

Tests de primalité

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 2.

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.

Exercice n°3 En utilisant le petit théorème de Fermat

On considère un entier n supérieur ou égal à 3 et le programme Python ci-dessous :

```
1 def testFermat(n):
2     F=2**(n-1)%n
3     return F
```

1. Que permet d'effectuer la ligne 2 de ce programme ?
2. D'après le petit théorème de Fermat, que renvoie l'algorithme quand n est un nombre premier ?
3. Que peut-on dire lorsque la valeur de F affichée en sortie de programme n'est pas égale à 1 ?
4. Est-ce que 561 est un nombre premier ?
5. Modifier le précédent programme pour élaborer un test permettant de savoir si le nombre n saisi est premier ou non.
6. Tester le programme pour $n = 154\ 515\ 677$.
7. 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.

Exercice n°3

- Cette deuxième ligne renvoie le reste de la division euclidienne de 2^{n-1} par n .
- Si n est un nombre premier, le programme va renvoyer la valeur 1.
- Si le programme ne renvoie pas 1, le nombre n n'est pas premier.

4. 561 est divisible par 3. Pourtant, il passe au travers de notre test de primalité.

5. Voici le programme modifié :

```
1 def testFermat(n):
2     F=2**(n-1)%n
3     if F==1:
4         return n, "est peut-être un nombre premier"
5     else :
6         return n, "n'est pas un nombre premier"
7     return F
```

6. `>> testFermat(154515677)`

`>> 154 515 677 est peut-être un nombre premier.`

7. C'est un test qui est beaucoup plus rapide que les précédents. Le problème, c'est qu'il existe certains nombres qui ne sont pas premiers et qui passent au travers de ce test. On appelle ces nombres les nombres de Carmichael.