DS2 : ITC - 2h

DS1 : ITC - Correction

Exercice 1

Variable	C	r	e	t	o	i	s
Valeur	True	3	14	6	12	-6	[True,[-6]]

Exercice 2

1. Premier appel : `"un phrase d tst"`, deuxième appel : `"unephrasedetest"`, troisième appel : `"une phrase de test"`.
2. La fonction prend en arguments une chaine de caractère `C` et un caractère `x` et renvoie une nouvelle chaine de caractère contenant les mêmes caractères que `C` sauf tous les `x` qui ont été supprimés.
3. Wikipédia + Perec la disparition.

Exercice 3 L'appel renvoie 5 : c'est un algorithme de maximum, mais on cherche ici la plus grande distance entre deux valeurs de `L`

Exercice 4

```
def est_monotone(L):
    n = len(L)
    test = 'c ou dc'
    for i in range(1, n-1):
        if test == 'c ou dc':
            if L[i+1] < L[i]:
                test = 'dc'
            elif L[i+1] > L[i]:
                test = 'c'
        if (test == 'c' and L[i+1] < L[i]) or (test == 'dc' and L[i+1] > L[i]):
            return False
    return True
```

Exercice 5

```
def f(L):
    if L[0] >= 0:
        return True
    L[2] = 2*L[2]
    L.append(L[0])
    return [e**2 for e in L]
```

Exercice 6

```
def Enleve_Repet(L):
    M = []
    for e in L:
        i = 0
        while i < len(M) and M[i] != e:
            i = i + 1
        if i == len(M):
            M.append(e)
    return M
```

Exercice 7

1.

```
def moy(U):
    s = 0
    for e in U:
        s = s + e
    return s / len(U)
```

2.

```python
def surtension(U):
    for e in U:
        if e > 220:
            return True
    return False
```

3.

```python
def amplitude(U):
    mini = U[0]
    maxi = U[0]
    for e in U:
        if e > maxi:
            maxi = e
        if e < mini:
            mini = e
    return maxi - mini
```

4.

```python
def sous(tab1, tab2):
    L = []
    for i in range(len(tab1)):
        L.append(tab1[i] - tab2[i])
    return L
```

5.

```python
def detect_front(U):
    L = []
    m = moy(U)
    for i in range(len(U) - 1):
        if U[i] < m and U[i+1] > m:
            L.append(i)
    return L
```

6.

```python
def periode(T, U):
    L = detect_front(U)
    return (T[L[-1]] - T[L[0]]) / (len(L) - 1)
```