1 Utilisation des piles

Exercice nº 1

Exercice nº 2 Implémentation d ?une pile par tableau

Rappel : La pile est une structure de type LIFO (Last In, First Out), on accède uniquement au sommet et la première valeur accessible est la dernière ajoutée. Pour manipuler une pile, on définit les fonctions suivantes :

- ***estVide*** : teste si la pile est vide (retourne un booléen)
- ***afficherPile*** : affiche les eléments de la pile.
- ***empiler*** : ajoute un élément au sommet de la pile
- ***depiler*** : retourne lobjet situé au sommet de la pile et le retire

1. Écrire et tester la fonction est ***estVide***

```
>>>tab1=[]
>>>tab2=[”p”,”y”,”t”,”h”,”o”]
>>> estVide(tab1)
True
>>>estVide(tab2)
False
```

2. Écrire et tester la fonction est ***afficherPile***

```
>>>afficherPile(tab2) o
h
t
y
p
```

3. Écrire et tester la fonction est ***empiler***

```
>>>empiler(tab2,”n”)
[”p”,”y”,”t”,”h”,”o”,”n”]
```

4. Écrire et tester la fonction est ***depiler***

```
>>>depile(tab2)
[”p”,”y”,”t”,”h”,”o”]
>>>depile(tab2)
[”p”,”y”,”t”,”h”]
>>>depile(tab2)
[”p”,”y”,”t”]
```

5. Écrire une fonction qui permet de tester si un élément est présent dans une pile.

```
>>>estPresent(tab2,”y”)
True
>>>estPresent(tab2,”a”)
False
```

1. Écrire une fonction qui prend en paramètre une chaîne de caractère et qui inverse celle-ci.

```
>>>inverse(”Hello”)
olleH
```

2. Réécrire la fonction inverse, mais en utilisant une structure de pile

Exercice nº 3

Nous allons étudier ici la vérification des délimiteurs dans une expression.
On distingue ici trois délimiteurs, les parenthèses”(” et ”)”, les accolades ”{” et ”}” et les crochets ”[” et ”]”. L ?objectif est de déterminer si une chaîne de caractères (qui pourrait être un programme) est correctement délimitée, c ?est-à-dire si chacun des délimiteurs ouvrant est suivi par un délimiteur fermant.

1. Les exemples suivants sont-ils bien délimitées ?

a(b)c ; a[b(c)}d ; {a(b[c[d))]e]} ; {a(b(d[d]))e}

2. Est-il possible de vérifier une chaîne uniquement en comptant les délimiteurs ? Si non, proposez un contre-exemple.
3. Écrivez une fonction ***verifDelim*** qui prend en paramètre une chaîne de caractère est qui renvoie un boolean indiquant si la chaîne est correctement délimitée. Cette méthode ne considérera que les délimiteurs "(" et ")"
4. Écrivez une fonction ***verifDelimStack*** qui réalise vérification de délimiteurs à partir d ?une PILE.
5. **[BONUS]** Modifiez votre algorithme de manière à afficher les indices des caractères qui posent problème, par exemple lorsqu ?il manque un délimiteur fermant ou qu ?il ne s ?agit pas du délimiteur fermant correspondant au délimiteur ouvrant.

Exercice n° 4 Notation polonaise inversée : NPI

Avecla NPI,les opérandes (nombres) précèdent l'opérateur (+, -, /,*).
Par exemples :
23+ signifie $2+3$ $12+34+*$ signifie $(1+2)*(3+4)$
Pour effectuer le calcul d'une NPI, on applique l'algorithme suivant :
On lit un élément à la fois.

- Si c ?est un opérande (un entier), on l ?empile
- Si c ?est un opérateur binaire, on dépile deux éléments sur lesquelles on applique l ?opérateur et on empile le résultat.

Appliquer cet algorithme pour calculer les expressions ci-dessous :

1. 4 5 * 2 * 1 −
2. 7 3 + 7 3 3 * +
3. 5 9 3 + 4 2 * * 7 + *
4. 3 7 + 6 5 − * 4 2 − 2 3 + * /

Remarques : L'avantage de cette notation est que les calculs peuvent être écrit sans parenthèses.

2 Utilisation des files

Exercice n° 5

Exercice n° 6 Implémentation d ?une file par tableau

Rappel : La file est une structure de type FIFO (First In, First Out), on accède uniquement au début de la file et la première valeur accessible est la première ajoutée. Pour manipuler une file, on définit les fonctions suivantes

- ***estVide*** : teste si la file est vide (retourne un booléen)
- ***afficherFile*** : affiche les éléments de la file.
- ***emfiler*** : ajoute un élément au sommet de la file
- ***defiler*** : retourne l'objet situé au sommet de la file et le retire

1. Écrire et tester la fonction est ***estVide***

```
>>>tab1=[]
>>>tab2=["p","y","t","h","o"]
>>> estVide(tab1)
True
>>>estVide(tab2)
False
```

2. Écrire et tester la fonction est ***afficherPile***

```
>>>tab=["H", "e" ,"l", "l"]
>>>afficherPile(tab) "Hell"
```

3. Écrire et tester la fonction est ***emfiler***

```
>>>emfiler(tab,"o")
["H", "e" ,"l", "l", "o"]
```

4. Écrire et tester la fonction est ***depiler***

```
>>>depile(tab)
["H", "e" ,"l", "l"
>>>depile(tab)
["H", "e" ,"l"
>>>depile(tab)
["H", "e"
```

5. Écrire une fonction qui permet de tester si un élément est présent dans une pile.

```
>>>tab="azerty" >>>estPresent(tab,"y")
True
>>>estPresent(tab,"u")
False
```