NB. thnombres.ijs R.Coquidé (27/11/2019)

NB. Théorie des nombres

require 'users~/boitaoutils.ijs'

NB. Sauf indication contraire, y est un entier >0 dans ce qui suit

NB. GENERALITES

NB. R =. pppsey y : plus petit nb premier sup ou eg à y
```
pppsey=: (*1&p:)+4&p:*0&p:
```
NB. R =. pppssy y : plus petit nb premier strictement sup à y
```
pppssy=: 4&p:
```
NB. R =. pgpiey y : plus grand nb premier inf ou eg à y (1 si y=1)
```
pgpiey=: (*1&p:)+_4&p: ::1:"0*0&p:
```
NB. R =. pgpsiy y : plus grand nb premier strictement inf à y (1 si y=1 ou 2)
```
pgpsiy=: _4&p: ::1:"0
```
NB. [x] prem y : retourne 1 si y premier(ou x et y premiers entre eux), 0 sinon
```
prem =: ((1&p:) :(1=+.))"0 0 0
```
NB. R =. pgcd v : pgcd suite d'entiers ; R=. x pgcd y : pgcd de x et y
```
pgcd =: +./ :+."_ 0 0
```
NB. R =. ppcm V : ppcm suite d'entiers ; R=. x ppcm y : ppcm de x et y
```
ppcm =: *./ :*."_ 0 0
```
NB. R =.dpgcd v : Division des composantes d un vecteur V par leur pgcd
```
dpgcd =: ]%+./
```
NB. R =. dppcm V : division du ppcm des composantes du vecteur V par celles-ci
```
dppcm =: *./%]
```
NB. R =. p valp N : Valuation p-adique de N ;p premier ; N entier > 0
```
valp =: 4 : '(x i.~0{M){(1{M=. __ q: y),0'"0 0
```
NB. ex : 5 valp !20

NB. 4

NB. R =. abond y : abondance de y

NB. =0 (parfait) ; >0 (abondant)

NB. =_1 (presque parfait); <0 (déficient)
```
abond =: (sds-)"0
```
NB. Soit S la somme des diviseurs stricts d un entier>0

NB. R =. dpa y : _1 (nombre déficient: S<y)

NB. 0 (nombre parfait : S=y)

NB. 1 (nombre abondant : S>y)
```
dpa =: *@abond"0
```

NB. LISTES

NB. R =. [x] lnp y : liste des nb premiers (ou premiers avec x) et <: y
```
lnp =: p:@i.@(p:^:_1)@>:"0 :((1=[+.1+[:i.])#1+[:i.])"0 0
```
NB. R =.ldv y : liste des diviseurs de y
```
ldv =: 3 : '~.z,|.Y%z=.(0=z|Y)#z=.1x+i.x:<.%:Y=.x:y'"0
```
NB. R =.ldp y : liste des diviseurs premiers de y
```
ldp =: (1&p:@ldv)#ldv
```

NB. R =.lds y : liste des diviseurs stricts de y
```
lds =: }:@ldv"0
```
NB. R =. lparf y : liste des nombres parfaits inf ou eg à y
```
lparf =: (( 0:= abond@])#])@:>:@i.
```
NB. R =. lppar y : liste des nombres presque parfaits inf ou eg à y
```
lppar =: ((_1:= abond@])#])@:>:@i.
```
NB. R =. labon y : liste des nombres abondants inf ou eg à y
```
labon =: (( 0:< abond@])#])@:>:@i.
```
NB. R =. ldefi y : liste des nombres deficients inf ou eg à y
```
ldefi =: (( 0:> abond@])#])@:>:@i.
```
NB. R =. nbprem y : liste des nombres premiers inf ou eg à y
```
lprem =: p:@i.
```
NB.Liste des nombres pseudo premiers de base a inf ou eg à y

