

Programmation en C

Atelier 02 : Les fonctions

Pour mener à bien cet atelier, chaque étudiant devra créer un projet par exercice en suivant la notation : *Atl02_ExoX_Prenom_Nom*.

Le *X* doit être remplacé le numéro correspondant de l'exercice. *Prenom* et *Nom* doit être remplacé par votre prenom et nom sans les accents.

Exercice 1

Essayez de coder une minicalculatrice qui :

- Dit bonjour ;
- Demande deux nombres entiers à l'utilisateur ;
- Les additionne, soustrait, multiplie et les divise (avec un arrondi au millième) ;
- Dit au revoir.

Au final, ce pourrait donner :

```
Bonjour !  
  
Veuillez saisir le premier nombre : 4  
Veuillez saisir le deuxième nombre : 7  
  
Calculs :  
  
    4 + 7 = 11  
    4 - 7 = -3  
    4 * 7 = 28  
    4 / 7 = 0.571  
  
Au revoir !
```

Exercice 2

Imaginez que vous ayez un score de jeu vidéo sous la main :

- Si le score est inférieur à 2000, afficher « *C'est la catastrophe* »
- Si le score est supérieur ou égal à 2000 et que le score est inférieur à 5000, afficher « *Tu peux mieux faire* »
- Si le score est supérieur ou égal à 5000 et que le score est inférieur à 9000, afficher « *Tu es sur la bonne voie* »
- Sinon, afficher « *Tu es le meilleur !* »

À vous de codez ça.

Exercice 3 – L'opérateur ternaire

L'opérateur ternaire, qui est une autre façon de faire un test de condition, tient son nom du fait qu'il est le seul à avoir trois opérandes. En effet, il se compose comme suit :

`(condition) ? instruction si vrai : instruction si faux`

Les parenthèses ne sont pas obligatoires. Ce qu'il y a à retenir, c'est qu'il se comporte comme un `if/else`, tout en étant plus condensé et plus rapide à écrire. Voyez par vous-mêmes :

```
#include <stdio.h>

int main(void)
{
    int heure;

    scanf("%d", &heure);

    (heure > 8 && heure < 20) ? printf("Il fait jour.") : printf("Il
fait nuit.");

    return 0;
}
```

Pour bien comprendre cette nouvelle notion, nous allons faire un petit exercice. Imaginez qu'on veuille faire un mini jeu vidéo dans lequel on affiche le nombre de coups du joueur. Seulement voilà, vous êtes maniaques du français et vous ne supportez pas qu'il y ait un 's' en trop ou en moins.

Essayez de coder un programme dans lequel on demande à l'utilisateur le nombre de coups puis on affiche le résultat.

Exercice 4

Ecrire en C une fonction permettant d’afficher un rectangle d’étoile “*” suivant les dimensions saisies par l’utilisateur. Ces dimensions doivent être envoyées en paramètres à la fonction.

Exercice 5

Afficher un triangle formé d’étoiles. La hauteur du triangle (c’est-à-dire le nombre de lignes) sera fournie en donnée, comme dans l’exemple ci-dessous.

```
Donnez un nombre :  
5  
  
*  
**  
***  
****  
*****
```

Exercice 6

Écrire une fonction qui convertit les kilomètres en miles (1 mile = 1,609 km). Ecrire dans le main le code principal qui permet de tester la fonction.

Exercice 7

Écrire une fonction qui convertit les degrés Fahrenheit en degrés centigrades. Ecrire dans le main le code principal qui permet de tester la fonction.

$$\text{Formule : } \theta_C = \frac{5}{9}(\theta_F - 32)$$

Exercice 8

Écrire une fonction qui calcule le volume d’une sphère. Ecrire dans le main le code principal qui permet de tester la fonction.

Exercice 9

Écrire une fonction ayant en paramètres le nombre d’heures effectuées par un salarié dans la semaine et son salaire horaire, qui retourne sa paye hebdomadaire.

On prendra en compte les heures supplémentaires (au-delà de 35 heures) payées à 150%.

Ecrire dans le main le code principal qui permet de tester la fonction.

Exercice 10

Ecrire une fonction qui prend en entrée les coefficients d'une équation du second degré et affiche les racines réelles s'il y en a. Ecrire dans le main le code principal qui permet de tester la fonction.

Exercice 11

Soit la fonction Moyenne qui retourne la moyenne de trois réels

- Ecrire le programme de la fonction
- Ecrire un programme principal de test qui contient sa déclaration ainsi que les instructions l'appelant

Exercice 12

Ecrire un programme déclarant une fonction calculant le volume d'un cône de révolution et affiche le résultat de cette fonction pour :

- Rayon = 12 et Hauteur = 8
- Rayon = 5 et Hauteur = 7

Exercice 13

Ecrire en C un programme comportant :

1. La déclaration de trois variables globales entières *iHeures*, *iMinutes*, *iSecondes*.
2. Un procédure *affiche_heure* qui imprimera le message suivant :
Il est ... heure(s) ... minute(s) ... seconde(s)
En respectant l'orthographe du singulier et du pluriel.
3. Une procédure *saisir_heure* qui admettra trois paramètres entiers *iH*, *iM* et *iS*, dont elle affectera les valeurs respectivement à *iHeures*, *iMinutes* et *iSecondes*.
4. Un procédure *tick* qui incrémentera l'heure d'une seconde.
5. La procédure *main* sera un jeu d'essais des procédures précédentes.

Exercice 14

Écrire un programme qui détermine la *n* ième valeur *Un* (*n* étant fourni en donnée) de la «suite de Fibonacci» définie comme suit :

```
u1 = 1
u2 = 1
un = un-1 + un-2    pour n>2
```

Exercice 15

Écrire une fonction qui retourne le plus grand commun diviseur (pgcd) de deux nombres entiers positifs a et b .

L'algorithme d'Euclide est basé sur le principe suivant : On suppose que a est supérieur à b . On commence par affecter la valeur de a à b , ensuite, on attribue la valeur de b à r (le reste). Puis, calculer le reste r de la division de a par b ; et on recommence toutes ces étapes jusqu'à ce que le reste soit nul. On a alors trouvé le résultat : c'est le b qui a été obtenu à la fin de ce processus.