

**A/ Soit le tableau t suivant :**

1	4	5	6	3
0	1	2	3	4

**Donner le résultat du script suivant appliqué au tableau ci-dessus**

1. `j = 4`
2. `temp=t[4]`
3. `while t[j-1]> temp:`
4. `t[j]=t[j-1]`

**Correction**

j	t[j-1]>temp	Exécution	t=
4			

**B/ Soit le tableau t suivant :**

1	4	5	6	3
0	1	2	3	4

**a) Donner le résultat du script suivant appliqué au tableau ci-dessus**

1. `j = 4`
2. `temp=t[4]`
3. `while t[j-1]> temp:`
4. `t[j]=t[j-1]`
5. `j = j-1`

**Correction**

j	t[j-1]>temp	Exécution	t=
4			
3			

2			
1			

...	...	...	...	....
0	1	2	3	4

**b) Appliquer le même script au tableau suivant**

3	4	5	6	1
0	1	2	3	4

j	t[j-1]>temp	Exécution	t=
4			
3			
2			
1			
0			

**c) Donner une amélioration pour que ce script ne cause aucun problème**

1.           j = 4
2.           temp=t[4]
3.           while t[j-1]> temp .....
4.           t[j]=t[j-1]
5.           j = j-1
- 6.

**d/ Placer l'élément temp à place adéquate**

## Le tri par insertion: Principe

- 1) considérer que les (i-1) premiers éléments de la liste sont triés et placer le i<sup>ème</sup> élément à sa place parmi les (i-1) déjà triés.
- 2) répéter cette action jusqu'à atteindre la fin de la liste.

L'action d'insertion se traduit par :

- utiliser une variable intermédiaire **temp** pour conserver la valeur à insérer.
- déplacer les éléments T[i-1], T[i-2]... vers la droite tant que leur valeur est supérieur à celle de **temp**.
- insérer l'élément **temp** à la bonne place.

Compléter le tableau suivant afin de trier ses éléments par la méthode du tri par insertion

							temp
	3	7	4	1	8	2	
<u>i</u> = 1							
<u>i</u> = 2							
<u>i</u> = 3							
<u>i</u> = 4							
<u>i</u> = 5							

```
1. def tri_insertion(t,n):
2.     for i in range(1,n):
3.         temp = t [i]
4.         j = i
5.         #décalage des éléments du tableau }
6.         while j>0 and t [j-1]>temp:
7.             t [j]=t [j-1]
8.             j = j-1
9.         #on insère l'élément à sa place
10.        t[j]=temp
```