Exercice n° 1

Compléter le tableau avec les propositions ci-dessous :

1. LIFO
2. FIFO
3. L.pop()
4. L.pop(0)
5. len[L]==0
6. L.append("truc")
7. Distributeur PEZ
8. Parcours en largeur
9. Parcours en profondeur
10. Calculer une expression en notation polonaise inversée.
11. Gestion des clients ayant pré-commandé la PS5.

PILE	FILE	LES DEUX

Exercice n° 2

1. Que fais le programme ci-dessous ?
2. Les données contenues dans **Tab** sont-elles traitées comme une file ou comme une pile ? (Expliquer)

```
def enigme1(Tab) :
    """
    :param Tab : list
    :return Boolean
    """
    while Tab ! = [] :
        nbElt=len(Tab)
        if Tab[nbElt-1]=="a" :
            return True
        else :
            Tab.pop()
    return False
```

Exercice n° 3

```
# Question 1 :
On crée la liste suivante :
t = [ [1,2,3,4], [5,6,7,8], [9,10,11,12] ]
Que vaut t[1][2] ?
```

```
# Question 2 :

Quel est le résultat du script suivant :
L = [12,0,8,7,3,1,5,3,8]
a = [elt for elt in L if elt<4]
print(a)
```

```
# Question 3 :
On définit :
dico = { "Herve " : 15, "Kevin " :17, "Fatima " :16}
qui associe nom et âge de trois élèves.
Comment accéder à l'âge de Kevin ? :
```

1. dico[1]
2. dico[Kevin]
3. dico["Kevin "]
4. dico("Kevin ")

```
# Question 4 :
L est une liste d'entiers.
On définit la fonction suivante :
def f(L) :
    m = L[0]
    for x in L :
        if x < m :
            m = x
return m
Que calcule cette fonction ?
```

1. le premier terme de la liste L
2. le dernier terme de la liste L
3. le maximum de la liste L
4. le minimum de la liste L

Question 5 :

Parmi les propositions suivantes, lesquelles permettent de créer en Python la liste des nombres impairs de 1 à 399(inclus)

1. [i + 2 for i in range(1,200)]
2. [1 + nb*2 for nb in range(200)]
3. [nb for nb in range(400) if nb%2==1]
4. [1, 3, 5, 7, 9] * 40

Question 6 :

Quelle est la valeur affichée à l'exécution du programme Python suivant ?

```
L = [7,3,1,5,3,8]
L.pop(0)
L.append(7)
print(L)
```

1. [7,3,1,5,3,8]
2. [3,1,5,3,8,7]
3. [3,1,5,3,8]
4. 7

Question 7 :

On dispose du dictionnaire régions ci-dessous : regions = { 'Mayotte' : 376, 'Pays de la Loire' : 32082, 'La Réunion' : 2504, 'Grand Est' : 57441, 'Martinique' : 1128, 'Corse' : 8680, 'Bretagne' : 27208, 'Nouvelle-Aquitaine' : 84036 } Parmi les instructions suivantes, laquelle permet d'ajouter une nouvelle région ?

1. INSERT "'Hauts de France' :31806" INTO regions
2. regions = dict(['Hauts de France'] = 31806)
3. regions('Hauts de France') = 31806
4. regions['Hauts de France'] = 31806