

Algorithmique 1 : Devoir Surveillé 2

Pile et files

Durée : 45mn

Sans documents

Exercice 2.1 *Implémentation d'une pile dans un tableau*

Soit l'implémentation d'une pile de caractères dans un tableau :

```
pile de car= structure
    taille: entier;
    sommet: entier;
    pile: tableau[1..taille] de car
finstructure
```

1. Ecrire les primitives `creerPile`, `pileVide`, `valeur`, `empiler` et `depiler` dans cette implémentation.
2. Soit la pile `p` implémentée par un tableau de `taille=5` et la suite de primitives ci-dessous.

```
empiler(p, '1');
empiler(p, 'a');
empiler(p, '3');
empiler(p, 'B');
char c= valeur(p);
empiler(p, 'B');
empiler(p, c);
empiler(p, '1');
empiler(p, 'd');
char b= valeur(p);
depiler(p);
depiler(p);
empiler(p, 'b');
```

Dessiner le contenu de la pile après l'exécution de cette suite à partir d'une pile vide.

Exercice 2.2 *Implémentation d'une file dans une liste listeDC*

Soit l'implémentation d'une file de caractères dans une liste `listeDC` de caractères.

```
file de car= listeDC de car
```

1. Ecrire les primitives `creerFile`, `fileVide`, `valeur`, `enfiler` et `defiler` dans cette implémentation.
2. Soit la file `f` de caractères implémentée dans `listeDC` de caractères et la suite de primitives ci-dessous.

```
enfiler(f, '1');
enfiler(f, 'a');
enfiler(f, '3');
```

```

enfiler(f,'B');
char c= valeur(f);
enfiler(f,'B');
enfiler(f,c);
enfiler(f,'1');
enfiler(f,'d');
char b= valeur(f);
defiler(f);
defiler(f);
enfiler(f,b);

```

Dessiner le contenu de la file après l'exécution de cette suite à partir d'une file vide.

Exercice 2.3 *Application*

On souhaite construire un mini éditeur ligne. On donne une chaîne contenant les commandes d'édition. Il s'agit d'afficher la suite de caractères résultant de l'exécution de ces commandes. Ces caractères sont stockés dans un tampon. Chaque caractère est une commande ; on distingue quatre catégories de caractères :

- tout caractère différent des trois caractères # , \$ et \newline est un caractère du texte, on le stocke dans le tampon ;
- le caractère # signifie supprimer le caractère précédent dans le tampon ;
- le caractère \$ signifie supprimer tous les caractères du tampon ;
- le caractère \newline signifie la fin de chaîne de commande et déclenche l'affichage du tampon constitué par les caractères du texte.

Ainsi par exemple, la chaîne "ABC\$abce#de#\newline" sera transformée en "abcd".

Pour réaliser ces opérations, on représentera le tampon par une pile de caractères.

ANNEXE A **Type abstrait** *pile de objet*

```

fonction valeur(val P:pile de objet):objet;
fonction pileVide(val P:pile de objet):booleen;
fonction creerPile(ref P:pile de objet): vide;
fonction empiler(ref P:pile de objet, val x:objet):vide;
fonction depiler (ref P:pile de objet):vide;
fonction detruirePile(ref P:pile de objet):vide;

```

ANNEXE B **Type abstrait** *file de objet*

```

fonction valeur(val F:file de objet):objet;
fonction fileVide(val F:file de objet):booleen;
fonction creerFile(ref F:file de objet): vide;
fonction enfiler(ref F:file de objet, val v:objet):vide;
fonction defiler(ref F:file de objet):vide;
fonction detruireFile(ref F:file de objet):vide;

```