La pile d'exécution

1 Fonctionnement général

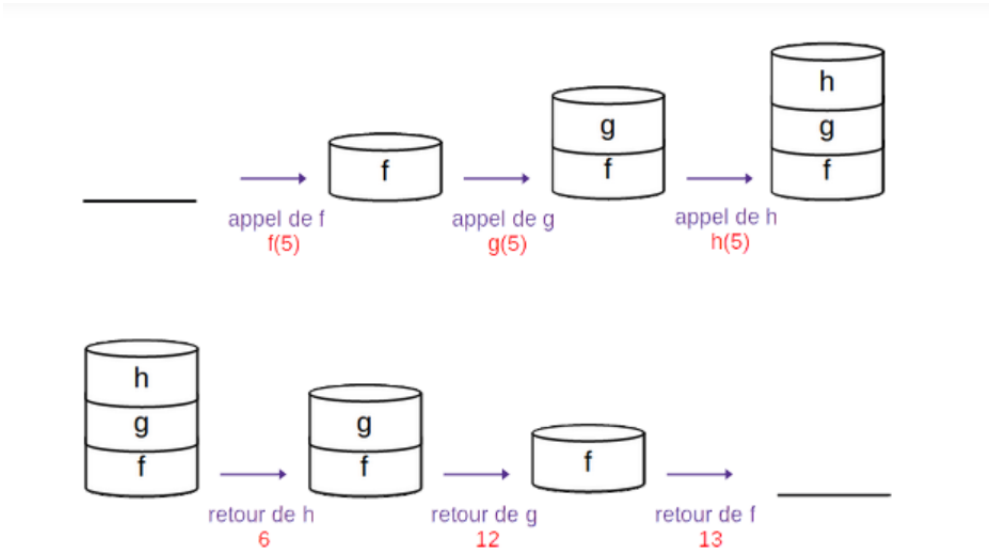
Lors de l'appel d'une fonction, le système sauvegarde différents paramètres comme ses arguments et les variables utilisées à l'intérieur de celle-ci. C'est son contexte d'exécution . Cela permet d'interrompre l'exécution d'une fonction lorsque celle-ci en appelle d'autres, et de reprendre son exécution ensuite.

Exemple :

```
def h(x):
    return x + 1

def g(x):
    return h(x) * 2

def f(x):
    return g(x) + 1
```



2 Fonctionnement dans le cas de la récursivité

Une fonction récursive est une fonction dans laquelle il y a des appels de fonctions, avec la particularité qu'il s'agit de la même fonction. La même chose se produit donc : une pile d'exécution est utilisée pour stocker le contexte d'exécution de ses différents appels. On peut comparer cette pile à une pile de crêpes : impossible de prendre la première crêpe tant que celles du dessus n'ont pas été enlevées.

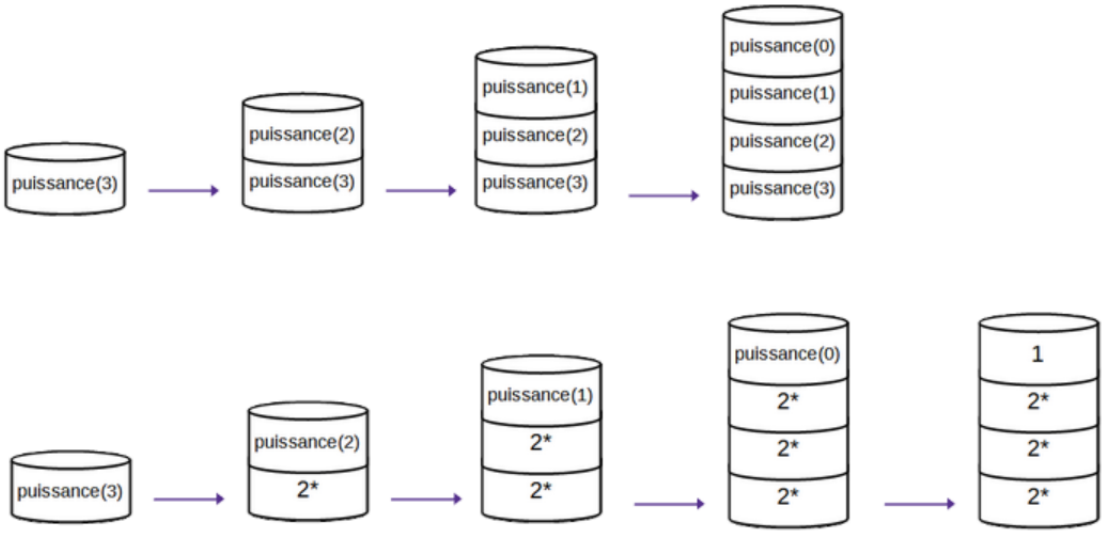
```
def deux_puissance(n):
    if n == 0:
        return 1

    return 2*deux_puissance(n-1)
```

Exercice 1

Écrire sur papier les différents appels récursifs à l'appel de fonction `deux_puissance(3)` :

Le schéma suivant explique le processus en terme de pile d'exécution :



3 Limites de la pile d'exécution

La pile d'exécution a une taille maximale. Lorsque la pile déborde - *stack overflow* - (par défaut au bout de 1000 appels), Python renvoie le message suivant :

```
RecursionError: maximum recursion depth exceeded while calling a Python object
```