TP6 (objectif : comprendre ce qui se passe dans la mémoire lors de l'appel d'une fonction et comprendre la notion de portée ou visibilité des variables)

Gestion de la mémoire

Dans le TP sur la récursivité et celui sur la façon dont la mémoire gère une variable de type simple ou de type structuré, il est important de comprendre ce qui se passe lors de l'appel d'une fonction.

Par exemple, dans le programme ci-dessous, à la première boucle du for, au moment de l'instruction print(f(i)) :
Un espace mémoire est alloué pour l'exécution du corps de cette fonction
Dans cet espace :

- une case c1 recopie la valeur de la case mémoire qui est réservée à la variable i (ici 0 dans la première boucle)
- une case c2 est réservée pour t et contient 7
- une case c3 est réservée pour y et contient le résultat de 3x(le contenu de c1)+20+(le contenu de c2)
- c3 contient ce que renvoie ou retourne f(i)

l'instruction « print(f(i) » affichera alors le contenu de c3
et tout l'espace utilisé lors de cet appel à f est libéré

Première partie du TP

Portée

```
2  def f(x):
3
4      t=7
5      y=3*x+20+t
6      return y
7
8  a=4
9  print(f(4))
10 for i in range(3):
11     print(f(i))
12
13
```

Voici la liste des variables utilisées : x, y, t, a, i
Pour connaître leur portée (là où elles sont visibles), vous allez insérer l'instruction print(une de ces variables) sur une des lignes 3(dans le corps de f) 7(principal) 12 (dans le corps du for) et 13(principal)

Que peut-on conclure , dans le cas de Python

Deuxième partie du TP

1) Rajoutez l'instruction « a=a+2 » à la ligne3
Il y a une erreur pourtant a est globale, commentez ?
2) Continuez le mini projet