

Informatique

TP n°1 - Manipulation des listes

PSI - Lycée Rabelais

CORRIGÉ

1 Petits exercices

1.1 Centrale 2015 (premières questions)

Question 1.

- $[6, 5, 4] + [3, 2, 1]$ renvoie $[6, 5, 4, 3, 2, 1]$
- $3 * [1, 2, 3]$ renvoie $[1, 2, 3, 1, 2, 3, 1, 2, 3]$

Question 2.

```
1 def smul(n,L):
2     res=[]
3     for li in L:
4         res.append(n*li)
5     return res
```

Question 3.

```
1 def ssom(L1,L2):
2     res=[]
3     n1 = len(L1)
4     n2 = len(L2)
5     if n1!=n2:
6         return 'Les listes ne sont pas de la même taille !'
7     else:
8         for i in range(0,n1):
9             res.append(L1[i]+L2[i])
10        return res
```

1.2 Calcul d'une somme

Question 1.

```
1 def sommepuissance(n,p):
2     som = 0
3     for i in range(0,n+1):
4         som += i**p
5     return som
```

Question 2. `print(sommepuissance(12,4)-sommepuissance(21,3))` affiche bien 7349.

NOTA : \t correspond à une tabulation et \n correspond à un retour à la ligne.

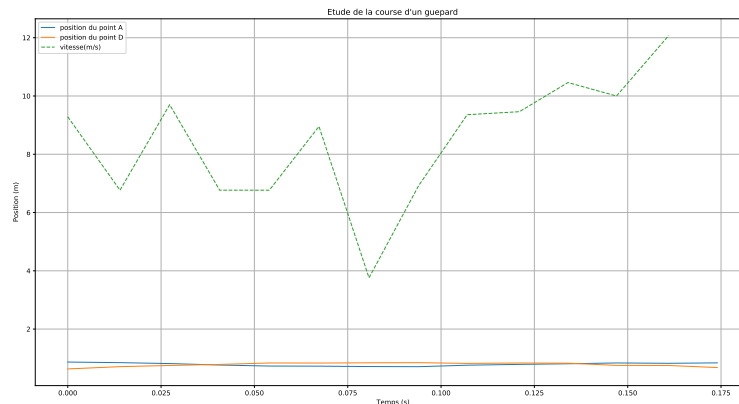
Question 2.

```
1 for i in range(3,len(texte)) :
2     ligne=texte[i].rstrip("\n") # Suppression du retour à la ligne
3     ligne=ligne.replace(",",".") # Suppression du retour à la ligne
4     ligne=ligne.split("\t")      # Découpage au niveau des tabulations
5     ligne = [float(ligne[i]) for i in range(0,18)] # Ajout des valeurs aux
tableaux T, Xi, Yi
6     donnees_completes.append(ligne)
```

Question 3.

```
1 for i in range(0,len(donnees_completes)):
2     T.append(donnees_completes[i][0])
3     XA.append(donnees_completes[i][1])
4     YA.append(donnees_completes[i][2])
5     XB.append(donnees_completes[i][4])
6     YB.append(donnees_completes[i][5])
7     XC.append(donnees_completes[i][7])
8     YC.append(donnees_completes[i][8])
9     YD.append(donnees_completes[i][11])
```

Et on obtient :



Question 4.

```
1 import matplotlib.pyplot as plt
2 plt.plot(T,YA,label='position du point A')
3 plt.plot(T,YD,label='position du point D')
4 plt.legend(loc='upper left')
5 plt.title(texte[0].rstrip("\n").rstrip("\t")) # Récupération ligne 1 pour le
titre
6 plt.ylabel('Position (m)') # Récupération ligne 2 pour le titre de l'axe y
7 plt.xlabel('Temps (s)') # Récupération ligne 3 pour le titre de l'axe x
8 plt.grid(True)# Affichage du quadrillage
9 plt.show() # Affichage du graphique
```

Question 5.

```
1 maxi = -1
2 mini = 1000
3 nvaleurs = len(T)
4 for i in range(0, nvaleurs):
5     if YA[i] > maxi:
6         maxi = YA[i]
7     elif YA[i] < mini:
8         mini = YA[i]
9 amplitude = maxi - mini
10 print(amplitude)
```

Question 6. On calcule la vitesse comme ceci :

```
1 # Calcul de la vitesse instantanée à partir des mesures
2 V = [] # Initialisation de la liste V
3 for i in range(1, nvaleurs) :
4     V.append(-(XA[i] - XA[i-1]) / (T[i] - T[i-1])) # V = dx/dt
5
6
7 # Tracé de la vitesse instantanée
8 Tmodif = T[:nvaleurs-1]
9 plt.plot(Tmodif, V, '--', label='vitesse (m/s)')
10 plt.legend(loc='upper left') # Insertion légende et placement
    dans le graphique
11
12 plt.show()
```

Question 7. Et l'angle du genou :

```
1 # Tracé angle genou
2 theta_g = []
3 import numpy as np
4 for i in range(0, nvaleurs):
5     pdt_norme = np.sqrt((XA[i] - XB[i])**2 + (YA[i] - YB[i])**2) * np.sqrt((XC[i]
6     ] - XB[i])**2 + (YC[i] - YB[i])**2)
7     cos_t_g = ((XA[i] - XB[i]) * (XC[i] - XB[i]) + (YA[i] - YB[i]) * (YC[i] - YB[i]
8     ])) / pdt_norme
9     theta_g.append((180. / np.pi) * np.arccos(cos_t_g))
10
11 plt.plot(T, theta_g)
```

Question 8. Ici, le guépard se déplace à une vitesse d'environ 30 km/h. Il trotte calmement !

4 Suite de Conway

Question 1.

```
1 def repetition(L):
2     i = 1
```

```

3 | while i<len(L) and L[i]==L[0]:
4 |     i+=1
5 | return i

```

Question 2.

```

1 | def repetitionbis(L):
2 |     n = repetition(L)
3 |     m = repetition(L[n:])
4 |     return [L[0],n,L[n],m]

```

Retour à la suite de Conway :

Question 3. $x_6 = 13112221$ et $x_7 = 1113213211$.

Question 4.

```

1 | def conway(L):
2 |     liste_resultat = []
3 |     i = 0
4 |     while i<len(L):
5 |         liste_resultat.append(repetition(L[i:]))
6 |         liste_resultat.append(L[i])
7 |
8 |         i = i + repetition(L[i:])
9 |
10 | return liste_resultat

```

Question 5.

```

1 | L = [1]
2 | for j in range(0,11):
3 |     print('x'+str(j)+'=' +str(L))
4 |     L = conway(L)

```

Demandez à votre professeur de mathématique pour une éventuelle démonstration !