

Exercice 4 : Structures de Données – Les Piles

1.

```
def est_triee(self):
    if not self.est_vide():
        e1 = self.depiler()
        while not self.est_vide():
            e2 = self.depiler()
            if (e1 > e2):
                return False
            e1 = e2
    return True
```

2. a. L'appel A.est_triee() renvoie False.

2. b. Contenu de la pile A après l'exécution : [1, 2]

3.

```
def depileMax(self):
    assert not self.est_vide(), "Pile vide"
    q = Pile()
    maxi = self.depiler()
    while not self.est_vide():
        elt = self.depiler()
        if (maxi < elt) :
            q.empiler(maxi)
            maxi = elt
        else:
            q.empiler(elt)
    while not q.est_vide():
        self.empiler(q.depiler())
    return maxi
```

4. a.

	B	q
Fin tour 1	[9, -7, 8]	[4]
Fin tour 2	[9, -7]	[4, 8]
Fin tour 3	[9]	[4, 8, -7]
Fin tour 4	[]	[4, 8, -7, 9]

4. b.

	B	q
Avant la ligne 14	[9, -7, 8, 4]	[]

CORRECTION – NSI - 2022 Sujet Polynésie

4. c. Par exemple si on a $B = [9, 1, 2]$, à la fin de la première boucle «while» on a $q = [1, 2]$, puis quand on exécute «depileMax» sur B on obtient $B = [2, 1]$, l'ordre n'est pas préservé.

5. a.

	B	q
Avant la ligne 3	[1, 6, 4, 3, 7, 2]	[]
Avant la ligne 5	[]	[7, 6, 4, 3, 2, 1]
À la fin de l'exécution de la fonction	[1, 2, 3, 4, 6, 7]	[]

5. b. Cette méthode trie les valeurs de la pile d'entrée dans l'ordre croissant.