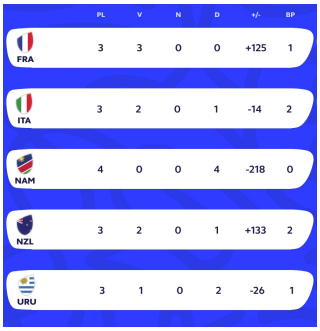
DS n° 01 – Initiation à l'algorithmie

- Faire tous les exercices dans un fichier NomPrenom.py à sauvegarder,
- mettre en commentaire l'exercice et la question traités (ex : # Exercice 1),
- ne pas oublier pas de commenter ce qui est fait dans votre code (ex : # Je crée une fonction pour calculer la racine d'un nombre),
- il est possible de demander un déblocage pour une question, mais celle-ci sera notée 0,
- il faut vérifier avant de partir que le code peut s'exécuter et qu'il affiche les résultats que vous attendez. Les lignes de code qui doivent s'exécuter sont décommentées.

1 Coupe du monde de rugby

Le classement des poules s'effectue en comptant le nombre de points de chaque équipe.
Une victoire (V) rapporte 4 points, un match nul (N) (égalité) 2 points et une défaite (D) 0 point et à ses points s'ajoutent des points de bonus (BP).
Ainsi la France possède pour le moment $3 \cdot 4 + 0 \cdot 2 + 1 = 13$ points.



Le script python ci-dessous permet d'établir le classement mais il doit être complété.
Il est disponible dans le dossier « `/home/eleve/Ressources/PTSI/script_rugby.py` », vous pouvez copier/coller son contenu dans votre script.

```
colonnes=['Nom pays','Victoires','Nuls','Défaites','Differences points',\
'Points bonus','Total points']
groupe_A=[['France',3,0,0,125,1,0],['Italie',2,0,1,-14,2,0],\
['Namibie',0,0,4,-218,0,0],['Nouvelle-Zélande',2,0,1,133,2,0],\
['Urugay',1,0,2,-26,1,0]]

def calcul_points(equipe):
    return equipe[1]+equipe[2]

for equipe in groupe_A:
    equipe[...]=calcul_points(equipe)

def sort_key(equipe):
    return equipe[6],equipe[4]

groupe_A.sort(key=sort_key, reverse=True)

for equipe in groupe_A:
    print(equipe)
```

Question 1 **Modifier** la fonction `calcul_points` aux lignes 7 et 8 afin qu'elle utilise les critères propres à la compétition afin d'établir les scores. Tester le résultat en affichant `print(calcul_points(['France',3,0,0,125,1,0]))`.

Question 2 **Modifier** la ligne 11 afin que le script enregistre les scores à la suite des données de chaque équipe. Ainsi, la liste liée à la France devient `['France', 3, 0, 0, 125, 1, 13]`.

2 La légende de Sissa

La légende de Sissa est un mythe qui raconte l'invention du Chaturanga, la forme indienne du jeu d'échecs.

Le brahmane ou sage Sissa (ou Sessa) Sissa Ibn Dahir est un personnage mythique de l'Inde.

En Inde, le roi Belkib (ou Bathait), qui s'ennuie à la cour, demande qu'on lui invente un jeu pour le distraire. Le sage Sissa invente alors le Chaturanga, l'ancêtre du jeu d'échecs, ce qui ravit le roi. Pour remercier Sissa, le roi lui demande de choisir sa récompense, aussi fastueuse qu'elle puisse être. Sissa choisit de demander au roi de prendre le plateau du jeu et, sur la première case, poser un grain de riz, ensuite deux sur la deuxième, puis quatre sur la troisième, et ainsi de suite, jusqu'à la 64ème, en doublant à chaque fois le nombre de grains de riz que l'on met.

Le roi et la cour sont amusés par la modestie de cette demande. Le roi tente de persuader Sissa d'accepter une récompense d'une valeur plus importante. Devant le refus de Sissa, il ordonne que les grains soient apportés. Mais lorsqu'on met en œuvre la demande, on s'aperçoit qu'il n'y a pas assez de grains de riz dans tout le royaume pour la satisfaire.

Extrait de la page Wikipédia « `https://fr.wikipedia.org/wiki/Légende_de_Sissa` ».

