISCUSSION: Computed lambda of each of the 6 partitions
of 8 into distinct parts

SEE ALSO: overptncrank

FUNCTION: partitions[numLE] - Number of times the largest even part occurs

CALLING SEQUENCE: numLE(ptn)

PARAMETERS: ptn - partition

SYNOPSIS:
number of times the largest even part occurs

EXAMPLES:

DISCUSSION:

SEE ALSO:

FUNCTION: partitions[overptncrank] - overpartition crank

CALLING SEQUENCE: overptncrank(optn)

PARAMETERS: optn - list [dptn, ptn]
where dptn is a partition into distinct parts
and ptn is a partition

SYNOPSIS:
overptncrank(optn) returns the overpartition crank of the
overpartition optn

EXAMPLES:

```
> with(combinat):
> with(partitions):
> with(ocrank):
> CT4:=[seq(overptncrank(overptns(4)[k]),k=1..14)];
            CT4 := [-4, -2, 2, 0, 4, -3, 0, 3, -2, 2, -1, -1, 1, 1]

> seq(nops(select(x->if x=k then true else false fi,CT4)),k=-4..4);
                          1, 1, 2, 2, 2, 2, 2, 1, 1

> seq(MBAR(abs(k),4),k=-4..4);
                          1, 1, 2, 2, 2, 2, 2, 1, 1
```

DISCUSSION:
We calculated the 14 overpartitions of 4 and their
overpartition cranks. We also confirmed the values
of MBAR(m,n) for n = 4.
MBAR(m,n) = number of overpartitions of n with overpartition crank m

SEE ALSO: overptns, overptncrank

```
FUNCTION:  partitions[overptnrank] - overpartition rank

CALLING SEQUENCE:  overptnrank(optn)

PARAMETERS:    optn - LIST [dptn, ptn]
                      where dptn is a partition into distinct parts
                      and ptn is a partition

SYNOPSIS:
  overptnrank(optn) returns the overpartition rank of the
  of the overpartition optn

EXAMPLES:

> with(combinat):
> with(partitions):
> with(orank):
> RT4:=[seq(overptnrank(overptns(4)[k]),k=1..14)];
            RT4 := [-3, -1, 0, 1, 3, -3, -1, 1, -1, 0, -1, 1, 1, 3]

> seq(nops(select(x->if x=k then true else false fi,RT4)),k=-4..4);
                           0, 2, 0, 4, 2, 4, 0, 2, 0

> seq(NBAR(abs(k),4),k=-4..4);
                           0, 2, 0, 4, 2, 4, 0, 2, 0

DISCUSSION:
We calculated the 14 overpartitions of 4 and their
   overpartition ranks. We also confirmed the values
   of NBAR(m,n) for n = 4.
   NBAR(m,n) = number of overpartitions of n with overpartition rank m

SEE ALSO:  overptns, overptncrank
```

```
FUNCTION:  partitions[overptns] - Overpartitions of n

CALLING SEQUENCE:  overptns(n)

PARAMETERS:     n  -  nonnegative integer
                   -

GLOBAL VARIABLES:
SYNOPSIS:
   overptns(n) generates a list of the overpartitions of n
   Here an overpartition is an element of DP X P
   where DP is set of partitions into distinct parts and
   P is set of unrestricted partitions of n.

EXAMPLES:

> with(combinat):
> with(partitions):
> ovptns4:=overptns(4);
ovptns4 := [[[], [1, 1, 1, 1]], [[], [1, 1, 2]], [[], [2, 2]], [[], [1, 3]],

    [[], [4]], [[1], [1, 1, 1]], [[1], [1, 2]], [[1], [3]], [[2], [1, 1]],

    [[2], [2]], [[1, 2], [1]], [[3], [1]], [[1, 3], []], [[4], []]]

> nops(ovptns4);
                                    14

> for ovptn in ovptns4 do
> print(ovptn);
> od:
                              [[], [1, 1, 1, 1]]

                               [[], [1, 1, 2]]

                                 [[], [2, 2]]

                                 [[], [1, 3]]

                                  [[], [4]]

                               [[1], [1, 1, 1]]

                                [[1], [1, 2]]

                                  [[1], [3]]

                                [[2], [1, 1]]

                                  [[2], [2]]

                                [[1, 2], [1]]

                                  [[3], [1]]

                                 [[1, 3], []]

                                  [[4], []]

DISCUSSION:
   There are 14 overpartitions of 4.
   They are listed above.

SEE ALSO:  overptncrank, overptnrank
```

