

Activité du cours 4, corrigé

Piles et Files

1 Structure de pile

```
# Module de gestion de piles et de files

# fonctions pour les gestions de piles sous forme de listes
def creer():
    '''renvoie une pile ou une file vide sous forme de liste.'''
    return([])

def empiler(pile, elt):
    '''ajoute l'élément elt dans la pile ou file passée en argument.'''
    if type(elt) != list:
        pile.append(elt)
        return(None)
    if elt == []:
        return(None)
    pile += elt

def depiler(pile):
    '''retire un élément de la pile passée en argument.'''
    return(pile.pop())

def lire_sommet(pile):
    '''renvoie le dernier élément de la pile sans la modifier.'''
    return(pile[-1])

def afficher(pile):
    '''affiche la pile ou file passée en argument.'''
    for elt in pile:
        print(elt)

def inverser(pile):
    '''inverse l'ordre de la pile ou de la file passée en argument.'''
    pile = [pile[len(pile) - 1 - i] for i in range(len(pile))]
```

2 Structure de file

```
def enfiler(file,elt):
    '''ajoute l'élément elt dans la file.'''
    # c'est la même fonction que empiler
    if type(elt) != list:
        file.append(elt)
        return(None)
    if elt == []:
        return(None)
    file += elt

def defiler(file):
    '''renvoie le premier élément de la file passée en argument et la file
    modifiée.'''
    elt = file[0]
    nouv_file = file[1:]
    # Ici, contrairement à la pile, on est obligé de retourner file car
    # l'opération file = elt + file fait une nouvelle affectation du résultat
    # à une adresse différente de file entrée en argument.
    # Cette nouvelle adresse reste dans l'espace local de la fonction enfile
    # et ne modifie donc pas la file initiale qui conserve son adresse et
    # donc ses données. Subtil !!!
    return (elt, nouv_file)
```