La résolution du problème de l'échiquier de Sissa est un exercice algorithmique consistant à déterminer le nombre de grains de riz que devrait obtenir Sissa.

Question 3 **Coder** la résolution de ce problème et **afficher** le nombre de grains de riz que demande Sissa.

Le poids d'un grain de riz est d'environ 28g.

Question 4 **Calculer**, à l'aide de python à la suite de votre script, la masse (en tonnes) des grains de riz dus à Sissa.

3 Propriété des multiples de 3

Un nombre entier est divisible par 3 quand la somme de ses chiffres est un multiple de 3 et uniquement dans ce cas.

ex :

— 7 152 est divisible par 3 car 7+1+5+2=15 et 15 est un multiple de 3,

— 835 n'est pas divisible par 3 car 8+3+5=16 n'est pas divisible par 3.

Dans le but de chercher un contre exemple, on va vérifier cette propriété pour les 1000 premiers entiers.

En sachant que :

— `nombre//10` renvoi le quotient de la division euclidienne de `nombre` par `10`,

— `nombre%10` renvoi le reste de la division euclidienne de `nombre` par `10`,

— `a==b` renvoie `True` si $a = b$ et 0 si ce n'est pas le cas,

— `a!=b` renvoie `True` si $a \neq b$ et 0 si ce n'est pas le cas.

L'algorithme suivant permet de coder la fonction `somme_chiffres` qui permet de sommer tous les chiffres d'un nombre :

1 Mise à 0 de `somme`,

2 Tant que le quotient de la division de `nombre` par 10 est positif :

3.1 `reste` est le reste de la division euclidienne de `nombre` par 10,

3.2 `nombre` est le quotient de la division euclidienne de `nombre` par 10,

3.3 ajouter `reste` à `somme`

4 ajouter `nombre` à `somme`

5 retourner `somme`.

Exemple : soit le `nombre` 1234.

nombre	reste	somme
1234	0	0
123	4	4
12	3	7
1	2	9
0	1	10

Le résultat est 10, et en effet, 1+2+3+4=10.

Question 5 **Coder** la fonction `somme_chiffres(nombre)` qui renvoie la somme des chiffres de nombre. La tester sur le nombre 1234.

Le code suivant utilise la fonction déterminée à la question précédente, il renvoie 'Faux' si l'hypothèse précédente n'est pas validée (en théorie, cela ne doit jamais arriver.)

```
1 for i in range(1,10**4):
2     if somme_chiffres(i).......: # si la somme des chiffres est un multiple de 3
3         if ......:  # si le nombre n'est pas un multiple de 3
4             print('Faux')
5     else:
6         if ......: si le nombre est un multiple de 3
7             print('Faux')
```

Question 6 **Compléter** le code précédent afin de vérifier l'hypothèse pour les entiers de 1 à 9999.

Afin de tester si le résultat de la somme des chiffres est divisible par 3, on souhaite remplacer le test de la ligne 2 par un test qui consisterait à déterminer la somme des chiffres de ce résultats jusqu'à ce que celui-ci soit ramené à un seul chiffre : 3, 6 ou 9.

Exemple : `somme_chiffres(9384)=24` or `somme_chiffres(24)=6` donc divisible par 3.

Question 7 **Proposer** une nouvelle version du code précédent incluant cette nouvelle solution.

FIN

Correction

Question 1

```
def calcul_points(equipe):
    return equipe[1]*4+equipe[2]*2+equipe[5]
print(calcul_points(['France',3,0,0,125,1,0]))
```

Question 2

```
for equipe in groupe_A:
    equipe[6]=calcul_points(equipe)
```

Question 3

```
somme=0
for i in range(64):
    somme+=2**i

print(somme)
```

Question 4

```
poids=0.028

total=somme*poids
print(total/1000)
```

Question 5

```
def somme_chiffres(nombre):
    somme=0
    while nombre//10>0:
        reste=nombre%10
        nombre=nombre//10
        somme+=reste
    somme+=nombre
    return somme
```

Question 6

```
for i in range(10**4):
    if somme_chiffres(i)%3==0:
        if i%3!=0:
            print('Faux')
    else:
        if i%3==0:
            print('Faux')
```