FUNCTION: partitions[PDP] - Number of partitions of n (hard way)

CALLING SEQUENCE: PDP(n)

PARAMETERS: n - nonnegative integer

GLOBAL VARIABLES:
SYNOPSIS:
PDP(n) computes p(D,n) the hard way (by counting a list of partitions)
p(D,n) is the number of partitions of n into distinct parts

EXAMPLES:

```
> with(qseries):
> with(partitions):
> PDP(9);
                                    8

> series(etaq(q,2,12)/etaq(q,1,12),q,11);
          2     3     4     5     6     7     8     9      10     11
 1 + q + q  + 2 q  + 2 q  + 3 q  + 4 q  + 5 q  + 6 q  + 8 q  + 10 q   + O(q  )
```

DISCUSSION:
We see that p(D,9) = 8.
Confirmed from the generating function

SEE ALSO: ptnDP

FUNCTION: partitions[POE] - OE(n)

CALLING SEQUENCE: POE(n)

PARAMETERS: n - positive integer

SYNOPSIS:
POE(n) computes OE(n) (hard way). This is the number of partitions of n in which every even part is below each odd part.

EXAMPLES:

```
> with(qseries):
> with(partitions):
> POE(12);
                                    30

> GENFUNC:=1+add(POE(n)*q^n,n=1..10);
GENFUNC :=

        10       9       8      7      6      5      4      3      2
    19 q   + 12 q  + 12 q  + 7 q  + 7 q  + 4 q  + 4 q  + 2 q  + 2 q  + q + 1

> prodmake(GENFUNC,q,10);
                                1
              ------------------------------------------
                         2         4         6         8
              (1 - q) (-q  + 1) (-q  + 1) (-q  + 1) (-q  + 1)
```

DISCUSSION:
OE(12) = 30 and the generating function seems like a nice product

SEE ALSO: ptnOE

FUNCTION: partitions[printptns] - print partitions

CALLING SEQUENCE: printptns(ptns)

PARAMETERS: ptns - list of partitions

GLOBAL VARIABLES:

SYNOPSIS:

Prints a list of ptns in standard form

EXAMPLES:

```
> with(combinat):
> with(partitions):
> ptns9:=partition(9):
> ptns1:=select(ptnDP,ptns9);
ptns1 := [[2, 3, 4], [1, 3, 5], [4, 5], [1, 2, 6], [3, 6], [2, 7], [1, 8], [9]]

> printptns(ptns1);
9
`8 + 1`
`7 + 2`
`6 + 3`
`6 + 2 + 1`
`5 + 4`
`5 + 3 + 1`
`4 + 3 + 2`
```

DISCUSSION:

SEE ALSO: standptn

FUNCTION: partitions[PRR] - number of Rogers-Ramanujan partitions

CALLING SEQUENCE: PRR(n)

PARAMETERS: n - positive integer
-

SYNOPSIS:
Computes the number of ptns of n in which difference between parts
is at least 2.

EXAMPLES:

```
> with(qseries):
> with(partitions):
> with(qseries):
> PRR(9);
                                    5

> JP:=JAC(0,5,infinity)/JAC(1,5,infinity):
> series(jac2series(JP,20),q,11);
           2    3      4      5      6      7      8      9      10      11
   1 + q + q  + q  + 2 q  + 2 q  + 3 q  + 3 q  + 4 q  + 5 q  + 6 q   + O(q  )
```

DISCUSSION:
Number of Rogers-Ramanujan partitions of 9 is 5

SEE ALSO: ptnRR

FUNCTION: partitions[PSCHUR] - number of Schur partitions

CALLING SEQUENCE: PSCHUR(n)

