Algorithmique 1
Feuille 9 : Tables de hachage

Exercice 9.1 *Construction*
On considère un ensemble de clés données directement sous forme numérique, que l'on veut stocker dans une table de hachage de taille $m = 11$.
Soit la fonction de hachage : $h(x) = x \ mod \ m$.

1. Décrire les structures de données et la gestion des collisions pour la séquence d'ajouts : 10, 22, 31, 4, 15, 28, 83, 88, 59, 37, dans les cas suivants :
   (a) Adressage chaîné ;
   (b) Adressage ouvert et traitement des collisions par placement dans la première case libre (modulo m) à partir de la case calculée ;
   (c) Adressage ouvert avec, pour rechercher une nouvelle case libre en cas d'occupation, un pas donné par une seconde fonction $h'(x) = 1 + x \ mod \ (m - 1)$.
2. Décrire ce qui se passe dans les trois cas si on poursuit par les 2 opérations suivantes, suppression de la clé 83 et recherche de la clé 59.

Exercice 9.2 *Extension*
1. Rappeler dans quel cas il est utile de redimensionner une table. Quelles sont les méthodes utilisées ?
2. On considère la table de l'exercice précédent. Pour chaque cas d'adressage indiquer quelle(s) méthode(s) de redimensionnement peut être utilisée(s).
3. Chaque fois que c'est possible appliquer la méthode.

Exercice 9.3 *Adressage ouvert*
Soit une table de symboles `S` qui contient les mots clés `"if"`, `"then"`, `"else"`, `"begin"` et `"end"`. On souhaite construire une table de hachage `T` de taille `m=11` en utilisant l'adressage ouvert [1] pour stocker ces mots clés.
On utilisera la fonction de hachage `h(x)`,

$$h(x) = (asc(x_1) * 2^{l-1} + asc(x_2) * 2^{l-2} + ... + asc(x_l)) \ \ mod \ \ m$$

$$x = "x_1 x_2 \dots x_l"$$

Proposer une solution aux problèmes de collisions.

Problème récurrent (notre fil d'ariane)
Exercice 9.4 Gestion d'une piste d'atterrissage des avions

Un avion est un enregistrement contenant :
– l'indicatif (6 caractères)
– la destination (30 caractères)
– l'autonomie résiduelle de carburant comptée en heures de vol (entier)
– deux booléens indiquant s'il y a un pirate à bord et s'il y a le feu.

1. *Pour stocker les avions on souhaite construire une table de hachage* `Piste` *de taille* `m=19` *en utilisant l'adressage ouvert. Représenter cette table en précisant quelles sont les clés et la méthode de hachage choisie. Indiquer où est concrètement stocké un avion*
2. Définir les structures de données nécessaires.

1. En adressage ouvert tous les éléments sont conservés dans la table de hachage elle-même.

3. Ecrire la fonction `Priorité` ainsi que la gestion complète de la piste.
4. Envisager le cas de suppression d'un élément quelconque de la file lorsque le pirate a mis sa menace de détournement à exécution.

Quelles sont les notions que vous venez de voir qui peuvent permettre d'amorcer le fil d'ariane ?

Annexe A : **Type abstrait tableHash**

```
fonction creerTableHachage(ref T: tableHash de cle, ref h: fonction): vide;
fonction chercher(ref T: tableHash de cle, val v: cle): curseur;
ajouter(ref T: tableHash de cle, val v: cle): booleen;
supprimer(ref T: tableHash de cle, val v: cle): vide;
```

Annexe B : **Adressage chaîné**

```
tableHash de cle= structure
                    table: tableau[0..m-1] de listeDC de cle;
                    h: fonction(val v: cle): curseur;
                  finstructure
```

Annexe C : **Adressage ouvert**

```
tableHash de cle= structure
                    table: tableau[0..m-1] de cle;
                    h: fonction(val v: cle): curseur;
                    s: fonction(val v: cle, i: entier): curseur;
                  finstructure
```