

Perl TD3 : Tableaux associatifs (ou hachages)

Résumé sur les tableaux associatifs

Les tableaux associatifs (ou hachages), comme les tableaux classiques, sont une collection de données, mais les indices repérant les éléments ne sont pas forcément des entiers, mais des valeurs quelconques que l'on nomme clés. L'ordre des couples (clé/élément) dans un hachage est imprévisible. Il ne dépend que de la manière dont Perl les stocke pour faciliter sa recherche en fonction des clés. De plus, l'accès à un hash inexistant implique sa création. Le nom d'une variable de type hachage commence par % (Rq : les variables %hash, @hash et \$hash n'ont rien à voir entre elles).

```
# Assigner un element (crée le hachage s'il n'existe pas)
my %hash;
$hach{"aaa"} = 3; # cle = "aaa", valeur = 3
$hach{5.6} = "abc"; # cle = "5.6", valeur = "abc"
# Initialisation globale ou copie de hachage
%hach = ("aaa",3,5.6,"abc"); # idem que ci-dessus
#ou encore
%hach = (aaa => 3, 5.6 => abc); # idem que ci-dessus
%cop = %hach; # recopie de hachage
```

Fonction **keys** : fournit la liste de toutes les clés d'un hachage. Utilisée avec une boucle foreach, elle permet de parcourir tous les éléments du hachage (sans bien sûr maîtriser l'ordre du parcours) :

```
@liste = keys(%hach); # @liste = ("aaa", 5.6) ou (5.6, "aaa")
for $cle (keys(%hach)) {
    print "cle: $cle, valeur: $hach{$cle}\n";
}
```

Fonction **values** : fournit la liste de tous les éléments d'un hachage, dans le même ordre que la liste des clés renvoyées par keys :

```
@liste = values(%hach); # @liste = (3,"abc") ou ("abc",3)
```

Fonction **each** : fournit un couple (clé/valeur) sous forme d'une liste à 2 éléments. A chaque application sur le même hachage, le couple suivant est renvoyé (liste vide si c'est fini) :

```
while ( ($cle,$val)=each(%hach) ) {
    print "cle: $cle, valeur: $val\n";
}
```

Fonction **delete** : permet de supprimer un élément d'un hachage :

```
delete $hach{"aaa"}; # %hach ne contient plus le couple cle/element: "aaa"/3
```

Exercice 1

La table suivante spécifie la maladie associée à chaque gène :

AD4	Alzheimer's Disease
BRCA1	Breast Cancer
DMD	Duchenne Muscular Dystrophy
FMR1	Fragile X Syndrome
GBA	Gaucher's Disease

Ecrivez un script Perl qui demande à l'utilisateur d'entrer un nom de gène, et affiche la maladie correspondante. Le programme devra afficher un message d'erreur si le gène entré n'est pas trouvé dans la table.

Remarque : pour définir la table de hachage, vous pouvez utiliser ces 2 syntaxes :

```
%genes = ("AD4", "Alzheimer's disease", "BRCA1",  
          "Breast Cancer",...);
```

OU

```
%genes = (AD4 => "Alzheimer's disease",  
          BRCA1 => Breast Cancer,  
          ...);
```

Exercice 2

Reprenez le fichier **kinases_map.txt** du TD précédent.

Ecrivez un script Perl qui construit une table de hachage dont les couples clés/valeurs sont les données 'Accession number in OMIM'/'Gene name' trouvés dans le fichier, avec un numéro strictement inférieur à 600000.

Exercice 3

Considérez un fichier contenant un texte quelconque.

1. Ecrivez un script Perl qui compte le nombre total d'occurrences de chaque mot du texte. Si le texte est "les coccinelles dévorent les pucerons", vous devez obtenir le résultat :

```
2  les  
1  coccinelles  
1  dévorent  
1  pucerons
```

indice : expérimentez l'opérateur `each`.

2. Affichez également le nombre total de mots du texte, ainsi que le nombre de mots différents (i.e. chaque mot doit être compté une seule fois).
Pour l'exemple précédent :

```
Nb total mots: 5  
Nb mots differents: 4
```

indice : expérimentez les opérateurs `keys` et `values`.

3. Trouvez une autre solution qui n'utilise pas l'opérateur `values`.

Exercice 4

Ecrivez un script Perl qui permet de saisir deux chaînes : Nom et Prénom. Lorsque l'utilisateur soumet ses données, le script consulte une table de hachage dont les clés sont la concaténation "nom-prénom".

- Si la table ne contient pas les données saisies, le script demandera un numéro de téléphone et ajoutera dans la table la clé "nom-prénom" avec comme valeur le numéro de téléphone donné.
- Si la table contient les données saisies, le script affichera le numéro de téléphone correspondant.