PARAMETERS: n - positive integer

SYNOPSIS:
Computes the number of ptns of n in which difference between parts is at least 3 and such that no two consecutive multiples of 3 occur as parts

EXAMPLES:

```
> with(qseries):
> with(partitions):
> PSCHURC(12);
                                     6

> GENFUNC:=1+add(PSCHURC(n)*q^n,n=1..40);
                 40        39        38        37        36        35       34
GENFUNC := 169 q   + 153 q   + 139 q   + 126 q   + 114 q   + 102 q   + 91 q

           33       32       31       30       29       28       27       26
     + 82 q   + 74 q   + 67 q   + 60 q   + 53 q   + 47 q   + 42 q   + 38 q

           25       24       23       22       21       20       19       18
     + 34 q   + 30 q   + 26 q   + 23 q   + 20 q   + 18 q   + 16 q   + 14 q

           17       16      15      14      13      12      11      10      9
     + 12 q   + 10 q   + 9 q   + 8 q   + 7 q   + 6 q   + 5 q   + 4 q   + 3 q

          8      7      6      5    4    3    2
     + 3 q  + 3 q  + 2 q  + 2 q  + q  + q  + q  + q + 1

> prodmake(GENFUNC,q,40);
              5         7         11         13         17         19
1/((1 - q) (-q  + 1) (-q  + 1) (-q   + 1) (-q   + 1) (-q   + 1) (-q   + 1)

       23         25         29         31         35         37
    (-q   + 1) (-q   + 1) (-q   + 1) (-q   + 1) (-q   + 1) (-q   + 1))

> prodmake(GENFUNC,q,40,list);
[-1, 0, 0, 0, -1, 0, -1, 0, 0, 0, -1, 0, -1, 0, 0, 0, -1, 0, -1, 0, 0, 0, -1, 0,

    -1, 0, 0, 0, -1, 0, -1, 0, 0, 0, -1, 0, -1, 0, 0]

> seq(PSCHURC(n),n=1..20);
        1, 1, 1, 1, 2, 2, 3, 3, 3, 4, 5, 6, 7, 8, 9, 10, 12, 14, 16, 18
```

DISCUSSION: Number of Schur partitions of 12 is 6. The generating function looks like a nice product.

SEE ALSO: ptnSCHUR

```
FUNCTION:  partitions[ptnCC] - partitions with part diff at least 3

CALLING SEQUENCE:  ptnCC(ptn)

PARAMETERS:    ptn -  partition

GLOBAL VARIABLES: NONE
SYNOPSIS:
   Returns true if ptn is a partition in which difference between parts
   is at least 3.

EXAMPLES:

> with(combinat):
> with(partitions):
> with(qseries):
> ptn1:=[1,3,6];
                                ptn1 := [1, 3, 6]

> ptnCC(ptn1);
                                      false

> ptn2:=[1,6,9];
                                ptn2 := [1, 6, 9]

> ptnCC(ptn2);
                                       true

> ptns:=partition(9):
> ptns1:=select(ptnCC,ptns);
                    ptns1 := [[3, 6], [2, 7], [1, 8], [9]]

> nops(ptns), nops(ptns1);
                                      30, 4

> GENFUNC:=1+add(nops(select(ptnCC,partition(n)))*q^n,n=1..30);
               30       29       28       27       26       25       24
GENFUNC := 66 q   + 58 q   + 52 q   + 46 q   + 41 q   + 36 q   + 32 q

           23       22       21       20       19       18       17       16
     + 28 q   + 25 q   + 22 q   + 19 q   + 17 q   + 15 q   + 13 q   + 11 q

           15      14      13      12      11      10      9      8      7
     + 10 q   + 8 q   + 7 q   + 6 q   + 5 q   + 4 q   + 4 q  + 3 q  + 3 q

          6      5    4    3    2
     + 2 q  + 2 q  + q  + q  + q  + q + 1

> prodmake(GENFUNC,q,30,list);
[-1, 0, 0, 0, -1, 0, -1, 0, -1, 1, -1, 0, -1, 1, -2, 1, -2, 1, -2, 2, -3,
  3, -3, 3, -5, 5, -6, 5, -8]

DISCUSSION:  ptn1 is not such a partition, ptn2 is
             Of the 30 partitions of 9 only 4 satisfy the condition
             The generating function does not look like a nice product

SEE ALSO: ptnDP, ptnRR
```

FUNCTION : pntDP - Returns true if a partition into distinct parts

CALLING SEQUENCE : pntDP(ptn)

PARAMETERS : ptn - partition (list of nonnegative integers)

SYNOPSIS :
pntDP(ptn) returns true if the partition ptn is a partition into
distinct parts.

