

Code Xlogo : Diviseurs et multiples

pour pgcd :t1 :t2	2		
pour ppcm :t1 :t2	2		
pour etrangers? :a :b	2		
pour multiple? :n :q	2		
pour bezout :a :b	2		
pour synchro :l1 :l2	3		
pour syncx :cmp :l1 :l2 :arret1 :arret2	3		
pour cycle :l	3		
pour combi :x :a :b	3		
pour pleindecycles :l	3		
pour pdc :fois :l	3		
pour tablo :fd :fg :t	3		
pour tabld :d :fd :fg :c	3		
pour tablg :d :fg :c	4		
pour ft6 :x	4		
pour str0 :l	4		
pour facteurs :x	4		
pour xdecomposition :x	4		
pour factorisation :laliste	4		
pour diviseurs :n	5		
pour premier? :n	5		
pour premierdiviseur :n	5		
pour suitepremiers :n	6		
pour enrichie :o :maliste :mespremiers	6		
pour puis2 :x	6		
pour multiple :qui :dequoi	6		
pour lpremiers :d :a	6		
pour ulam :max	7		
pour visu :n :max :x	7		
pour pointe :x	7		
.....			
		pour ppte :x	7
pour bezout :a :b	2	pour ppcm :t1 :t2	2
pour combi :x :a :b	3	pour premier? :n	5
pour cycle :l	3	pour puis2 :x	6
pour diviseurs :n	5	pour str0 :l	4
pour etrangers? :a :b	2	pour suitepremiers :n	6
pour facteurs :x	4	pour synchro :l1 :l2	3
pour factorisation :laliste	4	pour syncx :cmp :l1 :l2 :arret1 :arret2	3
pour ft6 :x	4	pour tabld :d :fd :fg :c	3
pour lpremiers :d :a	6	pour tablg :d :fg :c	4
pour multiple :qui :dequoi	6	pour tablo :fd :fg :t	3
pour multiple? :n :q	2	pour ulam :max	7
pour pdc :fois :l	3	pour visu :n :max :x	7
pour pgcd :t1 :t2	2	pour xdecomposition :x	4
pour pleindecycles :l	3		

pour pgcd :t1 :t2

```
#t1 et t2 sont assumés positifs
soit "repons 0 soit "aux 0 soit "lereste 0
tantque [:repons = 0] [
si :t1 < :t2 [donne "aux :t1 donne "t1 :t2 donne "t2 :aux] #un petit swap
donne "lereste reste :t1 :t2
si egal? 0 :lereste [donne "repons :t2] [donne "aux :t2 donne "t2 :lereste donne "t1 :aux]
] # fin de la boucle
ret :repons
fin
```

pour ppcm :t1 :t2

```
si :t1 < :t2 [ret ppcm :t2 :t1]
# t1 est donc le plus grand
soit "p pgcd :t1 :t2
soit "q quotient :t1 :p
ret produit :q :t2
fin
```

pour etrangers? :a :b

```
ret egal? 1 pgcd :a :b
fin
```

pour multiple? :n :q

```
ret egal? 0 reste :n :q
fin
```

pour bezout :a :b

```
soit "a1 :b soit "a0 :a soit "x1 0 soit "x0 1 soit "y1 1 soit "y0 0
soit "q 0
tantque [:a1 > 0] [ #je dérécursivise ma procédure initiale BZ2
donne "q quotient :a0 :a1
soit "newa1 :a0 - :q * :a1 soit "newa0 :a1
soit "newx1 :x0 - :q * :x1 soit "newx0 :x1
soit "newy1 :y0 - :q * :y1 soit "newy0 :y1
donne "a1 :newa1 donne "a0 :newa0
donne "x1 :newx1 donne "x0 :newx0
donne "y1 :newy1 donne "y0 :newy0
]
soit "resul (liste :x0 :y0 :a0)
si (prem :resul) > 0 [ret (liste prem :resul moins prem sp :resul der :resul)]
soit "p ppcm :a :b soit "ax quotient :p :a soit "bx quotient :p :b
ret (liste somme prem :resul :ax (:bx - prem sp :resul) der :resul)
fin
```

