

1 Tester si un entier est diviseur d'un autre

Soient a et b des entiers . a est-il diviseur de b ?

1.1 L'idée mathématique

Pour savoir si a divise b , il faut savoir si b est un multiple de a , autrement dit si b divisé par a donne un entier

1.2 La mise en algorithme

Variables

a : entier

b : entier

Début de l'algorithme

Saisir a

Saisir b

Si $b/a = \text{entier}$ **Alors**

| Afficher " a divise b "

Sinon

| Afficher " a ne divise pas b "

Finsi

```
1 from math import*
2 def div(a, b) :
3     if b%a ==0 :
4         print(a) ; print(" divise ") ; print(b)
5     else :
6         print(a) ; print(" ne divise pas ") ; print(b)
```

2 Recherche des diviseurs d'un nombre entier

Soit a un entier . On veut lister tous les diviseurs de a .

2.1 L'idée mathématique

On teste comme dans l'algorithme précédent pour savoir si un entier en divise un autre .

Et pour avoir tous les diviseurs de a , on va regarder tous les entiers inférieurs à a , donc une boucle pour .

2.2 La mise en algorithme

Variables

a : entier

i : entier

Début de l'algorithme

Saisir a

Pour i allant de 1 à a **Faire**

Si $a/i = \text{entier}$ **Alors**

 | Afficher i

Finsi

FinPour

```
1 from math import*
2 def diviseurs (a) :
3     for i in range(1,a+1):
4         if a%i ==0 :
5             print(i)
```

3 Tester si un nombre entier est premier

Soit n un entier . Est-il premier ?

3.1 L'idée mathématique

Un nombre est premier s'il a seulement deux diviseurs , 1 et lui même . On va donc compter les diviseurs de n .

On teste tous les entiers inférieurs à n et on crée un compteur qui donne le nombre de diviseurs de n .

Si on trouve deux diviseurs , n est premier ; sinon , n n'est pas premier .

3.2 La mise en algorithme

Variables

n : entier

i : entier

c : entier

Début de l'algorithme

Saisir n

$c \leftarrow 0$

Pour i allant de 1 à n **Faire**

Si $n/i = \text{entier}$ **Alors**

$c \leftarrow c + 1$

Finsi

FinPour

Si $c=2$ **Alors**

 afficher "n premier"

Sinon

 afficher " n est non premier"

Finsi

```
1 from math import*
2 def prem(n) :
3     c=0
4     for i in range(1,n+1):
5         if n%i ==0 :
6             c=c+1
7         if c==2 :
8             print(" n est premier")
9     else :
10        print ("n est non premier")
```

4 Déterminer le plus grand multiple d'un entier

Soient a et b des entiers . Déterminer le plus grand multiple de a inférieur à b

4.1 L'idée mathématique

On regarde comment trouver des multiples ; pour passer d'un multiple de a au suivant , on ajoute a . On doit calculer tous les multiples tant qu'ils sont plus petits que b .

4.2 La mise en algorithme

Variables

a , b , c : entiers

i : entier

Début de l'algorithme

Saisir a , b

$c \leftarrow 0$

Tant que $c \leq b$ **Faire**

$c = c + a$

FinTantque

Sorties :

Afficher $c - a$

```
1 from math import*
2 def multiple(a,b) :
3     c=0
4     while c <=b:
5         c=c+a
6     return(c-a)
```

**Attention**

Si on affiche à la fin c et non $c - a$, on affiche en fait le premier multiple plus grand que b car la boucle tant que s'arrête quand on a obtenu ce multiple .

Exemple

On va faire tourner cet algorithme avec $a = 3$ et $b = 14$

Passage dans la boucle	c
Avant	
1er passage	
2ème passage	
3ème passage	
4ème passage	
5ème passage	
sortie	

5 Encadrement d'un réel par balayage

Soit x un réel et n un entier . Donner un encadrement à 10^{-n} de x .

5.1 L'idée mathématique

On commence par trouver un entier plus petit que le réel puis on affine en ajoutant les décimales

5.2 La mise en algorithme

Variables

a, x, k : réels

i : entier

Début de l'algorithme

Saisir x, n

$a \leftarrow$ partie entière de x

Pour i allant de 1 à n **Faire**

$k \leftarrow 10^{-i}$

Tant que $a < x$ **Faire**

$a \leftarrow a + k$

FinTantque

$a \leftarrow a - k$

FinPour

Sorties :

Afficher $a, a + k$

```

1 from math import*
2 def balayage(x,n) :
3     a=int(x)
4     for i in range(1,n+1):
5         k=10**(-i)
6         while a < x :
7             a=a+k
8         a=a-k
9     return(round(a,n) , round( a+k, n))
    
```



Astuce

Python a quelques problèmes d’affichage avec les arrondis décimaux , l’instruction round

- oblige à arrondir le nombre voulu à n décimales .

Exemple

Faites tourner cet algorithme avec $x = 2,21345$ et $n = 2$

	i	k		a
avant boucle pour				
1er passage boucle pour			1er passage boucle tant que	
			2ème passage	
			3ème passage	
			sortie	
2ème passage boucle pour			1er passage boucle tant que	
			2ème passage	
			sortie	
sortie finale				

6 Déterminer un seuil

Soit x un réel et n un entier . Soit A un réel . Déterminer le plus petit n tel que $x^n > A$.

6.1 L'idée mathématique

On calcule les puissances de x tant qu'elles sont plus petites que A

6.2 La mise en algorithme

Variables

a , x , k : réels

i : entier

Début de l'algorithme

Saisir x , A

$n \leftarrow 0$

Tant que $x^n \leq A$ **Faire**

$n \leftarrow n + 1$

FinTantque

Sorties :

Afficher n

```
1 from math import*
2 def seuil (x,A):
3     n=0
4     while x**n <= A :
5         n=n+1
6     return(n)
```