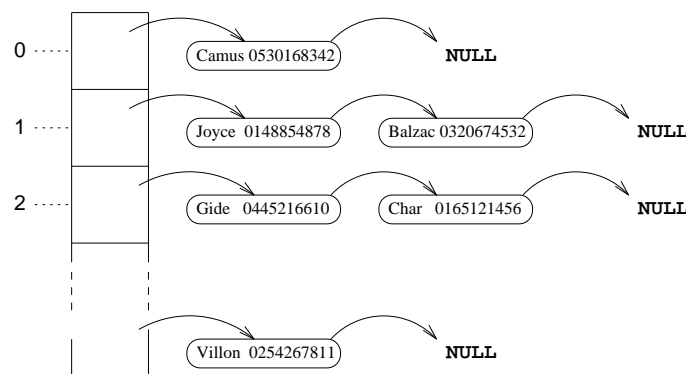


## Table de hachage

Une table de hachage est une généralisation de la notion de tableau ordinaire, destinée à stocker des informations indexées par des clés. Prenons comme exemple un annuaire téléphonique, où les numéros de téléphone sont indexés par les noms des abonnés. La représentation doit être optimisée pour faciliter les opérations élémentaires d'insertion, suppression, et recherche d'un abonné. L'idéal serait donc disposer d'une structure qui combinerait la rapidité d'accès d'un tableau (pour la recherche) et la souplesse d'une liste chaînée (pour l'insertion et la suppression).

Le principe de construction de la table de hachage est le suivant : on répartit les données en fonction de leur clé entre plusieurs listes chaînées de taille restreinte, ces listes chaînes étant ensuite regroupées dans un tableau à une dimension. La répartition en sous-listes se fait avec la définition d'une *fonction de hachage*, qui associe à chaque clé (chaîne de caractères, ici) un indice dans le tableau.



## Implémentation

La spécification d'un paquetage pour une table de hachage, ainsi que l'esquisse du corps de ce paquetage sont disponibles à

```

~touzet/Algo/Hachage/paq_hachage.ads
~touzet/Algo/Hachage/paq_hachage.adb
  
```

Le fichier `paq_hachage.adb` contient déjà l'implémentation de la procédure `Affichage`. Vous pouvez donc l'utiliser pour tester vos procédures.

**Question 1.** Écrire une fonction `Hachage`, qui prend en argument une chaîne de caractères  $s$  et renvoie un entier compris entre 0 et 9. Pour cela, on peut choisir la somme des positions des lettres de  $s$  modulo la taille de la table.

**Question 2.** Écrire une procédure `Inserer`, qui insère un nouvel abonné (nom et numéro de téléphone) dans l'annuaire.

**Question 3.** Écrire une procédure `Rechercher`, qui affiche le numéro de téléphone d'un abonné. Si plusieurs abonnés portent le même nom, la procédure affiche les différents numéros.

**Question 4.** Écrire une procédure `Supprimer`, qui retire un abonné de l'annuaire.

**Question 5.** Écrire une procédure `Rechercher_Inverse`, qui fait l'opération inverse : trouver le nom d'un abonné à partir de son numéro de téléphone. Que pensez-vous du temps d'accès dans ce cas ?

On veut maintenant interfacier la table de hachage avec un fichier texte "`annuaire`", contenant l'annuaire.

**Question 6.** Écrire une fonction `Lire(t:in out Table)` qui charge le contenu du fichier "annuaire", supposé existant, dans une table de hachage. L'ouverture du fichier se fait avec `Open(F, In_File, "annuaire")`, où `F` est une variable de type `File_Type`. La lecture utilise la commande `Get: Get(F, mot)`, par exemple. Le test de fin de fichier se fait avec `End_of_File(F)`. On ferme ensuite le fichier avec `Close(F)`.

Vous pouvez tester votre procédure avec le fichier `annuaire` fourni dans le répertoire habituel.

**Question 7.** Écrire une fonction `Ecrire(T:Table)`, qui sauvegarde le contenu de la table dans un fichier texte. La création du nouveau fichier "annuaire" se fait avec `Create(F, Out_File, "annuaire")`. Utiliser ensuite `Put(F, mot)` pour écrire dans `F`, `New_Line(F)` pour passer de ligne et `Close(F)`.