pour synchro :l1 :l2

```
ec (ph 0 [->] prem :l1 prem :l2)
syncx 1 cycle :l1 cycle :l2 prem :l1 prem :l2
fin
```

pour syncx :cmp :l1 :l2 :arret1 :arret2

```
ec (ph :cmp [->] prem :l1 prem :l2)
si et egal? prem :l1 :arret1 egal? prem :l2 :arret2 [ec [Terminé ...] stop]
syncx (1 + :cmp) cycle :l1 cycle :l2 :arret1 :arret2
fin
```

pour cycle :l

```
ret md prem :l sp :l
fin
```

pour combi :x :a :b

```
si :x < 0 [ec ph [combi n'aime pas] :x stoptout]
si :a < 1 [ec ph [combi n'aime pas] :a stoptout]
si :b < 0 [ec ph [combi n'aime pas] :b stoptout]
soit "d pgcd :a :b
si non multiple? :x :d [ec [Pas de solution !] ret []]
soit "encours vrai soit "u quotient :x :a soit "v reste :x :a soit "resul []
tantque [:encours] [ # je dérécursivise là encore
si :u < 0 [donne "encours faux ec [Pas de solution !] ] [si multiple? :v :b [donne "resul liste :u
quotient :v :b donne "encours faux] [donne "u difference :u 1 donne "v somme :v :a]]
]
ret :resul
fin
```

pour pleindecycles :l

```
pdc compte :l :l
fin
```

pour pdc :fois :l

```
si egal? 0 :fois [stop] [ec :l]
pdc (:fois - 1) cycle :l
fin
```

pour tablo :fd :fg :t

```
DONNE "Taille :t
TABLD 0 :FD :FG :Taille
fin
```

pour tabld :d :fd :fg :c

```
SI EGAL? :C 0 [STOP]
TABLG :D :FG :Taille
TABLD (:D + :FD) :FD :FG :C - 1
fin
```

pour tablg :d :fg :c

SI EGAL? :C 0 [EC " STOP]

TAPE FT6 :D

TABLG (:D + :FG) :FG :C - 1

fin

pour ft6 :x

SI :X > 99999 [RET MOT CAR 32 "99999] [RET mot STR0 (6 - COMPTE :X) :X]

fin

pour str0 :l

SI :L < 1 [RET car 32] [RET mot car 32 STR0 (:L - 1)]

fin

pour facteurs :x

si ou vide? :x egal? 0 :x [ret [0]]

soit "decompo []

soit "Q 2

tantque [egal? 0 reste :x :Q] [# premiere boucle pour Q = 2

donne "decompo md :Q :decompo donne "x quotient :x :Q] # fin première boucle

soit "Q 3

tantque [egal? 0 reste :x :Q] [# seconde boucle pour Q = 3

donne "decompo md :Q :decompo donne "x quotient :x :Q] # fin seconde boucle

donne "Q 5 donne "Q2 7

tantque [:x > 3] [# tierce boucle, qui va en contenir deux autres

tantque [egal? 0 reste :x :Q] [# premiere boucle interne

donne "decompo md :Q :decompo donne "x quotient :x :Q] # fin première boucle interne

tantque [egal? 0 reste :x :Q2] [# seconde boucle interne

donne "decompo md :Q2 :decompo donne "x quotient :x :Q2] # fin seconde boucle interne

donne "Q somme :Q 6 donne "Q2 somme :Q 2

] # fin de la tierce boucle

ret :decompo

fin

pour xdecomposition :x

ret facteurs :x

fin

pour factorisation :laliste

si vide? :laliste [RET 0]

soit "repons 1

tantque [non vide? :laliste] [

donne "repons produit :repons prem :laliste

donne "laliste sp :laliste

]

ret :repons

fin

pour diviseurs :n

```
si ou vide? :n egal? 0 :n [ret [0]]
soit "listediv [1] soit "listecompl md :n []
soit "arret somme 1 tronque racine :n
soit "Q 2
tantque [:Q < :arret] [ # boucle externe
si egal? 0 reste :n :Q [ donne "listediv md :Q :listediv donne "listecompl mp (quotient :n :Q)
:listecompl]
donne "Q somme :Q 1
] # fin boucle externe
si egal? der :listediv prem :listecompl [donne "listecompl sp :listecompl]
ret ph :listediv :listecompl
fin
```

