Compte Rendu Série : 6

Par : HAJAR ZGAOUA

Exercice 1 :

```
def push(element):
    pile.append(element)

def is_empty():
    return len(pile)==0

def pop():
    if not is_empty():
        return pile.pop()
    else:
        return 0

def top():
    if not is_empty():
        return pile[-1]
    else:
        return 0

pile=[]
push(1)
push(2)
push(3)
print("notre pile:",pile)
print("l'element à supprimer est:",pop())
print("l'etat finale de la pile :",pile)
```

Sortie :

```
notre pile: [1, 2, 3]
l'element à supprimer est: 3
l'etat finale de la pile : [1, 2]
```

```
[ ]:
```

Exercice 2 :

Code :

[213]:
```
def verifier(chaine):
    pile=[]
    for carat in chaine:
        if carat=="(":
            pile.append(carat)
        elif carat == ')':
            if len(pile) == 0:
                return False
            pile.pop()
    return len(pile)==0
machaine="("
res=verifier(machaine)
print(res)
```

```
False
```

Exercice 3 :

Code :

[222]:
```
def inverser(chaine):
    pile=[]
    for caract in chaine:
        pile.append(caract)
    chaine_inverse = ""
    while pile:
        chaine_inverse += pile.pop()
    return chaine_inverse
machaine="hajar zgaoua"
res=inverser(machaine)
print(res)
```

```
auoagz rajah
```

[ ]:
```

```

Exercice 4 :

Code :

[18]:
```python
def est_equilibree(chaine):
    pile = []
    correspondances = {')': '(', ']': '[', '}': '{'}

    for caractere in chaine:
        if caractere in correspondances.values():
            pile.append(caractere)
        elif caractere in correspondances.keys():
            if not pile or pile[-1] != correspondances[caractere]:
                return False
            pile.pop()

    return not pile
print(est_equilibree("{[()]}"))
```

```
True
```

Exercice 5 :

Code :

```python
def afficher_file(file):
    print("File actuelle :", file)

def enfiler(file, personne):
    file.append(personne)
    print(f"{personne} entre dans la file.")
    afficher_file(file)

def defiler(file):
    if len(file) == 0:
        print("La file est vide.")
    else:
        personne = file.pop(0)
        print(f"{personne} sort de la file.")
        afficher_file(file)

file_attente = []
enfiler(file_attente, "HAJAR")
enfiler(file_attente, "siham")
enfiler(file_attente, "ali")
defiler(file_attente)
defiler(file_attente)
defiler(file_attente)
defiler(file_attente)
```

Sortie :

```
HAJAR entre dans la file.
File actuelle : ['HAJAR']
siham entre dans la file.
File actuelle : ['HAJAR', 'siham']
ali entre dans la file.
File actuelle : ['HAJAR', 'siham', 'ali']
HAJAR sort de la file.
File actuelle : ['siham', 'ali']
siham sort de la file.
File actuelle : ['ali']
ali sort de la file.
File actuelle : []
La file est vide.
```

Exercice 6 :

Code :

[16]:
```python
file=[]

#chaine="abcd"
#file.append(chaine)
file.append("a")
file.append("b")
file.append("c")
file.append("d")
print(file)
chaine_inv=""
while len(file) > 0:
   chaine_inv = file.pop(0) + chaine_inv
print(chaine_inv)
```

```
['a', 'b', 'c', 'd']
dcba
```