



Listes, Piles et Files (Exercices)

S2 - Structures de données

Les exercices précédés du symbole  sont à faire sur machine, en sauvegardant le fichier si nécessaire.

Les exercices précédés du symbole  doivent être résolus par écrit.

Exercice 1 (listes)

On reprend l'implémentation des listes avec des tuples présentée dans le cours.

Prévoir l'effet et l'affichage en console des instructions suivantes, puis vérifier en exécutant ces instructions dans la console interactive après avoir créé et importé un fichier contenant le code du cours :

```
>>> L = creer()
>>> est_vide(L)
>>> L = ajouter(5, ajouter(4, ajouter(3, ajouter(2, ajouter(1, ajouter(0, ()))))))
>>> est_vide(L)
>>> longueur(L)
>>> L = ajouter(6, L)
>>> longueur(L)
>>> tete(L)
>>> queue(L)
>>> longueur(queue(L))
```

Exercice 2 (listes)

Soit la suite d'instructions suivantes :

```
L = creer()
L = ajouter(2, ajouter(15, ajouter (23, L)))
L1 = queue(L)
a = tete(L1)
L1 = ajouter(4, ajouter(3, L1))
```

Donnez le contenu des listes L et L1 et la valeur de a.

Exercice 3 (piles)

Soit une pile P initialement vide. Soit les instructions suivantes (implémentation des piles avec des listes Python) :

```
>>> empiler(P,4)
>>> empiler(P,7)
>>> a = depiler(P)
>>> b = taille(P)
>>> c = depiler(P)
>>> empiler(P,3)
>>> empiler(P,2)
>>> d = taille(P)
```

Donnez le contenu de la pile `P`, la valeur de `a`, la valeur de `b`, la valeur de `c` et la valeur de `d`.

Exercice 4 (piles)

Soit le programme Python suivant (on utilise l'implémentation des piles en POO) :

```
pile = Pile()
tab = [5,8,6,1,3,7]
for k in tab:
    pile.empiler(k)
pile.empiler(5)
pile.empiler(10)
pile.empiler(8)
pile.empiler(15)
for k in tab:
    if k > 5:
        pile.depiler()
```

Donnez l'état de la pile `pile` après l'exécution de ce programme.

Exercice 5 (piles)

Ce problème propose une application concrète des piles. Il s'agit d'écrire une fonction qui contrôle si une expression mathématique, donnée sous forme d'une chaîne de caractères, est bien parenthésée, c'est-à-dire s'il y a autant de parenthèses ouvrantes que de fermantes, et qu'elles sont bien placées

Par exemple :

- `(..(..)..)` est bien parenthésée
- `(...(..)...)` ne l'est pas

L'algorithme :

- On crée une pile
- On parcourt l'expression de gauche à droite.
- À chaque fois que l'on rencontre une parenthèse ouvrante “(” on l'empile
- Si on rencontre une parenthèse fermante “)” et que la pile n'est pas vide on dépile (sinon on retourne faux)
- À la fin la pile doit être vide...

1. En utilisant l'une des structures pile du cours, écrire une fonction `verification(expr)` qui vérifie si une expression mathématique passée en paramètre est correctement parenthésée.
2. Proposer un jeu de tests unitaires vérifiant le bon fonctionnement de la fonction.
3. Faire en sorte que le programme tienne compte également des `[`.

Exercice 6 (files)

Soit une file `F` initialement vide. Soit les instructions suivantes :

```
enfiler(F,6)
enfiler(F,3)
a = defiler(F)
enfiler(F,9)
b = taille_file(F)
enfiler(F,17)
c = defiler(F)
enfiler(F,2)
d = taille_file(F)
```

Donnez le contenu de la file `F`, la valeur de `a`, la valeur de `b`, la valeur de `c` et la valeur de `d`.

Exercice 7 (files)

Soit le programme Python suivant :

```
file = File()
tab = [2,78,6,89,3,17]
file.enfiler(5)
file.enfiler(10)
file.enfiler(8)
file.enfiler(15)
for i in tab:
    if i > 50:
        file.defiler()
```

Donnez l'état de la file `file` après l'exécution de ce programme

Exercices tombés au bac

- 2021 Métropole Jour 1 : exo 2.
- 2021 Centres étrangers Jour 2 : exo 1.
- 2021 Centres étrangers Jour 1 : exo 5.
- 2021 Amérique du Nord Jour1 : exo 5.
- 2021 Sujet zéro : exo 1.
- 2022 Centres étrangers Jour 1 : exo 2.
- 2022 Métropole Jour 1 : exo 1.
- 2022 Métropole Jour 2 : exo 2.
- 2022 Mayotte Liban Jour 1 : exo 1.

- 2022 Mayotte Liban Jour 2 : exo 1.
- 2022 Amérique du Nord Jour 1 : exo 5.