EXAMPLES :

```
> with(combinat):
> read "FUNCS.txt":
> ptn1:=[1,3,6];
                              ptn1 := [1, 3, 6]
> ptnDP(ptn1);
                                     true
> ptn2:=[1,3,3,6];
                             ptn2 := [1, 3, 3, 6]
> ptnDP(ptn2);
                                    false
> ptns:=partition(9):
> ptns1:=select(ptnDP,ptns);
ptns1 := [[2, 3, 4], [1, 3, 5], [4, 5], [1, 2, 6], [3, 6], [2, 7], [1, 8], [9]]
> printptns(ptns1);
9
`8 + 1`
`7 + 2`
`6 + 3`
`6 + 2 + 1`
`5 + 4`
`5 + 3 + 1`
`4 + 3 + 2`
> quit
```

DISCUSSION :
ptn1=[1,3,6] is a partition into distinct parts but ptn2=[1,3,3,6]
is not. There are 8 partitions of 9 into distinct parts:
9, 8 + 1, 7 + 2, 6 + 3, 6 + 2 + 1, 5 + 4, 5 + 3 + 1,
and 4 + 3 + 2.

SEE ALSO : ptnOP

FUNCTION: partitions[ptnOE] - partitions enumerated by OE(n)

CALLING SEQUENCE: ptnOE(ptn)

PARAMETERS: ptn - partition

GLOBAL VARIABLES: NONE

SYNOPSIS:
Returns true if ptn is a partition has all even parts are less than all odd

EXAMPLES:

```
> with(combinat):
> with(partitions):
> with(qseries):
> ptns:=partition(8):
> ptns1:=select(ptnOE,ptns);
ptns1 := [[1, 1, 1, 1, 1, 1, 1, 1], [2, 2, 2, 2], [1, 1, 1, 1, 1, 3],
    [1, 1, 3, 3], [2, 3, 3], [2, 2, 4], [4, 4], [1, 1, 1, 5], [3, 5], [2, 6],
    [1, 7], [8]]

> nops(ptns), nops(ptns1);
                                    22, 12
```

DISCUSSION: Of the 22 oartitions of 8 only 12 satisfy the condition and they are listed

SEE ALSO: POE

```
FUNCTION :  pntOP - Returns true if a partition into odd parts


CALLING SEQUENCE :  pntOP(ptn)


PARAMETERS :   ptn   -  partition (list of nonnegative integers)

SYNOPSIS :
 pntOP(ptn) returns true if the partition ptn is a partition into
 odd parts.

EXAMPLES :
> with(combinat):
> read "FUNCS.txt":
> ptn1:=[1,3,6];
                              ptn1 := [1, 3, 6]
> ptnOP(ptn1);
                                    false
> ptn2:=[1,3,3,5];
                             ptn2 := [1, 3, 3, 5]
> ptnOP(ptn2);
                                     true
> ptns:=partition(9):
> ptns1:=select(ptnOP,ptns);
ptns1 := [[1, 1, 1, 1, 1, 1, 1, 1, 1], [1, 1, 1, 1, 1, 1, 3], [1, 1, 1, 3, 3],
    [3, 3, 3], [1, 1, 1, 1, 5], [1, 3, 5], [1, 1, 7], [9]]
> printptns(ptns1);
9
`7 + 1 + 1`
`5 + 3 + 1`
`5 + 1 + 1 + 1 + 1`
`3 + 3 + 3`
`3 + 3 + 1 + 1 + 1`
`3 + 1 + 1 + 1 + 1 + 1 + 1`
`1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1`

DISCUSSION :
The ptn1=[1,3,6] is not a partition into odd parts but the
partition ptn2=[1,3,3,5] is. There are 8 partitions of 9 into
odd parts:
9, 7 + 1 + 1, 5 + 3 + 1, 5 + 1 + 1 + 1 + 1, 3 + 3 + 3,
3 + 3 + 1 + 1 + 1, 3 + 1 + 1 + 1 + 1 + 1 + 1,
and 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1.


SEE ALSO :  ptnDP
```

