

# TP - structure, enumeration et boucle

## Objectif(s)

- ★ Utilisation des énumérations en C.
- ★ Utilisation des structures en C.
- ★ Et quelques boucles pour ne pas oublier.

## Exercices obligatoires

### Exercice 1 – Les énumérations

#### Question 1

Écrivez une énumération `enum jour` qui permettra de définir les jours de la semaine. Rappelez vous d'utiliser les règles de nommage en C avec des noms d'éléments tout en majuscule.

#### Question 2

Écrivez une fonction `affiche_jour` qui prend en paramètre un élément `enum jour` et qui affiche le nom du jour correspondant (c'est à dire "lundi" et non 1).

#### Question 3

Reprendre les questions du TP sur la structure des dates (partie obligatoire) et remplacer le mois par une énumération.

### Exercice 2 – Le temps qui passe (avec des structures)

#### Question 1

Définissez une structure `struct horaire` qui permet de modéliser un horaire avec les champs heure, minute et seconde.

#### Question 2

Écrivez une fonction `affiche_horaire` qui prend en paramètre une `struct horaire` et qui affiche l'horaire à l'écran.

#### Question 3

Écrivez une fonction `seconde_vers_horaire` qui convertit un temps écoulé exprimé en seconde depuis minuit en heure, minute seconde. La fonction prend en entrée 2 paramètres :

**sjournee** : nombre de secondes depuis le début de la journée (entier compris en 0 et 86400)

**p\_hor** : une référence vers une structure horaire à modifier

exemple : Soit 47531s, le temps écoulé en seconde depuis le début de la journée, il correspond à 13h12min et 11s.

Ecrivez une fonction `main` permettant de tester votre fonction, en appelant `affiche_horaire` dans le `main`.

**indice** : la fonction modulo (%) pourrait vous aider.

#### Question 4

Écrivez une fonction `saisir_horaire` qui demande à l'utilisateur de saisir les heure, minute, seconde et qui renvoie une structure horaire qui contient des 3 valeurs.

Testez votre fonction saisissant un horaire au clavier et en l'affichant dans la fonction `main`.

#### Question 5

Écrivez une fonction `une_seconde_plus_tard` qui prend en entrée une structure horaire exprimé et qui calcule et renvoie l'horaire une seconde plus tard. La fonction ne fera pas d'affichage.

Modifiez la fonction `main` pour tester votre fonction. La fonction `main` demandera à l'utilisateur de saisir un horaire et affichera l'horaire une seconde plus tard. On supposera que la saisie des heures, minutes, secondes sera correcte.

#### Question 6

Écrivez une fonction qui fait le processus inverse de la question 3, à savoir prend les trois paramètres heure, minute et seconde et calcule le nombre de secondes que cela représente. Modifiez la fonction `main` pour tester votre fonction.

#### Question 7

Écrivez une fonction `duree` qui prend en paramètres deux horaire et qui calculent le temps qui sépare les horaires 1 et 2 (en prenant éventuellement en compte un jour d'écart si le deuxième horaire est plus tôt que le premier). Le temps qui les sépare sera renvoyer par un `struct` horaire.

## Exercice 3 – Table de multiplication

#### Question 1

Écrivez une fonction qui affiche la table de multiplication des nombres entre 1 et 10.

Il peut être intéressant d'utiliser les formats dans les `printf`. Pour rappel `printf("%4d", i);` affiche la valeur de `i` sur 4 colonnes en alignant la valeur à droite.

## Renforcements

## Exercice 4

Extrait de wikipédia :

*En arithmétique récréative, on appelle « nombres McNugget » une application particulière du problème des pièces de monnaie — aussi appelé problème diophantien linéaire de Frobenius — aux boîtes McNugget : Les nombres McNugget sont les nombres totaux de nuggets que peut contenir un ensemble varié de boîtes McNugget (chacune pouvant contenir soit 6, soit 9 soit 20 nuggets). Par exemple, 41 est un nombre McNugget car il est égal à  $6 + 6 + 9 + 20$ . Tous les nombres entiers à partir de 44 sont des nombres McNugget, tandis que parmi les 44 nombres de 0 à 43, seulement la moitié font partie de cette suite d'entiers (suite A214777 de l'OEIS).*

Le problème consiste donc à rechercher les valeurs  $i, j$  et  $k$  tel que le nombre  $n$  s'exprime sous la forme  $6i + 9j + 20k$  avec  $i, j$  et  $k$  des nombres entiers positifs.

#### Question 1

Écrivez une fonction qui retourne `true` si un nombre est un nombre McNugget sinon `false` en utilisant des boucles imbriquées. La fonction affichera le premier découpage qu'elle trouve.

#### Question 2

Affichez tous les nombres qui ne sont pas des nombres McNugget entre 1 et 1 000 000.

#### Question 3

Écrivez une fonction `nb_decoupage` qui prend en paramètre un nombre et compte le nombre de découpages différents en nombre de McNugget du paramètre.

#### Question 4

Écrivez une fonction `nb_avec_k_decoupages` qui prend en paramètre un nombre  $k$  et compte les nombres entre 1 et 1000 qui ont exactement  $k$  découpage en McNugget.

#### Question 5

Quel est le nombre de découpage le plus fréquent des nombres entre 1 et 1000 ?