

## TP de Sys ° 1 : Révis

Page de TD et TP :

### Exercice 1 : structure et tableaux

1. Créez une structure `permutation` composée d'un champ `tableau` d'entiers correspondant à la permutation et d'un champ `entier` correspondant à sa longueur. On considère un tableau statique de longueur `N`, où `N` est une constante prédéfinie.
2. Ecrivez une fonction `void applique_permutation(permutation p, int *t)` qui applique une permutation `p` à un tableau d'entiers `t` de même longueur que la permutation. Par exemple la permutation 31420 appliquée au tableau {2, 4, 3, 1, 7} donne {7, 4, 1, 2, 3}.
3. Ecrivez une fonction `void tri(int *t, int lg)` qui trie un tableau en calculant la permutation qui permet de trier le tableau et en l'appliquant ensuite. Le  $i^{\text{ème}}$  élément de la permutation est égal au nombre d'éléments du tableau plus petits que le  $i^{\text{ème}}$  élément du tableau.

### Exercice 2 : makefile et chaînes de caractères

1. Ecrivez une fonction qui retourne le nombre d'apparitions d'un mot dans un texte. On pourra considérer que le mot est composé uniquement de lettres de latin non accentuées.
2. Créez une structure `stat_mot` composée d'un champ `chaîne` de caractères représentant un mot et d'un champ `entier` représentant une fréquence.
3. Ecrivez une fonction qui retourne le mot d'une longueur minimale donnée, le plus fréquent d'un texte, ainsi que sa fréquence d'apparition. On pourra considérer que le mot n'exède pas une longueur `LONGUEURMAX` prédéfinie.
4. Créez un fichier `frequence.c` qui contiendra les deux fonctions précédentes, le fichier `frequence.h` afférent, un fichier `main.c` contenant la fonction `main` et le fichier `makefile` permettant la compilation et production de l'exécutable.
5. Ajoutez dans le fichier `frequence.c`, une fonction qui retourne les `N` premiers mots les plus fréquents d'un texte avec leur fréquence d'apparition.

### Exercice 3 : encore des chaînes de caractères On tautogramme un poème

dont chaque vers ne mentionne que des mots commençant pas la même lettre. Voici un exemple de tautogramme de deux vers : Les mots du premier vers commencent tous par "t", ceux du second commencent tous par "s".

triste, transi, tout terni, tout tremblant,  
sombre, songeant, sans sûre soutenance.

Écrire une fonction

```
int tautogramme(char *poeme)
```

renvoyant 1 si la chaîne stockée dans poeme est un tautogramme et 0 sinon. O  
cette chaîne les hypothèses suivantes :

- chaque mot est formé de caractères minuscules non accentués, *i.e.* de valeurs de char comprises entre 'a' et 'z'.
- les autres caractères, quels qu'ils soient, ne font partie d'aucun mot.
- chaque vers est séparé du suivant par un ou plusieurs retours chariot. La chaîne devra être lue qu'une fois au plus (aucun retour en arrière). La fonction devra renvoyer sa réponse le plus rapidement possible. Elle ne peut appeler aucune autre fonction (pas même strlen).

Attention aux cas limites : vers ne contenant aucun mot ou bien un seul mot, vers ne commençant pas par un mot, vers final non suivi d'un retour chariot, poème vide, etc..