

TP 3 : FONCTIONS, HEADERS, COMPILATION SÉPARÉE ET PROGRAMMATION RÉCURSIVE

TABLE DES MATIÈRES

| | |
|---|---|
| But | 1 |
| Exercice 1 : fonctions et compilation séparée | 1 |
| Exercice 2 : fonctions récursives | 2 |

BUT

Vous devez maîtriser à la fin de cette séance les notions suivantes : L'utilisation des fonctions, créer des **headers** - **.h**, utiliser plusieurs fichiers **.c** et créer des fonctions récursives

EXERCICE 1 : FONCTIONS ET COMPILATION SÉPARÉE

Reprendre l'exercice 2 du TP 1, avec l'utilisation de fonctions, appelés dans un **main**.

- (1) Écrire une fonction `int date_vers_secondes(int jours, int heures, int minutes, int secondes)`, qui converti un temps composé d'un nombre entier de jours, d'heures, de minutes et de secondes, en un temps en seconde.
- (2) Écrire une fonction `void affiche_secondes_en_date(int seconde)`, qui affiche sur la console en jours, heures, minutes, secondes un temps donné en seconde. seconde.
- (3) Écrire une fonction `float celcius_vers_fahrenheit(float celcius)` qui fait la conversion degré celcius vers degré fahrenheit
- (4) Écrire une fonction `float fahrenheit_vers_celcius(float fahrenheit)` qui fait la conversion degré fahrenheit vers degré celcius
- (5) Écrire un **main** qui permet d'appeler les diverses fonctions précédentes.
- (6) Écrire un fichier **.h** associé aux fonctions, ainsi qu'un **.c** contenant le code de ces fonctions sans le **main**. Créer le fichier **main.c** ne comportant que la fonction **main**.

EXERCICE 2 : FONCTIONS RÉCURSIVES

On programmera les fonctions dans un fichier `.c` dédié, auquel on associera son header. La fonction `main` sera programmer dans un fichier `main.c`.

- (1) Écrire une fonction récursive qui calcule $n!$
- (2) Écrire une fonction récursive qui calcule x^n . On utilisera la remarque suivante :

$$x^n = \begin{cases} 1 & \text{si } n = 0 \\ x.x^n & \text{si } n \text{ est impair} \\ (x^2)^{n/2} & \text{si } n \text{ est pair} \end{cases}$$

- (3) Écrire une fonction récursive qui calcule le nombre de combinaison de p parmi n : $\binom{n}{p}$, en utilisant la formule suivante :

$$\binom{n}{p} = \begin{cases} 1 & \text{si } p = 0 \text{ ou } p = n \\ \binom{n-1}{p-1} + \binom{n}{p-1} & \text{si } 0 < p < n \\ 0 & \text{sinon} \end{cases}$$

- (4) Écrire un `main` permettant d'appeler les fonctions précédentes.