Nombres premiers

Nombre premier ?

On souhaite réaliser une fonction `premier(n)` pour $n > 1$ qui renvoie `True` si un nombre est premier, `False` sinon.

1. Quelle condition faut-il respecter pour qu'un nombre soit considéré comme premier ?

   Un nombre est premier s'il n'a pas d'autre diviseur que 1 et lui-même.

2. Nous allons passer en revue des entiers pour tester si ce sont des diviseurs de n. Quelles sont les valeurs que nous allons tester ? Avec quelle ligne de code ?

   Nous allons tester tous les entiers entre 2 et $\sqrt{n}$ inclus, avec une boucle `for i in range(2,sqrt(n)+1):`.

3. Quelle sont les conditions pour le programme s'arrête ? Quelle instruction permet d'arrêter une fonction en affichant un résultat ?

   Le programme peut immédiatement s'arrêter s'il rencontre un diviseur. Sinon, le programme s'arrête après avoir testé tous les diviseurs potentiels. On utilisera l'instruction `return`.

4. A l'aide des questions précédentes, écrire la fonction `premier(n)`.

```
1 from math import *
2 def premier(n):
3   if n>1:
4     for k in range(2,sqrt(n)+1):
5       if k<n and n%k==0:
6         return False
7     return True
```

   A noter que la commande `premier(2)` risque de poser problème sans l'ajout du test `k<n` en ligne 5.

Lister les nombres premiers

On aimerait utiliser une fonction `liste_premiers(k)` qui liste les k premiers nombres premiers.
Pour cela, on propose d'écrire dans le même script pour pouvoir réutiliser la fonction `premier(n)` définie plus haut.

1. On propose d'utiliser une variable `liste` qui va stocker les nombres premiers. Comment initialiser cette variable au début du programme ?

   Au début du programme, la variable `liste` est vide, on l'initialiste avec `liste=[]`.

2. Quel est le premier entier à tester dans l'objectif de l'ajouter dans la liste ?

   On pourra commencer avec le premier nombre premier : 2.

3. Combien d'éléments doit compter la variable `liste` à la fin du programme ? Peut-on connaître le nombre d'entiers à tester ? En déduire le type de boucle à utiliser. On propose d'utiliser l'instruction `len(liste)` qui permet de retourner le nombre d'éléments dans une liste.

   La liste doit compter n éléments mais il n'est pas possible de connaître le nombre d'entiers à tester pour atteindre cette valeur. En revanche, on connaît la condition de sortie de la boucle, à savoir lorsque le nombre d'éléments dans la liste est atteint. On utilisera donc l'instruction `while len(liste)<n`.

4. Quelle condition faut-il vérifier pour qu'un nombre n se trouve dans cette liste ?

   Le nombre sera stocké dans la liste s'il s'agit d'un nombre premier, autrement dit si `premier(i)==True` avec i qui varie de 2 à potentiellement l'infini.

5. A l'aide des questions précédentes, écrire le programme souhaité.

```
def liste_premiers(n):
  liste=[]
  i=2
  while len(liste)<n:
    if nombre_premier(i)==True:
       liste.append(i)
    i+=1
  return liste
```

Pour les plus avancés !

On a vu dans les activités précédentes qu'il était possible d'utiliser les listes en compréhension pour écrire de très courtes fonctions. Ecrire une fonction `premiers_short(n)` dont l'objectif ici, attention, est de lister les nombres premiers compris entre 1 et n et qui ne dépasse pas deux lignes !

```
def liste_premiers_short(n):
  return [k for k in range(1,n+1) if premier(k)==True]
```