NSI · TERMINALE / EXERCICES CORRIGÉS

Listes, piles et files

Astrolab Cours · astrolabcours.fr

Ex. 1 Suivre une pile ●○○ Application

On part d'une pile vide. On empile 3, 7, 2, on dépile, on empile 5, on dépile deux fois. Que contient la pile ? Quelles valeurs ont été dépilées ?

Indice

On retire toujours le sommet.

Étape par étape

[3, 7, 2] → dépile 2 → [3, 7] → [3, 7, 5] → dépile 5 → dépile 7 → [3].

La réponse

La pile contient 3. Valeurs dépilées : 2, 5, 7.

Ex. 2 Les parenthèses bien placées ●●○ Méthode

Écrire une fonction `bien_parenthesee(s)` qui vérifie, avec une pile, qu'une chaîne ne contenant que `(` et `)` est bien parenthésée.

Indice

Empiler à chaque ouvrante, dépiler à chaque fermante.

L'idée

Une fermante doit correspondre à une ouvrante encore ouverte. À la fin, la pile doit être vide.

© Astrolab Cours · astrolabcours.fr/nsi/terminale/listes-piles-files/ · Tous droits réservés : usage personnel uniquement, republication et usage commercial interdits

La fonction

```
def bien_parenthesee(s):
    pile = []
    for c in s:
        if c == "(":
            pile.append(c)
        else:
            if pile == []:
                return False
            pile.pop()
```

Ex. 3 La file trop lente ●●○ Piège

Un programme gère une file de 100 000 clients avec `file.pop(0)`. Il est très lent. Pourquoi, et que faire ?

Indice

Que fait `pop(0)` aux autres éléments ?

L'erreur

`pop(0)` retire le premier élément puis décale tous les autres d'une case : 100 000 décalages à chaque fois.

La solution

Utiliser `collections.deque` et `popleft()`, qui retire en temps constant.

Ex. 4 Une file avec deux piles ●●● Type bac

On implémente une file avec deux piles `entree` et `sortie`. Enfiler : empiler sur `entree`. Défiler : si `sortie` est vide, on y transvase toute la pile `entree`, puis on dépile `sortie`. Écrire `defiler` et vérifier l'ordre sur l'exemple : enfiler 1, 2, 3, puis défiler deux fois.

Indice

Transvaser renverse l'ordre.

L'exemple

`entree` = [1, 2, 3]. Défiler : `sortie` vide, on transvase : `sortie` = [3, 2, 1]. On dépile 1, puis 2.

La fonction

On obtient 1 puis 2 : c'est bien une file.

```
def defiler(entree, sortie):
    if sortie == []:
        while entree != []:
            sortie.append(entree.pop())
    return sortie.pop()
```

© Astrolab Cours · astrolabcours.fr/nsi/terminale/listes-piles-files/ · Tous droits réservés : usage personnel uniquement, republication et usage commercial interdits

© Astrolab Cours · astrolabcours.fr/nsi/terminale/listes-piles-files/ · Tous droits réservés : usage personnel uniquement, republication et usage commercial interdits