

.1 Révisions

1. **Champion borné** Écrire un programme qui recherche dans un tableau la plus petite valeur comprises entre deux bornes `a` et `b` données par l'utilisateur.
2. On suppose que l'on dispose d'un tableau de chaînes de caractères (de longueur maximale 20 caractères : `string[20]`), de dimension N , correspondant à un ensemble de mots.

```
var Tm = array[1..N] of string[20]
```

Ce tableau n'est pas trié. On veut produire un tableau trié sur l'envers des mots (on considère chaque mot « à l'envers », de la dernière lettre à la première)¹.

- (a) Écrire une fonction `avant` qui prend en argument deux chaînes de caractère, et répond `vrai` si la première est avant la seconde, dans le sens défini précédemment (c'est-à-dire en prenant les mots à l'envers).

```
function avant(x, y : string[20]) : boolean;
```

- (b) Proposer un algorithme qui réalise le tri du tableau initial, en utilisant la fonction `avant`.
 - (c) Discuter et justifier l'algorithme que vous avez choisi.
3. On considère un tableau `tab` de n entiers.
 - Écrivez une fonction prenant en argument un tel tableau et qui rend les deux plus grands entiers contenus dans `tab`
 - Généralisez votre fonction pour qu'elle rende les m plus grands entiers, triés dans un tableau
 - Quelle est la complexité de cette dernière fonction ? Un tri préalable de `tab` par une des méthodes efficaces vues en cours aurait-il été préférable à la solution à laquelle vous êtes arrivé ?
 4. On considère deux tableaux A et B de n éléments chacun. On dit que A est *contenu* dans B si tous les éléments de A figurent au moins une fois dans B .
 - Écrire une fonction qui, sans trier les tableaux, dit si A est contenu dans B . Quelle est la complexité de l'algorithme ?
 - Améliore-t-on la complexité si on trie l'un des deux tableaux et si oui, préciser lequel. Améliore-t-on à nouveau la complexité si l'on trie les deux tableaux ?

¹On utilise fréquemment ce genre de technique pour travailler sur les suffixes.