

Nom :

Devoir 5

- 1) Ecrire une fonction `somme_double` prenant comme argument un entier  $n$  et retournant le calcul  $\sum_{i=1}^n \sum_{j=1}^n i^j$

```
def somme_double(n):
```

- 2) Ecrire une fonction `Deux_plus_proches_valeurs(L)` qui, étant donnée une liste de nombres  $L$  contenant au moins deux éléments, renvoie les deux valeurs les plus proches (en valeur absolu) sous forme d'une liste au moyen de deux boucles `for`.

```
def Deux_plus_proches_valeurs(L):
```

```
    d_min = abs(L[1]-L[0])
```

```
    n=len(L)
```

```
    couple=[L[0],L[1]]
```

```
    for i in range(.....):
```

```
        for j in range(.....):
```

```
            if .....
```

```
    return couple
```

- 3) On donne la fonction `remonte` ci-dessous. Utiliser cette fonction dans la fonction `tri(L)` afin que cette fonction `tri` renvoie une liste  $L$  d'entiers triés par ordre croissant.

```
def remonte(liste,j):
```

```
    for i in range(j):
```

```
        if liste[i]>liste[i+1]:
```

```
            liste[i],liste[i+1]=liste[i+1],liste[i]
```

```
    return liste
```

```
def tri(L):
```

```
    for j in range(.....):
```

```
        .....
```

```
    return L
```

Nom :

Devoir 5

- 1) Ecrire une fonction `somme_double` prenant comme argument un entier  $n$  et retournant le calcul  $\sum_{i=1}^n \sum_{j=1}^n i^j$

```
def somme_double(n):
```

- 2) Ecrire une fonction `Deux_plus_proches_valeurs(L)` qui, étant donnée une liste de nombres  $L$  contenant au moins deux éléments, renvoie les deux valeurs les plus proches sous forme d'une liste au moyen de deux boucles `for`.

```
def Deux_plus_proches_valeurs(L):
```

```
    d_min = abs(L[1]-L[0])
```

```
    n=len(L)
```

```
    couple=[L[0],L[1]]
```

```
    for i in range(.....):
```

```
        for j in range(.....):
```

```
            if .....
```

```
    return couple
```

- 3) On donne la fonction `remonte` ci-dessous. Utiliser cette fonction dans la fonction `tri(L)` afin que cette fonction `tri` renvoie une liste  $L$  d'entiers triés par ordre croissant.

```
def remonte(liste,j):
```

```
    for i in range(j):
```

```
        if liste[i]>liste[i+1]:
```

```
            liste[i],liste[i+1]=liste[i+1],liste[i]
```

```
    return liste
```

```
def tri(L):
```

```
    for j in range(.....):
```

```
        .....
```

```
    return L
```