FUNCTION: partitions[ptnRR] - Rogers-Ramanujan partitions

CALLING SEQUENCE: ptnRR(ptn)

PARAMETERS: ptn - partition

GLOBAL VARIABLES: NONE

SYNOPSIS:
Returns true if ptn is a partition in which difference between parts is at least 2.

EXAMPLES:

```
> with(combinat):
> with(partitions):
> ptns:=partition(8):
> ptns1:=select(ptnRR,ptns);
                    ptns1 := [[3, 5], [2, 6], [1, 7], [8]]

> ptns2:=select(ptn-> if convert(modp(ptn,5),set) subset {1,4} then true
  else false fi, ptns);
    ptns2 := [[1, 1, 1, 1, 1, 1, 1, 1], [1, 1, 1, 1, 4], [4, 4], [1, 1, 6]]

> nops(ptns), nops(ptns1), nops(ptns2);
                                  22, 4, 4
```

DISCUSSION: There are 22 partitions of 8.
Of these 4 satisfy the condition (listed)
We also find the partitions of 8 with parts congruent to 1 or 4 mod 5.

SEE ALSO: PRR

FUNCTION: partitions[ptnSCHUR] - Schur partitions

CALLING SEQUENCE: ptnSCHUR(ptn)

PARAMETERS: ptn - partition

GLOBAL VARIABLES: NONE

SYNOPSIS:
Returns true if ptn is a partition in which difference between parts
is at least 3 and such that no two consecutive multiples of 3
occur as parts

EXAMPLES:

```
> with(combinat):
> with(partitions):
> with(qseries):
> ptns:=partition(8):
> ptns1:=select(ptnSCHUR,ptns);
                        ptns1 := [[2, 6], [1, 7], [8]]

> ptns2:=select(ptn-> if convert(modp(ptn,6),set) subset {1,5} then true
        else false fi, ptns);
          ptns2 := [[1, 1, 1, 1, 1, 1, 1, 1], [1, 1, 1, 5], [1, 7]]

> nops(ptns), nops(ptns1), nops(ptns2);
                                  22, 3, 3
```

DISCUSSION: There are 22 partitions of 8.
Of these 3 satisfy the condition (listed)
We also find the partitions of 8 with parts congruent to
1 or 5 mod 6.

SEE ALSO: PSCHUR

```
FUNCTION :  ptnschanges - recent changes in partitions package


CALLING SEQUENCE :  ptnschanges()


PARAMETERS :      NONE

GLOBAL VARIABLES :  NONE

SYNOPSIS :  Lists recent changes in partitions package


EXAMPLES:

> with(partitions):
> ptnschanges();
************************************************************
*
*
*   thetaids package version 0.1 -  Wed, Apr 12, 2023  9:51:48 PM
*   thetaids package version 0.2 -  Sat, Apr 15, 2023  1:17:54 PM
*  This version tested on MAPLE 2022
*
*
*  Changes since previous version 0.1
*     * Fixed some bugs
        Make maple txt help files
*
*
*
************************************************************
DISCUSSION:

SEE ALSO:  ptnspversion
```

```
FUNCTION :  ptnspversion - version of partitions package

CALLING SEQUENCE :  ptnspversion()

PARAMETERS :      NONE

GLOBAL VARIABLES :  NONE

SYNOPSIS :  Returns version of partitions package

EXAMPLES:

> with(partitions):
> ptnspversion();
*****************************************************
*
*  partitions package version 0.2
*  Sat, Apr 15, 2023  1:17:54 PM
*  This version tested on MAPLE 2022
*
*  Please report any problems to fgarvan@ufl.edu
*  Previous versions:
      NONE
*
*  Please report any problems to fgarvan@ufl.edu
*  Previous versions:
       0.1 - Apr 2023 (MAPLE 2022)
*****************************************************
DISCUSSION:

SEE ALSO:  ptnschanges
```