pour premier? :n

```
SI :N < 4 [RET "VRAI]
SI OU MULTIPLE :N 2 MULTIPLE :N 3 [RET "FAUX]
soit "arret 1 + tronque racine :n
si egal? 0 reste :n 2 [ret Vrai]
si egal? 0 reste :n 3 [ret Vrai]
soit "t 6 soit "repons Vrai # sera valide si :n est premier, sinon tué ci-dessous
tantque [:arret >= :t] [ # debut corps tantque
soit "t1 difference :t 1 si egal? 0 reste :n :t1 [donne "repons Faux donne "t :arret + 1] [soit "t1
somme :t 1
si egal? 0 reste :n :t1 [donne "repons Faux donne "t :arret + 1]]
donne "t somme :t 6
] #fin du tantque
ret :repons
fin
```

pour premierdiviseur :n

```
si egal? 0 :n [ret 1]
si egal? 1 :n [ret 1]
soit "arret 1 + tronque racine :n
si egal? 0 reste :n 2 [ret 2]
si egal? 0 reste :n 3 [ret 3]
soit "t 6 soit "repons :n # sera valide si :n est premier, sinon tué ci-dessous
tantque [:arret >= :t] [ # debut corps tantque
soit "t1 difference :t 1 si egal? 0 reste :n :t1 [donne "repons :t1 donne "t :arret + 1] [soit "t1
somme :t 1
si egal? 0 reste :n :t1 [donne "repons :t1 donne "t :arret + 1]]
donne "t somme :t 6
] #fin du tantque
ret :repons
fin
```

pour suitepremiers :n

```

SI :n=2 [ret [2]]
soit "Premiers [2 3]
soit "Q 5
tantque [:n >= :Q] [ # début du tantque
donne "Premiers enrichie :Q :Premiers :Premiers
soit "Q2 somme :Q 2
si :n >= :Q2 [donne "Premiers enrichie :Q2 :Premiers :Premiers]
donne "Q somme :Q 6
] # fin du tantque
ret :Premiers
fin

```

pour enrichie :o :maliste :mespremiers

```

tantque [non vide? :maliste] [ # debut boucle tantque
soit "rg PREM :maliste
si egal? 0 reste :o :rg [donne "maliste []] [
si ou (:o < puis2 :rg) (egal? :rg der :maliste) [donne "mesPremiers md :o :mesPremiers donne
"maliste [] ]]
si non vide? :maliste [donne "maliste sp :maliste]
] # fin tantque
ret :mesPremiers
fin

```

pour puis2 :x

```

RET :X * :X
fin

```

pour multiple :qui :dequoi

```

ret egal? 0 reste :qui :dequoi
fin

```

pour lpremiers :d :a

```

si :d > :a [ret []]
soit "q quotient :d 6
soit "u 6 * :q - 1 soit "v :u + 2
si :u < :d [donne "u :v donne "v :v + 4]
si :u < :d [donne "u :v donne "v :v + 2]
soit "rpart []
tantque [:u <= :a] [ # debut boucle tantque
si et :u > :d premier? :u [donne "rpart md :u :rpart]
si :v > :a [donne "u :v] [si et :v > :d premier? :v [donne "rpart md :v :rpart]]
donne "u somme :u 6 donne "v somme :v 6
] #fin boucle tantque
ret :rpart
fin

```

pour ulam :max

donne "X 2 VE CT LC

soit "tailleC TC FTC 2

EC [APPUYEZ SUR UNE TOUCHE POUR INTERROMPRE.]

VISU 1 :MAX 2

ftc :tailleC

EC [TERMINE !]

fin

pour visu :n :max :x

SI TOUCHE? [EC (PH [INTERROMPU A] :N [SUR] :MAX [...]) STOP]

SI :N > :MAX [STOP]

REPETE 2[REPETE :N[DONNE "X :X+1 AV 5 POINTE :X] TG 90]

VISU :N+1 :MAX :X

fin

pour pointe :x

SI PREMIER? :X [BC POINT POS LC]

fin