

Tp Manipulation de piles

Créer un notebook `tp_manipuler_piles.ipynb` : y recopier votre classe `Pile` et créer une cellule de code pour chaque exercice.

Exercice 1 :

1) Créer une pile `p1` qui contiendra les valeurs suivantes :

10
9
5

`p1`

2) Il y a une erreur de saisie : il fallait mettre au milieu de la pile la valeur 8 au lieu de la valeur 9. Modifier le contenu de la pile `p1` (sans créer de nouvelle pile).

Exercice 2 :

Créer les quatre piles suivantes :

--

`p2` (pile vide)

100

`p3`

'O'
'F'
'N'
'I'

`p4`

100
...
3
2
1

`p5`

Exercice 3 :

Ecrire une fonction qui prend comme paramètre d'entrée une pile et permute les deux éléments situés au sommet de la pile. On supposera que la pile en entrée a une taille au moins égale à 2 et la fonction ne retournera rien.

On l'appellera `permute` et son entête est :

```
def permute(pile):
```

Exercice 4 :

a) Ecrire une fonction `troisieme_elem(pile)` qui prend comme paramètre d'entrée une pile et qui retourne l'antépénultième élément de cette pile. Le contenu de la pile ne sera pas modifié. On supposera que la pile en entrée a une taille au moins égale à 3. La tester avec la pile `p4`

b) Ecrire une fonction `dixieme_elem(pile)` qui prend comme paramètre d'entrée une pile et qui retourne le dixième élément de cette pile (dixième en comptant à partir du sommet de la pile). Le contenu de la pile ne sera pas modifié. On supposera que la pile en entrée a une taille au moins égale à 10. La tester avec la pile `p5`.

c) Ecrire une fonction `base_pile(pile)` qui prend comme paramètre d'entrée une pile et qui retourne l'élément situé à la base de la pile. Le contenu de la pile ne sera pas modifié. On supposera que la pile en entrée n'est pas vide. La tester avec les différentes piles.

Exercice 5 :

1) Ecrire une fonction `inverse(pile)` qui prend une pile en argument et renvoie une autre pile constituée des mêmes éléments placés dans l'ordre inverse. La pile peut être modifiée suite à l'exécution de la fonction

2) Ecrire une fonction `inversebis(p)` qui prend une pile `p` en argument et renvoie une autre pile constituée des mêmes éléments placés dans l'ordre inverse. Le contenu de la pile `p` ne doit pas être modifié.