

Nombres premiers

Exercice

Écrire une fonction `premiers(n)` qui détermine la liste des nombres premiers inférieurs ou égaux à un entier n donné.

Selon la méthode du crible d'Ératosthène, on peut supprimer d'une liste contenant les entiers entre 2 et n tous les multiples de 2, puis supprimer tous les multiples du nombre suivant 2 (sauf erreur, cela doit être 3), etc.

La méthode du crible d'Ératosthène est relativement simple à mettre en oeuvre : on parcourt la liste des nombres entiers inférieurs ou égaux à n autant de fois qu'il est possible en supprimant les multiples du premier nombre non supprimé.

La liste de départ est obtenue avec `list(range(2,n+1))` qui contient [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12] lorsque $n = 12$. La suppression des multiples de $k = 2$ (k indique le premier nombre non supprimé) s'obtient en redéfinissant la liste `prem`, par l'instruction `prem=[p for p in prem if p<=k or p%k!=0]` qui opère le tri `p<=k or p%k!=0` signifiant : soit p est premier (inférieur au dernier nombre non supprimé), soit p n'est pas divisible par k .

On recommence l'opération après avoir choisi la nouvelle valeur de k avec `k=prem[prem.index(k)+1]`.

On arrive ainsi à [2, 3, 5, 7, 11] après l'avoir parcourue pour $k = 2$ et $k = 3$.

Le prochain nombre non supprimé est $k = 5$, mais on n'a pas besoin de supprimer ses multiples car le premier qui n'a pas déjà été supprimé est 5×5 , un nombre supérieur à $n = 12$. Pour cette raison, on arrête la procédure quand k atteint ou dépasse \sqrt{n} .

```
1 from math import *
2 def premiers(n):
3     prem=list(range(2,n+1))
4     k=2
5     nRacine=sqrt(n)
6     while k<nRacine:
7         prem=[p for p in prem if p<=k or p%k!=0]
8         k=prem[prem.index(k)+1] # nouveau nombre premier
9     return prem
10
11 Liste_premiers=premiers(100)
12 print("Plus grand premier =",Liste_premiers[-1])
13 print("Nombre de premiers =",len(Liste_premiers))
14 print("Liste premiers :",Liste_premiers)
```

Il y a sans doute de nombreuses autres façons de générer cette liste. Nous en resterons à celle-là qui est

assez concise et efficace. Sur la calculatrice, la plus grande partie de la liste est cachée à l'affichage. Il est possible de la faire défiler avec les flèches directionnelles. On peut aussi mieux penser l'affichage en mettant à profit les 30 caractères affichés par ligne : un nombre par ligne devrait permettre d'avoir un affichage correct (mais est-ce utile?) jusqu'au plus grand nombre premier à 30 chiffres...

```
deg PYTHON   
>>> from premier import *  
Plus grand premier = 97  
Nombre de premiers = 25  
Liste premiers : [2, 3, 5, 7,  
9, 83, 89, 97]  
>>> |
```