



# Interrogation 1

## Algorithmie

### Correction

#### Exercice 1 :

Donner les syntaxes précises suivantes avec rédaction éventuelle :

1. Syntaxe d'une boucle **while** :

```
1 while condition : # condition est un
  booléen
2   instructions
```

2. Définition des fonctions **ord** et **chr** (avec remarques d'utilisation) :

La fonction **ord** prend en argument une chaîne de caractère composée d'un seul caractère et renvoie le numéro du caractère en question en Unicode. Donc **ord("a")** vaut 97.

La fonction **chr** fait le sens inverse de la fonction **ord** : **chr** prend en argument un entier et renvoie la caractère correspondant dans la table Unicode (donc **chr(97)** correspond à "a").

3. Sens de la commande **tab[i:j:k]** si **tab** est une liste :

Si **tab** est une liste (et en admettant que les indices *i* et *j* sont possibles pour la liste **tab**), la commande **tab[i:j:k]** correspond sous-liste de **tab** composée de tous les éléments de **tab** composée entre les indice *i* et *j* - 1 par pas de *k*.

4. Donner deux syntaxes pour rajouter un élément à la fin d'une liste **L** :

Si **L** est une liste, pour rajouter un élément **x** à la fin de **L**, on peut faire **L=L+[x]** ou **L.append(x)**.

5. Syntaxe de la boucle **for** :

```
1 for k in range(n,m,h) : # On peut remplacer
  range par n'importe quel itérable : une liste ou
  une chaîne de caractères
2   instructions
```

6. Comment obtenir le nombre d'élément d'une liste **L** ? Et le dernier élément ?

Si **L** est une liste, **len(L)** est la longueur de la liste, *i.e.* le nombre d'éléments de **L**.

Pour obtenir le dernier élément de **L**, on peut faire **L[len(L)-1]** ou **L[-1]**.

7. Définition du mot clé **assert** :

Le mot clé **assert** s'utilise dans une fonction et sert à levé une erreur si le booléen auquel il s'applique n'est pas vérifié.

8. Comment faire une documentation pour une fonction ?

Pour faire la documentation d'une fonction, on utilise une chaîne de caractère en tout début de fonction définit par trois guillemets :

```
1 def fct(var1:type1, ..., varn:typen) -> typef :
2   """ documentation de la fonction """
3   instructions
```

#### Exercice 2 :

Faire une fonction **diffCarre(tab:list)** -> **list** qui renvoie la liste des différences des carrés successifs des éléments de **tab** entre les éléments d'indice *i* et *i*-1. Par exemple, **diffCarre([3,-2,5,4])** renverra **[-5,21,-9]**.

```
1 def diffCarre(tab:list) -> list :
2   ntab=[]
3   for i in range(len(tab)-1) : # On s'arrête un cran avant la fin pour compenser le tab[i+1] et éviter
  de sortir de la liste
4       ntab = ntab + [tab[i+1]**2-tab[i]**2]
5   return(ntab)
```