

Exercice 1 :

res	l	
.	5 1 8 7 6 3 2 4	5 1 8 7 6 3 2 4
1	5 8 7 6 3 2 4	1 5 8 7 6 3 2 4
1 2	5 8 7 6 3 4	1 5 7 8 6 3 2 4
1 2 3	5 8 7 6 4	1 5 6 7 8 3 2 4
1 2 3 4	5 8 7 6	1 3 5 6 7 8 2 4
1 2 3 4 5	8 7 6	1 2 3 5 6 7 8 4
1 2 3 4 5 6	8 7	1 2 3 4 5 6 7 8
1 2 3 4 5 6 7	8	
1 2 3 4 5 6 7 8	.	

Complexité : quadratique

Exercice 2 :

```
def estPermu(T):
    Tab = [False]*len(T)
    for i in T:
        if (i <= len(T)) and (i >= 1) and (Tab[i-1] == False) :
            Tab[i-1] = True
        else :
            return False
    return True
```

```
def triPerm(T):
    Tab = [0]*len(T)
    for i in T :
        Tab[i-1] = i
    return Tab
```

```
def inverse(T):
    res = [0]*len(T)
    for i in range(len(T)):
        res[T[i]-1] = i+1
    return res
```

```
def produit(T1, T2):
    res = [0]*len(T1)
    for i in range(len(T2)):
        res[i] = T1[T2[i]-1]
    return res
```

Exercice 3 :

1/
Soit un tableau T de taille n
HR : à l'étape k : T[n-k : len(T)] correct
Pour k = 0 trivial
On suppose que c'est vrai pour k-1
A l'étape k :

```

i = n-k
Soit l tel que T[l] = T'[n-k]      T' : tableau trié
l <= n-k car T[n-k+1, n-1] = T'[n-k+1, n-1]
Forall m <= n-k, T[l] >= T[m]
Donc pour j de 0 à n-k-1
si T[j] = T[l] alors T[j] >= T[j+1]
si l = n-k alors T[n-k-1] <= T[n-k] donc pas d'échange

```

Complexité : quadratique

2/

Sur T[0, n-i] T[n-i+1, n-1] = T_{trié}[n-i+1, n-1]
Si aucun échange T[0, n-i] est trié donc T est trié

3/

Pas d'échange après T[j], T[j+1]
Le nouveau T[j+1] majore T[0, j+1]
Le nouveau T[j+1] majore T[j+1, n-1]
T[j+1, n-1] est trié et ne peut plus changer. On peut poser i=j.

4/

```

def TriABullesOptimise(T) :
    i = len(T)-1
    while i != 0 :
        k=0
        for j in range (0, i):
            if T[j] > T[j+1]:
                T[j], T[j+1] = T[j+1], T[j]
                k=j
        i = k
    return T

```

5/

```

def TriShaker(T):
    n = len(T)
    for i in range(n/2+1):
        for j in range(i-1, n-i-1):
            if(T[j] > T[j+1]):
                T[j], T[j+1] = T[j+1], T[j]
        for j in range(n-i-1, i, -1):
            if(T[j] < T[j-1]):
                T[j], T[j-1] = T[j-1], T[j]

```