FUNCTION: partitions[ptnsfunctions] - list functions in partitions package

CALLING SEQUENCE: ptnsfunctions()

PARAMETERS: NONE

GLOBAL VARIABLES: NONE

SYNOPSIS:
Lists functions in partitions package

EXAMPLES:

```
> with(partitions):
> ptnsfunctions();
[PDP, POE, PRR, PSCHUR, agcrank, briefptnshelp, drank, lamPD, numLE,
    overptncrank, overptnrank, overptns, printptns, ptnCC, ptnDP, ptnOE, ptnOP,
    ptnRR, ptnSCHUR, ptnschanges, ptnspversion, ptnsfunctions, standptn,
    vecptns, vecptnsC, vpcrank, vpw]
```

DISCUSSION:

SEE ALSO: briefptnshelp

FUNCTION: partitions[standptn] - print partition in standard form

CALLING SEQUENCE: standptn(ptn)

PARAMETERS: ptn - partition (list of nondecreasing positive integers)
-

GLOBAL VARIABLES: NONE

SYNOPSIS:
prints a ptn in standard form

EXAMPLES:

```
> with(partitions):
> ptn:=[1,1,2,2,2,5,6,8,8];
                        ptn := [1, 1, 2, 2, 2, 5, 6, 8, 8]

> standptn(ptn);
                        8 + 8 + 6 + 5 + 2 + 2 + 2 + 1 + 1
```

DISCUSSION:

SEE ALSO: printptns

FUNCTION: partitions[vecptns] - generate vector partitions

CALLING SEQUENCE: vecptns(n)

PARAMETERS: n - positive integer

GLOBAL VARIABLES: NONE

SYNOPSIS:
Generates a list of vector-partitions of n.
A vector partitions is an element of DP x P x P
where DP is the set of partitions into distinct parts, and
P is the set of unrestricted partitions

EXAMPLES:

```
> with(partitions):
> V:=vecptns(4):
> m:=matrix(nops(V),3):
> for i from 1 to nops(V) do
>     m[i,1]:=V[i]:
>     m[i,2]:=vpw(V[i]):
>     m[i,3]:=vpcrank(V[i]):
> od:
#Table of vector partitions of 4 with weight and crank
> print(m);
```

[[[], [], [1, 1, 1, 1]]	1	-4]
[[[], [], [1, 1, 2]]	1	-3]
[[[], [], [2, 2]]	1	-2]
[[[], [], [1, 3]]	1	-2]
[[[], [], [4]]	1	-1]
[[[], [1], [1, 1, 1]]	1	-2]
[[[], [1], [1, 2]]	1	-1]
[[[], [1], [3]]	1	0]
[[[], [1, 1], [1, 1]]	1	0]
[[[], [1, 1], [2]]	1	1]
[[[], [2], [1, 1]]	1	-1]
[[[], [2], [2]]	1	0]
[[[], [1, 1, 1], [1]]	1	2]
[[[], [1, 2], [1]]	1	1]
[[[], [3], [1]]	1	0]
[[[], [1, 1, 1, 1], []]	1	4]
[[[], [1, 1, 2], []]	1	3]
[[[], [2, 2], []]	1	2]
[[[], [1, 3], []]	1	2]
[[[], [4], []]	1	1]
[[[1], [], [1, 1, 1]]	-1	-3]
[[[1], [], [1, 2]]	-1	-2]

```
[    [[1], [], [3]]        -1    -1]
[                                  ]
[  [[1], [1], [1, 1]]      -1    -1]
[                                  ]
[   [[1], [1], [2]]        -1     0]
[                                  ]
[  [[1], [1, 1], [1]]      -1     1]
[                                  ]
[   [[1], [2], [1]]        -1     0]
[                                  ]
[ [[1], [1, 1, 1], []]     -1     3]
[                                  ]
[  [[1], [1, 2], []]       -1     2]
[                                  ]
[    [[1], [3], []]        -1     1]
[                                  ]
[  [[2], [], [1, 1]]       -1    -2]
[                                  ]
[    [[2], [], [2]]        -1    -1]
[                                  ]
[   [[2], [1], [1]]        -1     0]
[                                  ]
[  [[2], [1, 1], []]       -1     2]
[                                  ]
[    [[2], [2], []]        -1     1]
[                                  ]
[  [[1, 2], [], [1]]        1    -1]
[                                  ]
[    [[3], [], [1]]        -1    -1]
[                                  ]
[  [[1, 2], [1], []]        1     1]
[                                  ]
[    [[3], [1], []]        -1     1]
[                                  ]
[   [[1, 3], [], []]        1     0]
[                                  ]
[    [[4], [], []]         -1     0]
```

