

Algorithmique et Structures de Données

TD n°6

Piles et Files

Licence Informatique 2ème année
Université de Nice-Sophia Antipolis

1 Piles et Files

Une **pile** est une structure de données de type LIFO (last in first out) : le dernier entré est le premier sorti.

On supposera que l'on dispose des primitives suivantes :

- $\text{DONNÉE}(elt)$: renvoie la donnée associée à l'élément elt .
- $\text{ESTVIDE}(P)$: renvoie vrai si la pile est vide, faux sinon.
- $\text{SOMMET}(P)$: renvoie l'élément sommet de la pile P .
- $\text{DÉPILER}(P)$: supprime de la pile P le sommet.
- $\text{EMPLER}(P, elt)$: ajoute au sommet de la pile P l'élément elt .

Question : écrivez les fonctions suivantes.

1. $\text{afficher}(P)$: cette fonction affiche tous les éléments de la pile.
2. $\text{dépilerKelt}(P, k)$: cette fonction dépile k éléments si la pile contient au moins k éléments, sinon elle dépile toute la pile.
3. $\text{dépilerJusquà}(P, elt)$: cette fonction dépile la pile jusqu'à l'élément elt . L'élément elt n'est pas dépilé. Si l'élément n'appartient pas à la pile, alors la fonction dépile toute la pile.

Une **file** est une structure de données de type FIFO (first in first out) : le premier entré est le premier sorti.

On supposera que l'on dispose des primitives suivantes :

- $\text{DONNÉE}(elt)$: renvoie la donnée associée à l'élément elt .
- $\text{ESTVIDE}(F)$: renvoie vrai si la file est vide, faux sinon.
- $\text{PREMIER}(F)$: renvoie le premier élément de la file F .
- $\text{DÉFILER}(F)$: supprime de la file F le premier élément.
- $\text{ENFILER}(F, elt)$: ajoute dans la file F l'élément elt .

Question : écrivez les fonctions suivantes.

1. $\text{afficher}(F)$: cette fonction affiche tous les éléments de la file.
2. $\text{défilerJusquà}(F, elt)$: cette fonction défile la file jusqu'à l'élément elt . L'élément elt n'est pas défilé. Si l'élément n'appartient pas à la file, alors la fonction défile toute la file.

Question : écrivez les fonctions suivantes. On pourra éventuellement utiliser une ou des piles/files temporaires, on utilisera les primitives $\text{CRÉERPILE}()$ qui renvoie une pile vide et $\text{CRÉERFILE}()$ qui renvoie une file vide.

1. $\text{appartient}(P, elt)$: cette fonction renvoie vrai si l'élément appartient à la pile, faux sinon. Attention, il est important que la pile ne change pas.
2. $\text{inverser}(F)$: cette fonction inverse les éléments de la file F . On a le droit d'utiliser des files ou des piles temporaires.

3. $inverser(P)$: cette fonction inverse les éléments de la pile P . On interdit l'utilisation de files. Seules des piles temporaires peuvent être utilisées.

2 Notation polonaise inverse

La section suivante est empruntée à Wikipédia. La Notation Polonaise Inverse (NPI) a été inventée par le philosophe australien et informaticien Charles Hamblin dans le milieu des années 1950, pour permettre les calculs sans adresse mémoire.

Avec la NPI, les opérandes précèdent l'opérateur. Cette notation permet donc de se passer des parenthèses. Par exemple, l'expression $3 * (4 + 7)$ s'écrira $4 7 + 3 *$ ou bien $3 4 7 + *$.

En pratique, sur une calculatrice de NPI, le calcul sera saisi comme suit :

« 4 », « entrée », « 7 », « + », « 3 », « * »

ou « 3 », « entrée », « 4 », « entrée », « 7 », « + », « * »

La réalisation de calculatrices NPI est basée sur l'utilisation d'une pile ; c'est-à-dire, que les opérandes sont ajoutés en haut de la pile, et les résultats des calculs sont remis en haut de la pile.

Le calcul $((1 + 2) * 4) + 3$ peut être noté en NPI comme ceci : $1 2 + 4 * 3 +$ ou $3 4 1 2 + * +$.

L'expression est évaluée de la façon suivante (la pile est montrée après chaque opération) :

	Entrée	Opération	Pile
Étape no 1	1	Pousser l'opérande	1
Étape no 2	2	Pousser l'opérande	1, 2
Étape no 3	+	Addition	3
Étape no 4	4	Pousser l'opérande	3, 4
Étape no 5	*	Multiplication	12
Étape no 6	3	Pousser l'opérande	12, 3
Étape no 7	+	Addition	15

L'algorithme consiste donc à parcourir l'expression de gauche à droite, puis

à empiler les opérandes rencontrés,

à dépiler 2 opérandes à effectuer l'opération et à empiler le résultat quand un opérateur est rencontré.

Question : écrivez de façon détaillé l'algorithme d'un calcul.

3 Extrait du contrôle partiel 2009-2010

Une maîtresse d'école décide d'instaurer un système de points correspondant à la participation des élèves. A chaque élève, est associée une pile de type LIFO qui mémorise les différentes réponses qu'il a faites. Si un élève répond bien à une question, alors un feu vert est empilé sur sa pile. Si un élève répond de façon approximative à une question, alors un feu orange est empilé sur sa pile. Enfin, si un élève répond mal à une question, alors un feu rouge est empilé sur sa pile.

Un élève a 0 point initialement, puis les points sont comptés de la façon suivante : si la pile contient 3 fois de suite la même couleur de feu, alors le nombre de points de l'élève est incrémenté de 10 points si les feux sont verts, 5 points s'ils sont orange et décrétementé de 10 points s'ils sont rouges.

En outre, si 3 feux identiques apparaissent au sommet de la pile, alors les trois feux sont dépilés de la pile.

- Détaillez le principe mis en place par la maîtresse, le nombre de points obtenus et la pile finale pour l'élève Nicolas qui a reçu la suite de feux suivants (V désigne un feu vert, O un orange et R un rouge)

VVVOOORRRVVOORRROVVV

- Écrivez une fonction qui prend en entrée la pile d'un élève munie des primitives vues en cours, et un tableau de feux et qui modifie la pile afin de calculer la valeur associée à l'élève et qui renvoie cette valeur. On pourra considérer que la pile est vide lors de l'appel de la fonction.