

Devoir surveillé n°4 – NSI
Correction

Exercice 1

Le code est le suivant :

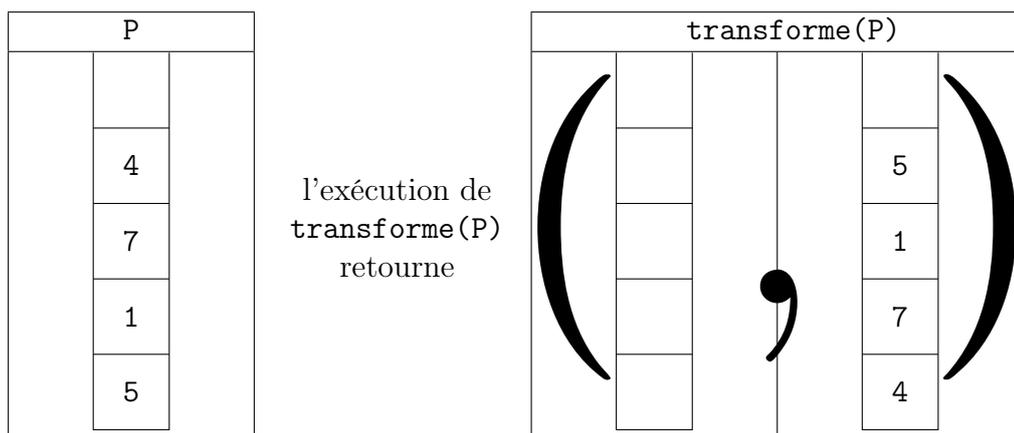
```
def parenthesage(ch):
    """
    Renvoie True si la chaîne ch est bien parenthésée et False sinon
    """
    p = Pile()
    for c in ch:
        if c == "(":
            p.empiler(c)
        elif c == ")":
            if p.est_vide():
                return False
            else:
            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

Exercice 2

1.

	Étape 0 Pile d'origine P		Étape 1 empiler(P,8)		Étape 2 depiler(P)		Étape 3 est_vide(P)	
				8				
		4		4		4		4
		7		7		7		7
		1		1		1		1
		5		5		5		5
Retour de la fonction				None		8		False

2.



3. Le code est le suivant :

```
def maximum(P):
    m=depiler(P)
    while not est_vide(P):
        v = depiler(P)
        if v > m:
            m = v
    return m
```

4. (a) Il suffit de mettre place une boucle qui s'arrêtera quand la pile P sera vide. À chaque tour de boucle, on dépile P, on empile les valeurs précédemment dépilées dans une pile auxiliaire Q et on incrémente un compteur de 1. Une fois la boucle terminée, on crée une nouvelle boucle où on dépile Q et on empile P avec les valeurs dépilées (l'idée est de retrouver l'état originel de pile). Il suffit ensuite de retourner la valeur du compteur.

(b) Le code est le suivant :

```
def taille(P):
    t = 0
    Q = creer_pile()
    while not est_vide(P):
        v = depiler(P)
        empiler(Q,v)
        t = t + 1
    while not est_vide(Q):
        v = depiler(Q)
        empiler(P,v)
    return t
```

Si on souhaite utiliser la fonction `transforme(P)` :

```
def taille(P):
    t = 0
    _,Q = transformer(P)
    while not est_vide(Q):
        empiler(P,depiler(Q))
        t += 1
    return t
```

L'idée est de « retourner » la pile P avec la fonction `transforme(P)` puis, successivement, en même temps que l'on empile les valeurs dépilées de Q, on ajoute 1 au compteur.