

TD semaine 4 - Modélisation et Robotique

Listes et Variables globales

1 Révisions du cours

Question A. Valeurs de retour ou print ? Selon la définition choisie parmi :

```
def double(x):
    return 2 * x
```

```
def double(x):
    print(2 * x)
```

```
def double(x):
    print(2 * x)
    return 42
```

```
def double(x):
    return 42
    print(2 * x)
```

dire ce qui sera affiché par l'instruction `print("double de trois " , double(3)).`

Question B. Fonctions retournant une valeur. Quel est le rôle de `a` dans la définition de la fonction suivante ? Donner des exemples d'utilisation de cette fonction.

```
def avancer_toute():
    a = 0
    while not obstacle():
        av()
        a += 1
    return a
```

2 Traitements sur les Listes

Question C. Qu'affiche le programme suivant ?

```
panier = ["banane", "mangue", 1, "papaye"]
for i, e in enumerate(panier):
    print("Je mange", i+1, e)
```

Question D. Prénoms Définir une fonction qui prend une liste de prénoms (par exemple ['jean', 'pierre', 'marie']) et renvoie la chaîne de caractère 'jean, pierre et marie'

Question E. Fonction mystère On définit la fonction mystère ci-dessous :

```
def mystere(liste):
    liste2=[]
    for i, e in enumerate(liste):
        if i > e:
            liste2 += [i]
    return liste2
```

Que font les instructions suivantes :

```
print(mystere([1, 0, 516, 4, 2, 0]))
print(mystere(["42", "3", "1", "1"]))
```

Question F. Traiter des listes de nombres Définir les fonctions suivantes :

- Une fonction qui renvoie la différence entre le maximum et le minimum dans une liste de nombres

- une fonction `filtrePos(L)` qui ne renvoie que les nombres non négatifs.
Par ex., `filtrePos([1,-10,0,10,4])` renvoie `[1,0,10,4]`
- Une fonction `somsucc(L)` qui renvoie la somme des paires d'éléments successifs.
Par ex., `somsucc([1,10,4,3,3])` renvoie `[11,14,7,6]`
- Une fonction `somsucck(L,k)` qui renvoie la somme des `k` éléments successifs.
Par ex., `somsucc([1,10,4,3,3],3)` renvoie `[15,17,10]`.
Vous pouvez utiliser la fonction `sum` de Python qui calcule la somme des éléments d'une liste
- Une fonction `argsomsucc(L)` qui renvoie l'index de la paire d'éléments dont la somme est la plus grande.
Par ex., `argsomsucc([1,10,4,3,3])` renvoie `1` car l'élément `1` de la liste `[11,14,7,6]` est le plus grand.

Pour cela nous avons commencé à écrire le code suivant qu'il ne reste plus qu'à compléter. Pour rappel, en Python, `x/y` calcule le reste de la division de `x` par `y`.

```
orientation = 0
def tgo():
    ...
    orientation = orientation + 1
    tg()
def quelle_orientation():
    if orientation % 4 == 1:
        return 'nord'
    ...
```

- Que doit-on écrire à la première ligne de `tgo()` et pourquoi ?
- Compléter le programme.

Question I. Variables globales ou passage de valeur ? (exercice optionnel) Grâce à votre code, il est possible d'exécuter le programme suivant :

```
def dto():
    tgo(); tgo()
def trajet():
    av(); tgo(); av(); av(); tgo()
    av(); av(); dto()
```

`trajet(); print(quelle_orientation())`

Proposer une solution pour écrire le même programme sans utiliser de variable globale. Pourquoi est-ce mal adapté ici ?

Question J. Compteur kilométrique (exercice optionnel) Sur le même modèle que l'orientation absolue, proposer une fonction `avk()` qui remplace `av()` et une fonction `kilometrage()` qui retourne le nombre de cases parcourues par le personnage depuis le début du programme.

3 Déplacements et Orientation

Question G. Aller-retour. Définir une fonction `aller_retour()` qui fait que le personnage avance dans la direction actuelle jusqu'à un premier obstacle puis revient à la position initiale.

Indice : vous pouvez appeler la fonction `avancer_toute` dans votre code.

Question H. Orientation absolue. Au début du programme le personnage est orienté vers l'est. Le but de cet exercice est de définir trois fonctions : `tgo()` qui tourne à gauche, `tdo()` qui tourne à droite et `quelle_orientation()` qui retourne l'orientation actuelle sous la forme d'une chaîne de caractère parmi 'sud', 'nord', 'est', 'ouest'.

Question K. Trouver le mur. Vous pouvez utiliser les fonctions définies jusqu'à maintenant pour les exercices qui suivent.

Votre personnage est dans une cour rectangulaire, le long d'un des murs, orienté vers l'est. L'objectif est de trouver de quel côté est le (ou les) mur(s).

- Définir une fonction `dirObstacles()` qui retourne la liste de toutes les directions dans lesquelles se trouve un obstacle dans la case adjacente au personnage. Votre fonction retournera son résultat sous la forme d'une liste de points cardinaux (nord,sud...). Cette fonction est utile pour cette question et elle le sera encore plus, plus tard.

Indice : utilisez les fonction `tgo,tdo` et `quelle_orientation`

- Définir une fonction `orienter(s)` qui prend en paramètre une direction et qui oriente le personnage vers cette direction (par exemple `orienter('nord')`).
- Définir une fonction `trouverMur()` qui oriente le personnage en direction d'un mur adjacent (s'il y en a un).
- Définir une fonction `regarde()` qui utilise la fonction `dirObstacles` pour afficher 'couloir' si le personnage est dans un couloir (c'est à dire entouré de 2 murs) et 'cul-de-sac' (c'est à dire entouré de 3 murs) s'il est dans un cul-

de-sac, ou rien sinon.

4 Fruits et Légumes

Question L. Ingrédients. Définir une fonction `types_elements(panier)` qui prend en entrée une liste de légumes (par exemple, la liste `['chou', 'chou', 'salade']`) et retourne une liste où chaque élément du panier apparaît une seule fois.

Question M. Pas de fruits. Le panier peut contenir toute sorte de légumes mais aussi quelques fruits parmi une liste connues : `types_de_fruits`. Écrire une fonction `legumes` qui prend en entrée le panier et la liste des types de fruits et retourne les éléments du panier qui ne sont pas des fruits.

Question N. Pas de légumes. Écrire une fonction `fruits` qui, à partir des mêmes paramètres, `panier` et `types_de_fruits`, retourne tous les fruits du panier.

Question O. Les légumes puis les fruits. Toujours à partir des mêmes paramètres, définir une fonction `ranger` qui retournera une liste contenant d'abord tous les légumes du panier puis tous les fruits.

Indice : utilisez les fonctions précédentes