DISCUSSION:
  This table of the 41 vector partitions of 4 with weight and crank

SEE ALSO: vpcrank, vpw

```
FUNCTION:  partitions[vecptnsC] - vector partitions with given crank class

CALLING SEQUENCE:  vecptnsC(n,k,t)

PARAMETERS:    n -  positive integer
               t -  positive integer
               k -  residue mod t

GLOBAL VARIABLES:
SYNOPSIS:
       #GENERATES vector partitions of n with crank congruent to
       #k  mod t

EXAMPLES:

> with(partitions):
> V405:=vecptnsC(4,0,5);
V405 := [[[], [1], [3]], [[], [1, 1], [1, 1]], [[], [2], [2]], [[], [3], [1]],
    [[1], [1], [2]], [[1], [2], [1]], [[2], [1], [1]], [[1, 3], [], []],
    [[4], [], []]]

> nops(V405);
                                      9
> add(vpw(vp), vp in V405);
                                      1

DISCUSSION:  There are 9 vector partitions of 4 with crank congruent to
             0 mod 5 with total weight 1

SEE ALSO:  vectptns, vpcrank, vpw
```

```
FUNCTION:  partitions[vpcrank] - vector partition crank

CALLING SEQUENCE:  vpcrank(vptn)

PARAMETERS:    vptn - vector partition [dptn, ptn, ptn]

GLOBAL VARIABLES: NONE

SYNOPSIS:
    crank of vector ptn [P1,P2,P3] is #(P2)-#(P3)

EXAMPLES:

> with(partitions):
> V3:=vecptns(3);
V3 := [[[], [], [1, 1, 1]], [[], [], [1, 2]], [[], [], [3]], [[], [1], [1, 1]],
    [[], [1], [2]], [[], [1, 1], [1]], [[], [2], [1]], [[], [1, 1, 1], []],
    [[], [1, 2], []], [[], [3], []], [[1], [], [1, 1]], [[1], [], [2]],
    [[1], [1], [1]], [[1], [1, 1], []], [[1], [2], []], [[2], [], [1]],
    [[2], [1], []], [[1, 2], [], []], [[3], [], []]]

> nops(V3);
                                    19

> [seq(vpcrank(vp), vp in V3)];
        [-3, -2, -1, -1, 0, 1, 0, 3, 2, 1, -2, -1, 0, 2, 1, -1, 1, 0, 0]

DISCUSSION: We calculate the vector partition crank of the 19
            vector partitions of 3

SEE ALSO: vecptns, vecptnsC, vpw
```

FUNCTION: partitions[vpw] - vector partition weight

CALLING SEQUENCE: vpw(vptn)

PARAMETERS: vptn - vector partition [dptn, ptn, ptn]

GLOBAL VARIABLES: NONE

SYNOPSIS:
weight of vector ptn [P1,P2,P3] = (-1)^#(P1)

EXAMPLES:

```
> with(partitions):
> V:=vecptns(4):
> nops(V);
                                    41

> add(vpw(vp), vp in V);
                                    5
```

DISCUSSION: The total weight of the 41 vector partitions of 4 is 5.
5 = p(4) as expected.

SEE ALSO: vectptns, vecptnsC, vpcrank