

```
# Rques_DS2_Info_MPSI_2023_2024.py
```

##Rques exo 1

```
'''
```

$a\%b$ renvoie le reste de la division euclidienne de a par b

$a//b$ renvoie le quotient de la DE de a par b

a/b renvoie un objet de type float même si b divise a

Pour savoir si b divise a :

- le mieux est d'utiliser `if a%b == 0:`

- utiliser `if int(a/b) == a/b:` fonctionne dans les cas rencontrés ici avec python 3... mais ne fonctionnait pas avec les versions antérieures.

D'un point de vue théorique ce n'est pas une bonne approche car a/b est de type float et $\text{int}(a/b)$ est de type int. De plus, si a et b sont des float, le résultat de ce test peut donner lieu à de surprises : `int((0.2+0.1)/0.3) == (0.2+0.1)/0.3` renvoie False.

Pour le calcul de la valuation p-adique de n , on effectue des divisions successives de n par p .

Si vous utilisez `n = n/p`, les nombres manipulés sont convertis en float et ils sont stockés de manière approchée, contrairement aux int qui sont stockés de manière exacte.

Une fonction `vp` codée avec `n = n/p` renverra :

- 1175 lors du calcul de `v2(10**30)` avec `n = n/p`

- 100 (bonne valeur) lors du calcul de `v2(10**30)` avec `n = n//p`

```
'''
```

```
def vp(n,p):
```

```
    puiss = 0
```

```
    while n > 0:
```

```
        n = n / p #mauvaise programmation
```

```
        puiss += 1
```

```

        print(n)
    return puiss

print(vp(10**30,2))

```

#Q1.

```

'''
- 1 n'est pas premier
- dans un range les arguments doivent être des entiers : range(2,sqrt(n)+1) ne fonctionne pas. Il faut par exemple écrire range(2, int(sqrt(n)) + 1)
- Nombreux sont ceux qui concluent trop vite que n est (ou n'est pas premier)'''

```

```

def estPremier(n): #mauvaise fonction
    for d in range(2,int(sqrt(n))+1):
        if n%d == 0:
            return True
        else:
            return False

```

```

def estPremier(n): #bonne fonction
    for d in range(2,int(sqrt(n))+1):
        if n%d == 0:
            return True
    return False

```

##Rques Pb

```

'''

```

Quand on a besoin d'avoir accès à l'indice d'un élément dans une liste L et que l'on parcourt L avec une boucle for le plus simple est de parcourir les indices de L et pas les éléments :

- for i in range(len(L)): i décrit les entiers de 0 à len(L)-1, c'est à dire directement les indices de L
- for x in L: x décrit les éléments de L. Pour avoir accès à l'indice il faut, soit rajouter un compteur (ce qui complique le code pour rien),

soit utiliser `L.index(x)` ce qui a une complexité en $O(\text{len}(L))$ et donc un coût élevé pour rien.
'''

#Q1. appartient(L,x,i0)

'''

- De nombreux élèves cherchent `x` dans `L` avant l'indice `i0` (mauvaise lecture de l'énoncé)
- Beaucoup concluent trop vite que `x` n'est pas présent
- Inutile de parcourir toute la liste : il suffit de commencer à l'indice `i0`. Et pour cela, inutile d'encombrer la mémoire en définissant la liste `LL = L[i0:]`

'''

#Q2.tousDiff(L)

'''

- il faut faire 2 boucles imbriquées
- attention à ne pas comparer `L[i]` avec `L[i]` (ou alors le prendre en compte pour ceux qui choisissent de compter le nombre de termes égaux)
- préférer une version où on itère sur les indices de `L` pour avoir un minimum d'itérations.

'''

#Q6.deuxMeilleurs(R)

'''Il s'agit de déterminer les indices des 2 premiers maximum.

- itérer sur les indices
- il est possible d'écrire un code qui ne parcourt `R` qu'une seule fois
- Utiliser `pop` n'est pas une bonne idée car l'indice dans la liste modifiée n'est pas le même que dans la liste originale. De plus, si on utilise `R.pop(i)`, il faut garder en tête que la liste `R` donnée en argument sera modifiée lors de l'appel de la fonction. Le `R` initial est donc perdu ce qui peut poser des problèmes pour une utilisation ultérieure.

'''

#Q10.Fonctions f et g

```
'''  
- Qd on demande le type d'un objet il faut donner  
le nom du type en langage Python (int, float, str,  
bool, list, ...)  
- Il faut décrire de manière précise ce que  
renvoie la fonction.  
f(m) renvoie une chaine de caractère composée de m  
caractères choisis aléatoirement entre les lettres  
a et z (minuscules uniquement)  
g(n) une liste de n chaines de caractères  
aléatoires contenant entre 4 et 10 caractères  
chacune (nbre de caractères aléatoires), toutes  
différentes  
'''
```