

```

"""
Série 1 : Rappel
Date : 07-09-2023
Groupe : Ipein/ST3/GB
"""

# Exercice 1
## Q1
"""
def init(L):
    ''' initialiser une liste avc 100 valeurs égales à 1'''
    L= [1] * 100
    return L

# prog principal
L = []
L = init(L)

print("L=",L)
"""

def init(L):
    #print("id(L) local avant for :",id(L))
    for i in range(100):
        L.append(1)
    #print("id(L) local après for :",id(L))

"""
L = []
#print("id(L) gloable avant init :",id(L))
init(L)
#print("id(L) gloable après init :",id(L))
print(L)
"""

# Q2
def multiple(L,i): # signature (contrat)
    ''' mettre à 0 les multiples de i ( sauf i)'''
    for j in range(i+1, len(L)):
        if j % i == 0:
            L[j] = 0

# test
#
#multiple(L,2)
#print(L)

# Q3
def suivant(L,i):
    """
    for j in range(i+1, len(L)):
        if L[j]==1 :
            return j # arrêter la fonction (y compris la boucle)
    #return j # arrêter la fonction

    j = i+1
    while L[j]==0:
        j += 1
    return j
    """
    return L.index(1,i+1)

# Q4
def crible(L):
    p = []
    import math
    end = int(math.sqrt(len(L)))
    i = 2
    while i <= end:
        multiple(L,i)
        i = suivant(L,i)

    #extraction des nombres premiers
    for i in range(2,len(L)):
        if L[i] == 1 :
            p.append(i)
    return p

L = []
init(L)
p = crible(L)
print(p)

import math
def crible_rec(L,i):
    if i <= int(math.sqrt(len(L))):
        multiple(L,i)

```

```
    i=suivant(L,i)
    crible_rec(L,i) ##
else:
    p = []
    for i in range(2,len(L)):
        if L[i] == 1 :
            p.append(i)
    return p
```