Devoir surveillé n°4 – NSI
27/11/2023

Certaines réponses seront à écrire sur le sujet qui sera donc à rendre avec la copie.

Exercice 1 (Type épreuve pratique – 2 points)
On dispose de chaînes de caractères contenant uniquement des parenthèses ouvrantes et fermantes.

Un parenthésage est correct si :

- le nombre de parenthèses ouvrantes de la chaîne est égal au nombre de parenthèses fermantes.
- en parcourant la chaîne de gauche à droite, le nombre de parenthèses déjà ouvertes doit être, à tout moment, supérieur ou égal au nombre de parenthèses déjà fermées.

Ainsi, `"((()())(()))"` est un parenthésage correct.
Les parenthésages `"())(()"` et `"(())(()"` sont, eux, incorrects.

On dispose du code de la classe Pile suivant :

```
class Pile:
    """
    Classe définissant une pile
    """
    def __init__(self):
        self.valeurs = []

    def est_vide(self):
    """
    Renvoie True si la pile est vide, False sinon
    """
        return self.valeurs == []

    def empiler(self, c):
    """
    Place l'élément c au sommet de la pile
    """
        self.valeurs.append(c)

    def depiler(self):
    """
    Supprime l'élément placé au sommet de la pile,
    à condition qu'elle soit non vide
    """
        if self.est_vide() == False:
            self.valeurs.pop()
```

On souhaite programmer une fonction `parenthesage` qui prend en paramètre une chaîne de caractères `ch` formée de parenthèses et renvoie `True` si la chaîne `ch` est bien parenthésée et `False` sinon.
Cette fonction utilise une pile et suit le principe suivant :
En parcourant la chaîne de gauche à droite, si on trouve une parenthèse ouvrante, on l'empile au sommet de la pile et si on trouve une parenthèse fermante, on dépile (si possible) la parenthèse ouvrante stockée au sommet de la pile.
La chaîne est alors bien parenthésée si, à la fin du parcours, la pile est vide.
Elle est, par contre, mal parenthésée :

- si dans le parcours, on trouve une parenthèse fermante, alors que la pile est vide ;
- ou si, à la fin du parcours, la pile n'est pas vide.

Compléter le code suivant de la fonction `parenthesage` :

```
def parenthesage(ch):
    """
    Renvoie True si la chaîne ch est bien parenthésée et False sinon
    """
    p = Pile()
    for c in ch:
        if c == ...:
            p.empiler(c)
        elif c == ...:
            if p.est_vide():
                return ...
            else:
                ...
    return p.est_vide()
```

Exercice 2 (Type épreuve écrite – 8 points)

Dans cet exercice, on considère une pile d'entiers positifs. On suppose que les quatre fonctions suivantes ont été programmées préalablement en langage Python :

`empiler(P, e)` : ajoute l'élément `e` sur la pile `P` ;

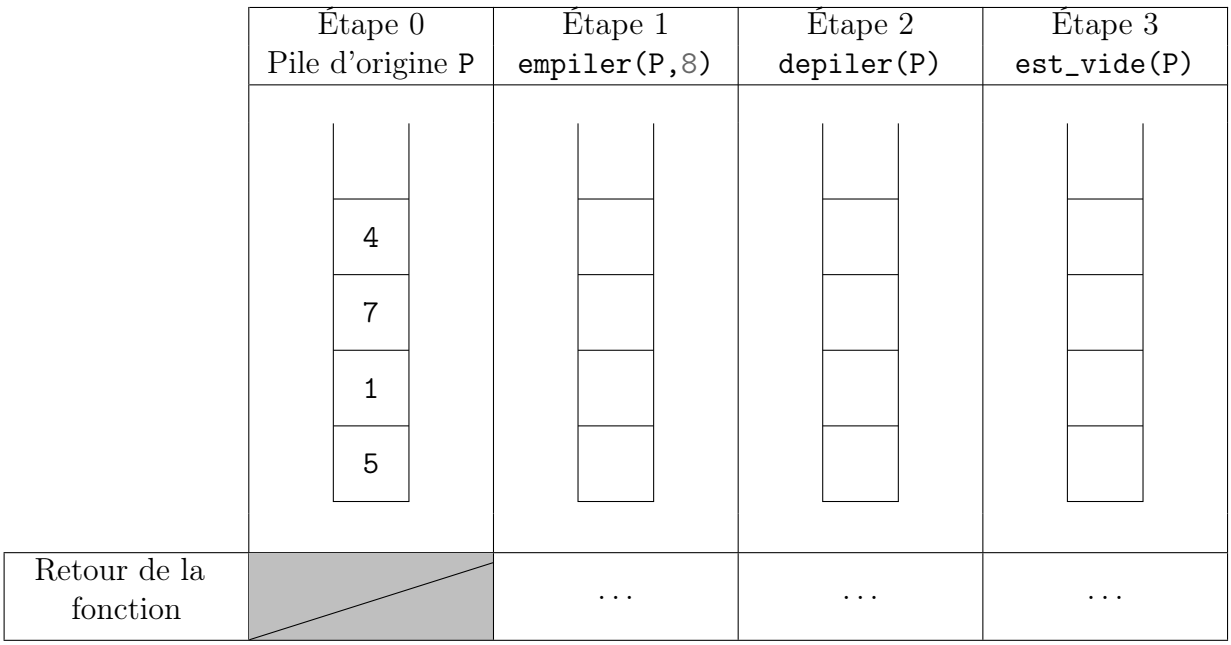
`depiler(P)` : enlève le sommet de la pile `P` et retourne la valeur de ce sommet ;

