

```

def listversnombre(L):
    res = 0
    for k in range(len(L)):
        res += L[k]*(10**(len(L)-k-1))
    return(res)

def kaprekar(n,b): #b est le nombre de chiffres du nombre de départ
    motn=str(n)
    L1=[int(x) for x in motn]
    long = len(motn)
    while long < b:
        L1.append(0)
        long +=1
    L1.sort()
    L2 = [x for x in L1[::-1]]
    n1 = listversnombre(L1)
    n2 = listversnombre(L2)
    return(n2-n1)

def bouclekap(n,b):
    L = []
    nb = kaprekar(n,b)
    while not(nb in L):
        L.append(nb)
        nb = kaprekar(nb,b)
    return(L)

for k in [242, 254, 484, 667, 148, 68, 59, 2548, 5789, 5888, 6214, 3215]:
    print(bouclekap(k,len(str(k))))

""" Pour 254, on trouve : 254, 297, 693, 594, 495
Pour 484 : 484, 396, 594, 495
Pour 667 : 667, 99, 0
Pour 148 : 148, 693, 594, 495
"""

L495 = ['L495']

for k in range(100,1000):
    if 495 in bouclekap(k,3):
        L495.append(k)

    else :
        print(k, bouclekap(k,3))

print(len(L495)-1) # on trouve 891
""" On trouve donc que pour tous les nombres de 3 chiffres différents
on arrive nécessairement sur 495"""

for k in range(1,100):
    print(k, bouclekap(k,2))
"""
L9 = ['L9']

for k in range(10,100):
    if 9 in bouclekap(k,2):
        L9.append(k)

    else :
        print(k, bouclekap(k,2))

print(len(L9)-1) # 81

```

Pour les nombres à 2 chiffres, on arrive presque toujours au cycle

63, 27, 45, 9, 81

les seules exceptions étant pour les multiples de 11

"""

```
L6174 = ['L6174']
for k in range(1000,10000):
    if 0 in bouclekap(k,4):
        L0.append(k)
    elif 6174 in bouclekap(k,4):
        L6174.append(k)

    else :
        print(k, bouclekap(k,4))
""" pour les nombres à 4 chiffres, à part les cas dégénérés, on arrive à 6174"""
print(len(L0)-1) # 9
print(len(L6174)-1) # 8991
```

#Exercice 2

```
def produit(n):
    motn = str(n)
    res = 1
    for k in motn:
        res *= int(k)
    return(res)
```

#Les entiers qui peuvent s'écrire comme produit des chiffres d'un entier sont
#les entiers qui n'ont comme diviseurs premiers que 2, 3, 5 et 7

```
def test(n):
    m = n
    for k in [2,3,5,7]:
        while m%k == 0 and m !=0:
            m = m//k
    if m>9:
        return False
    else:
        Lc=[]
        reste = n
        for k in range(9,1,-1):
            while reste%k ==0 and reste>1:
                Lc.append(k)
                reste = reste//k
        return(listversnombre(Lc[::-1]))
```

import random as rd

```
for k in range(100):
    m = 0
    for p in range(4):
        a = rd.randint(0,9)
        m=10*m+a
    print(m,test(m))
```