# Sujet 4

EXERCICE 1 (10 points)
Ecrire une fonction ecri ture_bi nai re_enti er_posi ti f qui prend en parametre un entier positif n et renvoie une chaine de caractere correspondant a l'ecriture binaire de n.
On rappelle que:
l'ecriture binaire de 25 est 11001 car 25 = I x + 1 x 23 +0 x 22 +0 x 21 + 1 x 2°
n % 2 vaut 0 ou 1 selon que n est pair ou impair ;
n // 2 donne le quotient de la division euclidienne de n par 2.
Il est interdit dans cet exercice d'utiliser la fonction bin de Python. Exemples: >>> 5 % 2
1
>>> 5 // 2
2
>>> ecri ture_binai re_entier_positif (0)
'0'
>>> ecriture_binai re_entier_positif (2)
'10 '
>>> ecri ture_binai re_entier_positif (105)
'1101001'

EXERCICE 2 (10 points)
Un nombre premier est un nombre entier nature) qui admet exactement deux diviseurs distincts entiers et positifs : **1** et lui-meme.

Le crible d'Eratosthene permet de determiner les nombres premiers plus petit qu'un certain nombre n fixe strictement superieur a 1.

On considere pour cela un tableau tab de n booleens (type li st), initialement tous egaux a True, sauf tab [0] et tab [1] qui valent False, 0 et 1 n'etant pas des nombres premiers.

On parcourt alors ce tableau de gauche a droite et pour chaque indice **i :**

- si tab **[i]** vaut True : le nombre **i** est premier et on donne la valeur False a toutes les cases du tableau dont l'indice est un multiple de **i, a** partir de 2*i (c'est-a-dire 2*1,3*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 c ri ble, donnee ci-dessous et a completer, prenant en parametre un entier n strictement superieur a 1 et renvoyant un tableau contenant tous les nombres premiers plus petits que n.

```
def crible(n):
    """Renvoie un tableau contenant tous les nombres premiers
   plus petits que n."""
    premiers = []
    tab = [True] * n
    tab [0] , tab [1] = False, False for i in
    range(n):
        if tab[i] :
            premiers....
            multi ple =
            while multiple < n: tab
                [multiple] = multiple =
    return premiers
```

Exemples :

```
crible(40)
3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37] crible(5)
3]
```