

## Enonce

Un nombre premier est un nombre entier naturel qui admet exactement deux diviseurs distincts entiers et positifs : 1 et lui-même.

Le crible d'Ératosthène permet de déterminer les nombres premiers plus petit qu'un certain nombre `n` fixé.

On considère pour cela un tableau `tab` de `n` booléens, initialement tous égaux à `True`, sauf `tab[0]` et `tab[1]` qui valent `False`, 0 et 1 n'étant pas des nombres premiers.

On parcourt alors ce tableau de gauche à droite.

Pour chaque indice `i` :

- si `tab[i]` vaut `True` : le nombre `i` est premier et on donne la valeur `False` à toutes les cases du tableau dont l'indice est un multiple de `i`, à partir de `2*i` (c'est-à-dire `2*i`, `3*i` ...).
- si `tab[i]` vaut `False` : le nombre `i` n'est pas premier et on n'effectue aucun changement sur le tableau.

On dispose de la fonction `crible`, incomplète et donnée ci-dessous, prenant en paramètre un entier `n` strictement positif et renvoyant un tableau contenant tous les nombres premiers plus petits que `n`.



Script Python

```

1 def crible(n):
2     """
3         Renvoie un tableau contenant tous les nombres premiers plus petits que N
4     """
5     premiers = []
6     tab = [True] * n
7     tab[0], tab[1] = False, False
8     for i in range(..., n):
9         if tab[i] == ...:
10             premiers.append(...)
11             for multiple in range(2*i, n, ...):
12                 tab[multiple] = ...
13
14     return premiers
15
16 assert crible(40) == [2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37]

```

Compléter le code de cette fonction.

