

## TD3 : structures linéaires - piles et files

### 1 Exercice : Pile

On se donne trois piles P1, P2 et P3. La pile P1 contient des nombres entiers positifs. Les piles P2 et P3 sont initialement vides. En n'utilisant que ces trois piles (et aucune autre structure de données linéaire) :

**Q1.1** Écrire un algorithme qui place les entiers pairs de P1 dans P2 et les entiers impairs de P1 dans P3. Après l'exécution de l'algorithme, P1 est vide.

**Q1.2** Écrire un algorithme qui place les entiers de P1 dans P2 en plaçant les entiers impairs au-dessus des entiers pairs. Après l'exécution de l'algorithme, P1 est vide et les entiers de P2 sont dans l'ordre inverse de l'ordre initial.

*Exemple : si au début P1 contient 1, 2, 3, 4 et 5, alors P2 contient 5, 3, 1, 4 et 2 après l'exécution de l'algorithme.*

**Q1.3** Écrire un algorithme qui modifie l'ordre des entiers de P1 de façon à avoir une alternance entre entiers pairs et impairs ; si le nombre d'entiers pairs et d'entiers impairs n'est pas le même, les entiers en surplus sont placés au sommet de la pile P1.

*Exemple : si au début P1 contient 4, 7, 1, 5 et 6, alors P2 contient 7, 1, 4, 5 et 6 après l'exécution de l'algorithme.*

Maintenant, on se donne deux piles seulement P1 et P2. La pile P1 n'est pas vide et contient des nombres entiers positifs. La pile P2 est initialement vide. En n'utilisant que ces deux piles (et aucune autre structure de données linéaire) :

**Q1.4** Écrire un algorithme qui renvoie l'élément situé à la base de la pile P1 ; après l'exécution, la pile P1 contient exactement les mêmes éléments qu'au début et dans le même ordre.

**Q1.5** On veut écrire un algorithme qui effectue une permutation circulaire d'ordre 1 des entiers de la pile P1 ; ainsi, le 1<sup>er</sup> élément se retrouve en dernière position, le 2<sup>me</sup> élément devient le 1<sup>er</sup>, etc.

### 2 Exercice : vérification de parenthésage

Dans cet exercice, vous allez écrire deux algorithmes de vérification de la validité d'un parenthésage. Ces algorithmes utilisent une pile d'entiers P ; ils retournent 0 si le parenthésage est correct et -1 dans le cas contraire.

**Q2.1** Écrire l'algorithme en supposant que la séquence de parenthèses est transmise à l'algorithme sous forme d'une file F dont les éléments sont soit '(' soit ')'.  
*Exemple : ()((()())) est correct et ()()() est incorrect.*

**Q2.2** Écrire l'algorithme en supposant que la séquence de parenthèses est transmise à l'algorithme sous forme d'une file F dont les éléments sont soit des parenthèses '(' soit ')', soit des crochets '[', ']'.

*Exemple : [()()](()) est correct et [] est incorrect.*

### 3 Exercice : File

On se donne trois files F1, F2 et F3. La file F1 contient des nombres entiers positifs. Les files F2 et F3 sont initialement vides. En n'utilisant que ces trois files (et aucune autre structure de données linéaire) :

- Q3.1** Écrire un algorithme pour placer tous les entiers de F1 qui sont des multiples de 5 dans F2. Faire en sorte que la file F1 soit vide après l'exécution de l'algorithme.
- Q3.2** Écrire un algorithme pour copier le contenu de F1 dans F2. Le contenu de F1 après exécution de l'algorithme doit être identique à celui avant exécution.
- Q3.3** Écrire un algorithme pour placer tous les entiers de F1 qui sont des multiples de 3 dans F2. Le contenu de F1 après exécution de l'algorithme doit être identique à celui avant exécution.
- Q3.4** Écrire un algorithme qui place les entiers pairs de F1 dans F2 et les entiers impairs de F1 dans F3. Après l'exécution de l'algorithme, F1 est vide.
- Q3.5** Écrire un algorithme qui place les entiers de F1 dans F2 en plaçant les entiers impairs derrière les entiers pairs. Après l'exécution de l'algorithme, F1 est vide et les entiers de F2 sont dans l'ordre initial.

*Exemple : si au début F1 contient 1, 2, 3, 4 et 5, alors F2 contient 2, 4, 1, 3 et 5 après l'exécution de l'algorithme.*

On se donne maintenant une seule file F non vide et qui contient des nombres entiers positifs. En n'utilisant que cette file (et aucune autre structure de données linéaire) :

- Q3.6** Écrire un algorithme qui effectue une permutation circulaire d'ordre 1 des entiers de la file F ; le 1<sup>er</sup> élément se retrouve en dernière position, le 2<sup>me</sup> élément devient le 1<sup>er</sup>, etc.  
*Exemple : si au départ F contient les éléments 1, 2 et 3, après l'exécution F contient 2, 3 et 1.*
- Q3.7** Écrire un algorithme qui renvoie “vrai” si la file contient au moins un entier pair et “faux” dans le cas contraire ; après l'exécution, la file contient exactement les mêmes éléments qu'au début et dans le même ordre.

### 4 Exercice : Pile et File

- Q4.1** Montrer comment implémenter une file en utilisant deux piles ; écrire les opérations **enfiler**, **défiler**.
- Q4.2** Montrer comment implémenter une pile en utilisant deux files ; écrire les opérations **empiler**, **dépiler**.