Structures de données 1
Hachage ouvert

Le but du tp est d'établir la liste des mots d'un texte contenu dans un fichier, chaque mot étant associé à son nombre d'occurence.
Dans toute la suite on designe par mot une chaine de caractères lue grace au format `%s` de scanf.
Les mots seront rangés dans une table de listes de mots de taille MAX. On pourra tester différentes valeurs de MAX.
On utilisera donc les types:

```
typedef struct cel{
char *mot;
int occurence;
struct cel * suivant;
} CelluleMot, *ListeMot;

typedef ListeMot TableHachage[MAX];
```

1. Gestion de listes
   Ecrire les fonctions d'allocation, d'affichage et d'ajout ordonné dans une liste de CelluleMot. Pour l'ajout,si une cellule est déjà présente on augmente le nombre d'occurence, sinon l'ajout est fait de manîre ordonnée (avec `strcmp`).

2. Hachage
   Ecrire les fonctions de hachage suivantes pour des chaines de caractères:

   - Somme des codes ASCII des lettres
   - Somme des produits du code ASCII de la lettre par sa position dans le mot ($\Sigma_{i=0}^{Mot[i]!='\backslash 0'}(i+1)*Mot[i]$)
   - *il existe des fonctions plus efficaces pour les chaines de caractéres*

3. Ecrire une fonction permettant l'ajout dans une table de hachage (tableau de MAX Liste)

4. Ecrire une fonction de recherche d'un mot.

5. Ecrire une fonction qui affiche le nombre d'occurence d'un mot dans un fichier.

6. Ecrire une fonction permettant d'afficher le contenu des listes contenues dans la table de hachage (fonction non destructrice).

7. En utilisant la fusion de listes ordonnées , écrire un programme qui affiche la suite des mots d'un fichier et leur occurence. Cette fonction est destructrice, les listes de la table de hachage sont vides après l'appel

8. Ecrire un programme qui reçoit une référence de fichier en argument et contruit la table de hachage pout les mots du fichier puis affiche la suite ordonnée des mots et de leur nombre d'occurence (un mot par ligne).