`est_vide(P)` : retourne `True` si la pile est vide et `False` sinon ;

`creer_pile()` : retourne une pile vide.

Dans cet exercice, seule l'utilisation de ces quatre fonctions sur la structure de données pile est autorisée.

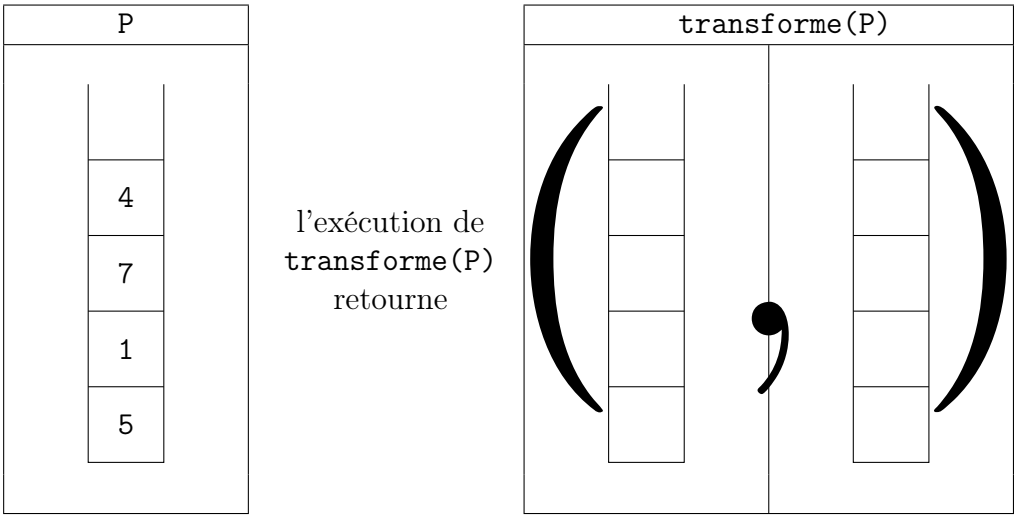
1. Compléter le schéma ci-dessous en exécutant les appels de fonctions donnés. On écrira ce que renvoie la fonction utilisée dans chaque cas, et on indiquera `None` si la fonction ne retourne aucune valeur.



2. On propose la fonction ci-dessous, qui prend en argument une pile `P` et retourne un couple de piles :

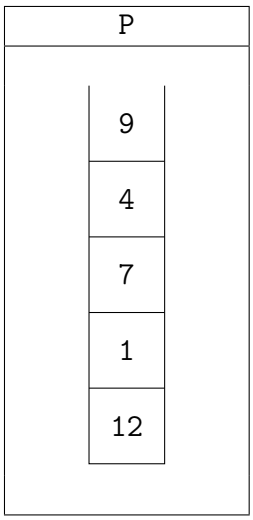
```
def transforme(P) :
    Q = creer_pile()
    while not est_vide(P) :
        v = depile(P)
        empile(Q,v)
    return (P,Q)
```

Compléter la figure ci-dessous :



3. Écrire une fonction en langage Python `maximum(P)` recevant une pile `P` (que l'on supposera non vide) comme argument et qui retourne la valeur maximale de cette pile.
On ne s'interdit pas qu'après exécution de la fonction, la pile soit vide.

4. On souhaite connaître le nombre d'éléments d'une pile à l'aide de la fonction `taille(P)`.



`taille(P)` retournera donc l'entier 5

On souhaite cependant qu'à la fin de l'exécution de la fonction `taille(P)`, la pile `P` ait retrouvé son état initial.

(a) Proposer une stratégie écrite en langage naturel et/ou expliquée à l'aide de schémas, qui permette de mettre en place une telle fonction.

(b) Donner le code Python de cette fonction `taille(P)`.
(on pourra éventuellement utiliser la fonction `transforme(P)` déjà programmée en plus des quatre fonctions initiales sur les piles).