NB. R =. a lpseudop y
```
lpseudop =: 4 : '(0&p:z)#z=.(1=z&|(x:x)^<:z)#z=.>:i.+:<.-:<:x:y'
```
NB. R=. lwils y : liste des nombres de Wilson inf ou eg à y
```
lwils =: >:@!@i.@<:
```
NB. R =. lcull y : liste des nombres de Cullen inf ou eg à y
```
lcull =: [:>:]*2x&^
```

NB. DENOMBREMENTS

NB. R =. [x] nnp y : nombre des nb premiers (ou premiers avec x) et <: y
```
nnp =: pi =: ([:#lnp)"0 0 0
```
NB. R =. npy y : nb des nb premiers avec y et <: y (indicatrice ou ft Totient d'Euler)
```
npy =: 5&p:
```
NB. R =. ndv y : Nombre des diviseurs de y
```
ndv =: #@ldv"0
```
NB. R =.ndp y : Nombre des diviseurs premiers de y
```
ndp =: #@ldp"0
```
NB. R =.nds y : nombre de diviseurs stricts de y
```
nds =: #@lds"0
```
NB. R =. npart y : nombre de partitions de y
```
npart =: -/@(+/)@:($:"0)@(-(-:(*"1)_1 1+/3*])@(>:@i.@>.@%:@((2r3)&*)))
`(x:@(0&=))@.(0 >: ])M."0
```
NB. R=. nparf y : nombre de nb parfaits inf ou eg à y
```
nparf =: #@lparf"0
```
NB. R=. nppar y : nombre de nb presque parfaits inf ou eg à y
```
nppar =: #@lppar"0
```
NB. R=. nabon y : nombre de nb abondants inf ou eg à y
```
nabon =: #@labon"0
```
NB. R=. ndefi y : nombre de nb déficients inf ou eg à y
```
ndefi =: #@ldefi"0
```

NB. SOMMES

NB. R =.sdv y : somme des diviseurs de y
```
sdv =: +/@ldv"0
```

NB. R =. sdp y : Somme des diviseurs premiers de y
```
sdp =: +/@ldp"0
```
NB. R =. sds y : somme des diviseurs stricts de y
```
sds =: +/@lds"0
```
NB. R =. x sdx y : Somme des diviseurs de y élevés à la puissance x

NB. x généralement entier m peut être réel ou complexe
```
sdx =: ([:+/([:ldv])^[)"0 0
```
NB. R =. x ssx y : Idem diviseurs stricts
```
ssx =: ([:+/([:lds])^[)"0 0
```
NB. R =. x spx y : Idem diviseurs premiers
```
spx =: ([:+/([:ldp])^[)"0 0
```

NB. PRODUITS

NB. R =. pdv y : produit des diviseurs de y
```
pdv =: */@ldv"0
```
NB. R =. pdp y : Produit des diviseurs premiers de y (ou noyau)
```
pdp =: noy =: */@ldp"0
```
NB. R =. pds y : produit des diviseurs stricts de y
```
pds =: */@lds"0
```

NB. DIVERS

NB. R =. mp M : Sélection des lignes de même parité dans une matrice M de 2 colonnes
```
mp =: =/@:|:@:(2&|)#]
```
NB. R =. tnpy n ==> triplets de nombres de pythagore premiers entre eux

NB. (*:a) = (*:b) + (*:c)
```
tnpy =: 3 : '0 Tricol 1 Tricol~.Tdc dpgcd"1 y,.~(-:@(+/,.-/)@|:)mp
(Y%V),.V=.V{.~N=.<.-:#V=.ldv Y=.*:y'
```
NB. R =. tnpy N NB. Calcul à partir de l entier strictement positif N

NB. R =. TNPY N : idem nombres <: N
```
TNPY =: 3 : 0
N=. 3+i.y-2 [ R=.0 3$0
for_i. N do. R=.R,tnpy i end.
(*./|:R<:y)#R=.~. 0 Tricol 1 Tricol R
)
```

