Index des sujets 2023

Corrigé sujet 10 - Année : 2023

Sujet 10 - 2023

Exercice 1

```python
def maxliste(tab):
    maxi = tab[0]
    for elt in tab:
        if elt > maxi:
            maxi=elt
    return maxi
```

Commentaires

L'énoncé précise que la liste est non vide, on peut donc se permettre d'initialiser le maximum courant avec le premier élément de la liste.

Exercice 2

```python
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()

```

```
20
21  def parenthesage (ch):
22      """Renvoie True si la chaîne ch est bien parenthésée et False
23  sinon"""
24      p = Pile()
25      for c in ch:
26          if c == "(": #1
27              p.empiler(c)
28          elif c == ")": #2
29              if p.est_vide():
30                  return False
31              else:
                    p.depiler()
      return p.est_vide() #3
```

1. On suit l'algorithme donné dans l'énonce : si on rencontre une parenthèse ouvrante alors on l'empile
2. Si c'est une parenthèse fermante, on dépile dans le cas où la pile est vide, l'expression est mal parenthésée.
3. Si à la fin du parcours la pile n'est pas vide, l'expression est mal parenthésée.

Remarque

L'utilisation d'une pile pour vérifier le bon parenthésage est surtout utile lorsqu'il y a plusieurs types de parenthèses ouvrantes et fermantes : `()` mais aussi `{}`, `[]` ...