

Déterminer si un entier naturel est premier

Principe

Un test de primalité permet de vérifier si un nombre est un nombre premier ou non.

Exercice n°1 Le crible d'Erathostène

Rappel du principe

On part de l'entier naturel 2 et on regarde tous ses multiples jusqu'à un entier n donnée.

On repart du premier entier non rencontré, 3. On regarde tous ses multiples puis on revient au premier entier non rencontré, 5.

On crée ainsi une liste d'entier [2; 3; 5; ...] qui est la liste des premiers nombres premiers.

1. Sous la forme d'un tableau, déterminer les premiers nombres premiers inférieurs à 100.
2. Vérifier le résultat en utilisant le programme Python ci-dessous :

```

1 def eratosthene(n):
2     L = [ i for i in range(2,n+1) ]
3     P = [ ]
4     while len(L) != 0:
5         P.append(L[0])
6         i = L[0]
7         for k in L:
8             if k % i == 0:
9                 del(L[L.index(k)])
10    return P

```

3. Donner un avantage et un inconvénient de ce test de primalité.

Exercice n°2 Critère d'arrêt

Propriété

Soit n un nombre entier supérieur ou égal à 2.

n est premier si et seulement si n n'a pas de diviseur premier inférieur ou égal à n .

1. Utiliser ce test pour vérifier si 109 est un nombre premier.
2. A l'aide du programme Python ci-dessous, vérifier si 527 et 719 sont des nombres premiers ou non.

```

1 def primalite(n):
2     i=2
3     if n%i==0:

```

```
4     return n,"est divisible par ", i
5     i=i+1
6     while i**2<=n:
7         if n%i==0:
8             return n, "est divisible par ", i
9         i=i+2
10    return n, "est un nombre premier."
```

3. Donner un avantage et un inconvénient de ce test de primalité.

[> Correction des exercices](#)[Exercice n°1](#)

1. Voici le crible d'Eratosthène pour les entiers inférieurs à 100 :

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

2. `>> eratosthene(100)`
`>> [2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97]`
3. Cela permet d'afficher toute la liste des premiers nombres premiers mais la procédure est longue et ne permet pas de vérifier si des grands nombres sont premiers ou non.

[Exercice n°2](#)

1. $\sqrt{109}$ est compris entre 10 et 11.
On va donc tester la divisibilité avec 2 ; 3 ; 5 et 7.
Or 109 n'est divisible par aucun de ces nombres premiers. Donc 109 est un nombre premier.
2. `>> primalite(527)`
`>> 527 est divisible par 17.`

`>> primalite(719)`
`>> 719 est un nombre premier.`
3. Ce test permet de vérifier directement si un nombre est premier mais, là encore, la procédure peut être longue pour des nombres entiers beaucoup plus grand.