

ALGORITHME D'EUCLIDE ET PGCD

```
# Algorithmique /  
Programmation  
# PGCD  
# Python
```



Énoncé :

L'algorithme d'Euclide permet de déterminer le Plus Grand Commun Diviseur (PGCD) de deux entiers positifs a et b non nuls.

On effectue la division euclidienne de a par b , si le reste r vaut 0 le PGCD est la valeur de b .

Sinon on effectue ensuite la division euclidienne de b et r ...

1. A l'aide de l'algorithme d'Euclide, déterminer le PGCD de 288 et 135.
2. Programmer l'algorithme en langage Python.
3. Modifier le programme en incluant la programmation de la division euclidienne.

1. Algorithme d'Euclide sur un exemple

$$a = 288 \text{ et } b = 135$$

On utilise la commande `{Rmdr}`

OPTN // {CALC}: [F4] // [F6] [F6] // [F2]

On effectue la division euclidienne de 288 par 135

$$288 = 135 \times 2 + 18$$

Le reste r vaut donc 18

On effectue la division euclidienne de 135 par 18

$$135 = 18 \times 7 + 9$$

Le reste vaut donc 9

On effectue la division euclidienne de 18 par 9

$$18 = 9 \times 2 + 0$$

Le reste vaut donc 0 et ainsi le PGCD de 288 par 135 vaut 9 (dernière valeur de b).

On peut vérifier avec la commande **{GCD}**

[OPTN] [F6] // {NUMERIC}: [F4] // [F6] [F2]

[Menu] [Math] [Rad] [Norm1] [d/c] [Real]
 288 Rmdr 135
 135 Rmdr 18
 18 Rmdr 9
 []
 Int÷ Rmdr Simp [Right Arrow]

[Mode] [Math] [Rad] [Norm1] [d/c] [Real]
 135 Rmdr 18
 18 Rmdr 9
 GCD(288, 135)
 [Solve] [d/dx] [d²/dx²] [∫dx] [SolveN] [▶]

2. Programme en Python

Le reste de la division euclidienne de a par b est donnée par la syntaxe " $a\%b$ "

Si r est nul le PGCD est b .

Si r est non nul on recommence le processus en prenant b comme nouvelle valeur de a et r comme nouvelle valeur de b .

Pour cela on appelle le programme dans lui-même.

```
euclide.py 006/007 ▶
def euclide(a,b):
    r=a%b
    if r==0:
        return b
    else:
        return euclide(b,
```

FILE RUN SYMBOL CHAR A↔a ▶

```
MicroPython v1.9.3
| CASIO COMPUTER CO.,
>>>from euclide impor
>>>euclide(288,135)
9
>>>|
```

RUN A↔a CHAR

3. Avec la programmation de la division euclidienne

Pour programmer la division euclidienne, on commence par initialiser la valeur du quotient q à 0.

Ensuite on utilise une boucle tant que :

Tant que $a - bq$ est supérieur ou égal à 0 on incrémente la valeur de q de 1. La valeur de q recherchée est la dernière valeur pour laquelle $a - bq$ est positif. Après la boucle Tant que il est donc nécessaire de retrancher 1 à q .

```
euclide.py 001/010 ▶
def euclide(a,b):
    q=0
    while a-b*q>=0:
        q=q+1
    q=q-1
    if a-b*q==0:
        return b
```

FILE RUN SYMBOL CHAR A↔a ▶

Retrouvez toutes nos ressources pédagogiques sur www.casio-education.fr