

Les structures linéaires

1. A part les piles, quelles sont les deux autres types de structures linéaires étudiées cette année ?
2. Attribuer à chaque situation décrite ci-dessous, la structure linéaire la mieux adaptée :
 - Gestion d'un serveur d'impression
 - Gestion de l'appel de fonctions (récursives)
 - Tableaux de valeurs (noms, notes, villes,....)

Pour implémenter une structure de pile, on a besoin d'un nombre réduit d'opérations de bases :

- **creer_pile_vide** : fonction renvoyant une pile vide.
- **empiler** : fonction ajoutant un élément e sur une pile p , p et e étant transmis en paramètres.
- **depiler** : fonction enlevant et renvoyant un élément d'une pile transmise en paramètre.
- **est_vide** : fonction renvoyant vrai si la pile transmise en paramètre est vide, faux sinon.

3. Indiquer l'état des piles $p1$ et $p2$ après l'exécution du code suivant :

```
p1 = creer_pile_vide()
p2 = creer_pile_vide()
empiler(p1, 'A')
empiler(p1, 'B')
empiler(p1, 'C')
empiler(p1, 'D')
empiler(p1, 'E')
while est_vide(p1) == False:
    n = depiler(p1)
    empiler(p2, n)
```

4. On implémente une pile à l'aide d'une structure de type *list*. Compléter les lignes de code nécessaires :

```
def creer_pile_vide():
    '''renvoie une pile vide'''
    return .....

def empiler(p, x):
    "Ajoute l'élément x à la pile p"
    p. ....

def depiler(p):
    '''dépile et renvoie l'élément au sommet de la pile p'''
    if not pile_vide(p):
        return p. ....
    else : print("pile vide")

def est_vide(p):
    '''renvoie True si la pile est vide et False sinon'''
    ..... (il y a plusieurs réponses possibles, sur
    ..... un nombre de lignes variables)
```