

## Tri par valeurs croissantes

On initialise le processus en se positionnant sur la première case, on compare 2 avec -6, il doit y avoir permutation des valeurs : ...

2	-6	4	-1	8
---	----	---	----	---

Permuter

... On obtient donc le tableau suivant : Puis on compare -6 avec 4, -1 et 8 ...

-6	2	4	-1	8
----	---	---	----	---

... Il n'y a plus de permutation possible avec le reste des éléments du tableau

On passe donc à la deuxième case : On compare 2 avec 4 pas de permutation possible, puis 2 avec -1 donc permutation :

-6	2	4	-1	8
----	---	---	----	---

Permuter

-6	-1	4	2	8
----	----	---	---	---

Il n'y a plus de permutation avec le reste des éléments du tableau

On passe donc à la troisième case : On compare 4 avec 2, il doit y avoir permutation ...

-6	-1	4	2	8
----	----	---	---	---

Permuter

-6	-1	2	4	8
----	----	---	---	---

... puis on compare 2 avec 4 et puis 2 avec 8, il n'y a pas de permutation.

-6	-1	2	4	8
----	----	---	---	---

... puis on compare 4 avec 8, il n'y a pas de permutation envisageable.

On arrive sur le dernier élément du tableau et il n'y a plus de comparaison envisageable, le tableau est trié !!!

## Tri par valeurs croissantes

Je vous demande donc d'interpréter ce processus et de le traduire en code pour AlgoBox, il suffit pour cela d'imbriquer une boucle dans une autre boucle. Il faudra tester à chaque fois les éléments à comparer, s'il doit y avoir une permutation on le fait en introduisant une troisième variable comme suit :

```
Temp PREND_LA_VALEUR Tab[i]
Tab[i] PREND_LA_VALEUR Tab[j]
Tab[j] PREND_LA_VALEUR Temp
```

Bon courage à vous !!!