

# Informatique – langage C

JB Kammerer

Examen pratique du mardi 6 décembre 2022

Durée : 2h

Tous documents autorisés.

## Exercice 1 : Statistiques (14 points)

1. Écrivez un programme nommé `exo1.c` permettant de faire des statistiques à propos d'une lettre dans un texte. On utilisera pour cela le texte « Lorem Ipsum » disponible sur Moodle.
  1. On utilisera une variable `char l` initialisée à la valeur `'s'`. Cette variable contient le code ASCII de la lettre objet des statistiques.
  2. Ouvrez un flux en lecture vers le fichier contenant le texte et, en lisant tous les caractères les un après les autres jusqu'à la fin du fichier, comptez le nombre d'occurrences `occ` (variable de type `unsigned int`) de la lettre dont le code ASCII est contenu dans la variable `l`. On ne fera pas la différence entre les majuscules et les minuscules. Il faudra donc convertir tous les caractères lus en minuscules (fonction `int tolower(int c)`). Affichez le nombre `occ` dans le terminal.
  3. Après avoir remis le curseur de lecture au début du fichier, relisez le fichier de manière à déterminer la distance maximale `D` entre deux occurrences de la lettre dont le code ASCII est contenu dans la variable `l`. On appelle « distance », le nombre de caractères séparant deux occurrences successives de la lettre analysée. Affichez le nombre `D` dans le terminal.
  4. Allouez un bloc de mémoire permettant de stocker `D+1` entiers de type `unsigned int`. Initialisez toutes les cases de cette table à zéro. On nommera cette table `dist`.
  5. Après avoir remis le curseur de lecture au début du fichier, relisez les données du fichier texte de manière à récolter des statistiques sur les distances entre occurrences successives de la lettre objet de l'analyse : à chaque fois que l'on trouvera une occurrence de la lettre dont le code ASCII est contenu dans `l`, après avoir déterminé la distance `d` la séparant de la précédente occurrence, on incrémentera `dist[d]`. Ainsi, une fois l'intégralité du fichier analysé, `dist[i]` contient le nombre de paires de caractères séparées d'une distance `i`.
  6. Recherchez le plus grand nombre `max_dist` contenu dans le tableau `dist` puis affichez dans le terminal un histogramme représentant les données contenues dans `dist`. Pour cela, on affichera `D+1` lignes dans le terminal. La première ligne doit afficher le nombre 0 suivi d'un nombre d'étoiles (\*) égal à `dist[0]*40/max_dist`. La deuxième ligne doit

afficher le nombre 1 suivi d'un nombre d'étoiles (\*) égal à `dist[1]×40/max_dist`. La  $i+1^{\text{ème}}$  ligne doit afficher le nombre `i` suivi d'un nombre d'étoiles (\*) égal à `dist[i]×40/max_dist`. Pour que tous les nombres soient bien alignés, on utilisera `%4u` dans la fonction `printf` pour afficher les nombres sur 4 cases.

## Exercice 2 : Rendre paramétrable (6 points)

2. Écrivez un programme nommé `exo2.c` faisant la même chose que le programme `exo1.c` mais où le nom du fichier à analyser ainsi que la lettre sur laquelle on cherche à extraire des statistiques sont passés en paramètres. Le premier paramètre est le nom du fichier et le second la lettre à rechercher pour faire les statistiques.
1. Si l'utilisateur ne passe pas le bon nombre de paramètres, un message rappelant la syntaxe attendue doit être affiché.
2. Si la chaîne passée en second paramètre contient plus d'une lettre (utilisez la fonction `size_t strlen(const char *str)` pour mesurer la longueur de cette chaîne). Un message rappelant la syntaxe attendue doit être affichée.
3. Si l'ouverture du fichier dont le nom est passé en premier paramètre échoue, un message d'erreur doit s'afficher.