NB. QUELQUES FONCTIONS CLASSIQUES

NB. R =. UN y : Ft constante égale à 1 pour tout n entier strictement positif
```
UN =: 1:"0
```
NB. R =. EN y : Elément neutre : ft égale à 1 si y=1 et à 0 sinon
```
EN =: 1x=x:@] "0
```
NB. R =. ID y : Ft identité égale à n pour tout n
```
ID =: ]"0
```
NB. R =. k IDk n : Ft puissance égale à n^k (où k peut être réel ou complexe)
```
IDk =: (x:@])^[ "0 0
```

NB. R =. pd y ; plus petit diviseur autre que 1 (1 si y premier ou ègal à 1)
```
pd =: {.@(1:,~}:@q:)"0
```
NB. R =. gd y ; Plus grand diviseur autre que y (1 si y premier ou égal à 1)
```
gd =: */@}.@q:
```
NB. R =. Gomega y ; Grand Omega : nb total de facteurs premiers de y
```
Gomega=: ([:#q:)@x:"0
```
NB. R =. Pomega n ; Petit Omega : nb de diviseurs premiers
```
Pomega=: #@~.@q:@x:"0
```
NB. R =. Mu n ; Ft Mu de MOEBIUS
```
Mu =: ((Pomega@]=Gomega@])*_1x^Pomega@])@x:"0
```
NB. R =. Lambda n ; Ft Lambda de Liouville
```
Lambda=: _1x^Gomega"0
```
NB. R =. fi y ou R =. tot y : ft indicatrice ou totient d'Euler
```
fi =: tot =: 5&p:
```

NB. NOMBRES REMARQUABLES (indicés en origine 1)

NB. R =. NFER n n ieme nombre de FERMAT
```
NFER =: 3 : '1x+2x^2x^ x: y'"0
```
NB. R =. NMER n n ieme nombre de Mersenne
```
NMER =: 3 : '_1x+2x^p: x: y'"0
```
NB. R =. NMERp p nombre de MERSENNE correspondant à p premier
```
NMERp =: 3 : '_1x+2x^ x: y'"0
```
NB. On a (NMER n) = (NMERp p: n)

NB. R =. NFIB n : n ieme nombre de Fibonacci
```
NFIB =: 3 : 'if. 1>:y do. y else. (NFIB y-1)+NFIB y-2 end.'M."0
```
NB. R =. NLUC n : n ieme nombre de Lucas
```
NLUC =: 3 : 'if. 1>:y do. y{2 1x else. (NLUC y-1)+NLUC y-2 end.' M."0
```

NB. Conjonction transformant une ft arithmétique F en (F modulo n) où n entier >0
```
MODULO =: 2 : ('n|u x:y',MD,'n|(x: x) u x:y')
```
NB. Ex1 : SM7 =: + MODULO 7 "0 0   NB. Somme Modulo 7

NB. Ex2 : PM12 =: * MODULO 12 "0 0   NB. Produit modulo 12

NB. Ex 3 : Table des produits modulo 7 :

(* MODULO 7 "0 0) TA 'PM7' ~ 1+i.6

PM7	1	2	3	4	5	6
1	1	2	3	4	5	6
2	2	4	6	1	3	5
3	3	6	2	5	1	4
4	4	1	5	2	6	3
5	5	3	1	6	4	2
6	6	5	4	3	2	1

NB. Ex 4 : carrés modulo 7

NB.   (*: MODULO 7) 1+i.6

NB. 1 4 2 2 4 1

NB. h =: f pced g"0 : Produit de Composition de Euler-Dirichlet de 2 fts arithmétiques

```
PCED =: 2 : '+/(u z)*v |.z=.ldv y=.x: y'
```

NB. Ex : H =: pd PCED fi "0

NB. H 1+i.15

NB. 1 2 3 5 5 7 7 10 11 11 11 16 13 15 17