

## 5.5

## Implémenter une file avec Python

NSI TLE - JB DUTHOIT

## 5.5.1 Implémenter une file avec les tableaux dynamiques python

Voici comment implémenter facilement une file avec les méthodes des tableaux dynamiques python (type `list`) :

```
ma_file = []
# Pour créer une file vide
ma_file.append(ma_valeur)
# permet d'enfiler une valeur
ma_file.pop(0)
# permet de défiler une valeur
len(ma_file)
# renvoie la longueur de ma_file
```

## 5.5.2 Construire une file avec deux piles

## Principe

Soient `p1` et `p2` deux piles vides.

## Principe pour enfiler

1. Avant d'enfiler `elt`, on déplace les éventuels éléments présents dans `p2` vers `p1` par dépilages et empilages successifs.
2. On enfile `elt` dans `p2` (qui est vide)
3. On déplace chaque élément de `p1` vers `p2` par dépilages et empilages successifs.

## Principe pour défiler

1. Pour défiler, il suffit de dépiler sur `p2`.

## Exemple pour enfiler 1, puis 2 puis 3 :

1. Initialisation :

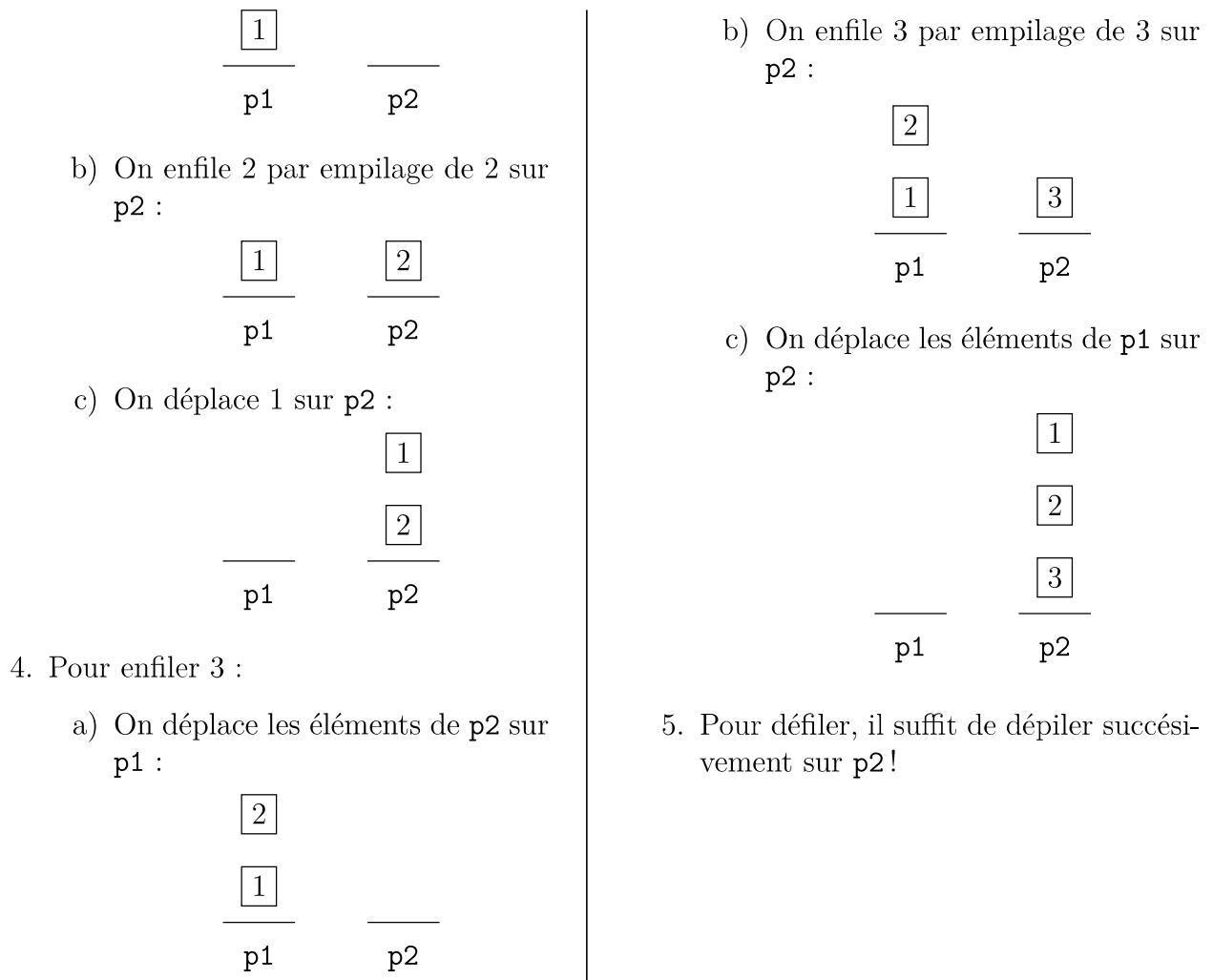
$\overline{\quad}$        $\overline{\quad}$   
p1      p2

2. On enfile 1 par empilage de 1 sur `p2` :

$\overline{\quad}$        $\overline{\boxed{1}}$   
p1      p2

3. Pour enfiler 2 :

- a) On déplace 1 sur `p1` :



### 5.5.3 Implémenter des files avec des classes

On pourra ici utiliser les tableaux dynamiques python par exemple.

#### Exercice 5.75

Créer la classe `File` qui dispose des méthodes suivantes :

- `estvide()` qui renvoie `True` si la file est vide, `False` sinon.
- `enfiler(elt)` qui permet d'enfiler un élément à la file
- `defiler()` qui défile un élément et le renvoie.
- `affiche()` qui affiche la file

### 5.5.4 Implémenter des files avec des listes chaînées

#### Exercice 5.76

On considère la classe `Cellule` suivante :

```
class Cellule:
    '''Une cellule d'une liste chaînée'''
    def __init__(self, v, s):
        self.valeur = v
```

```
self.suivante = s
```

Construire la classe `File` qui permettra d'instancier une pile. (cf le type abstrait de la pile pour les méthodes)