NOM :
Prénom :

Vendredi 03 Octobre 2025

Interrogation 1

Algorithmie

Exercice 1 :
Donner les syntaxes précises suivantes avec rédaction éventuelle :

1. Syntaxe d'une boucle `while` :

```
while condition :
    instructions
```

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

La fonction `ord(car:str) -> int` prend en argument un caractère et renvoie son indice Unicode. La fonction `chr(n:int) -> str` fait le contraire : elle prend en argument un entier et renvoie le caractère correspondant dans la table Unicode. Par exemple, `ord("a")` renvoie 97. Et `chr(97)` renvoie `"a"`.

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

La commande `tab[i:j:k]` renvoie la sous-liste des éléments de `tab` entre les positions `i` et `j` en comptant par pas de `k`.

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

Pour rajouter un élément `x` à la fin de la liste `L`, on peut soit utiliser `L.append(x)` ; soit faire une concaténation `L=L+[x]`.

5. Syntaxe de la boucle `for` :

```
for i in range(n,m,k) :
    instructions
```

Il est possible de remplacer le "`i in range(n,m,k)`" par n'importe quel itérable (caractère dans une chaîne, éléments d'une liste, etc...)

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

La commande `len(L)` permet d'obtenir le nombre d'éléments de `L` (la longueur). Et pour avoir le dernier élément de `L`, on peut faire `L[-1]` ou `L[len(L)-1]`.

7. Définition du mot clé `assert` :

Le mot clé `assert booleen` sert, dans une fonction, à lever un message d'erreur si la condition du `assert` n'est pas vérifié.

8. Comment faire une documentation pour une fonction ?

La documentation d'une fonction se définie au moyen de trois guillemets (`"""`) et se place juste après le nom de la fonction.

Exercice 2 :
Faire une fonction `haut(tab:list, val:float) -> list` qui renvoie la liste des éléments de `tab` qui sont supérieurs ou égaux à `val`.

```
def haut(tab:list, val:float) -> list :
    L = []
    for elmt in tab :
        if elmt>=val :
            L.append(elmt)
    return(L)
```