

Algorithmique 1

Feuille 3 : Pile et file

1 Pile

Dans les exercices suivants on considère le type :

```
type_pile= pile de type_predefini;
```

défini en cours avec les primitives suivantes :

```
fonction valeur(ref P:pile de objet):objet;
fonction pileVide(ref P:pile de objet):booléen;
fonction créerPile(P:pile de objet);
fonction empiler(ref P:pile de objet;
                 val x:objet):vide;
fonction dépiler (ref P:pile de objet):vide;
fonction detruirePile(P:pile de objet):vide;
```

Exercice 3.1

Evaluer à l'aide des primitives du type abstrait Pile la fonction suivante et donner le contenu de la pile après exécution.

```
fonction essaiPile():pile de char;
var P: pile de char;
var C:char;
  debut
    creerPile(P);
    empiler(P,"A");
    depiler(P);
    empiler(P,"B");
    C=valeur(P);
    empiler(P,"a");
    empiler(P,C);
    retourner(P);
  fin
finfonction
```

Exercice 3.2

On se donne trois piles P_1 , P_2 et P_3 , la pile P_1 contient une suite de nombres entiers positifs.

1. Ecrire un algorithme pour déplacer les entiers de P_1 dans P_2 de façon à avoir dans P_2 tous les nombres pairs au dessous des nombres impairs.
2. Ecrire un algorithme pour copier dans P_2 les nombres pairs contenus dans P_1 . Le contenu de P_1 après exécution de l'algorithme doit être identique à celui avant exécution. Les nombres pairs dans P_2 doivent être dans l'ordre où ils apparaissent dans P_1 .

Exercice 3.3

Utiliser une pile pour

1. Evaluer une expression arithmétique postfixée codée sur un tableau de caractères, en supposant pour simplifier que
 - tous les opérateurs sont binaires et limités à $+, -, *, /$,
 - on utilise uniquement des nombres sur un caractère,
2. Transformer une expression arithmétique infixée en une expression arithmétique postfixée codée sur un tableau de caractères, en supposant pour simplifier que tous les opérateurs sont binaires et limités à $+, -, *, /$.

Exercice 3.4

Un problème fréquent d'un compilateur et des traitements de textes est de déterminer si les parenthèses d'une chaîne de caractères sont balancées et proprement incluses l'une dans l'autre. Par exemple, la chaîne $((()) ()) ()$ est bien balancée et proprement écrite. Mais les chaînes $) ($ (ou $()$) ne le sont pas.

Ecrire une fonction :

1. Qui retourne TRUE si une chaîne de caractères est proprement écrite et bien balancée, et FALSE sinon.
2. Qui retourne la position de la première parenthèse qui déroge à cette règle si la chaîne n'est pas bien écrite et bien balancée.

2 Files

Dans les exercices suivants on considère le type :

```
type_file= file de type_predefini;
```

défini en cours avec les primitives suivantes :

```
fonction valeur(ref F:file de objet):objet;  
fonction fileVide(ref F:file de objet):booléen;  
fonction créerFile(F:file de objet);  
fonction enfiler(ref F:file de objet;  
                val v:objet):vide;  
fonction défiler(ref F:file de objet):vide;  
fonction detruireFile(F:file de objet):vide;
```

Exercice 3.5

Evaluer à l'aide des primitives du type abstrait File la fonction suivante et donner le contenu de la file après exécution.

```
fonction essaiFile():file de char;  
var F: file de char;  
var C:char;  
debut  
    créerFile(F);  
    enfiler(F,"A");  
    enfiler(F,"B");
```

```

    enfiler(F,"C");
    defiler(F);
    C=valeur(F);
    enfiler(F,"a");
    defiler(F);
    enfiler(F,"b");
    enfiler(F,C);
    retourner(F);
fin
finfonction

```

Exercice 3.6

Soit L une liste d'entiers strictement croissante et un entier N . Ecrire une fonction `compter` qui dénombre les couples $(x, x+N)$ avec x et $x+N$ présents dans la liste.

Exercice 3.7

Un palindrome est une chaîne de caractères qui se lit de la même manière de gauche à droite et de droite à gauche. En utilisant un nombre fixe de piles et de files, des primitives du type `Pile` et `File`, et un nombre fixe de variables de type `int` et `char`, écrire un algorithme qui détermine si une chaîne de caractère est un palindrome. On suppose que la chaîne de caractères est passée en paramètre dans un tableau qui ne sera parcouru qu'une seule fois. L'algorithme doit donner en sortie `TRUE` ou `FALSE` selon le cas.

Problème récurrent (notre fil d'ariane)

Exercice 3.8 Gestion d'une piste d'atterrissage des avions

Un avion est un enregistrement contenant :

- l'indicatif (6 caractères)
- la destination (30 caractères)
- l'autonomie résiduelle de carburant comptée en heures de vol (entier)
- deux booléens indiquant s'il y a un pirate à bord et s'il y a le feu.

1. Définir les structures de données nécessaires.
2. Ecrire la fonction `Priorité` ainsi que la gestion complète de la piste.
3. Envisager le cas de suppression d'un élément quelconque de la file lorsque le pirate a mis sa menace de détournement à exécution.

Quelles sont les notions que vous venez de voir qui peuvent permettre d'amorcer le fil d'ariane ?