

Exercices sur les fonctions

Exercice 1 : une addition

Écrivez une fonction `addition` qui prend deux entiers pour arguments et qui renvoie la somme des deux entiers.

Exercice sur les conditions

Exercice 2 : minimum entre deux valeurs

Écrivez la fonction `minimum` qui prend deux entiers pour arguments et qui renvoie la valeur du plus petit des deux.

Exercice 2bis : maximum entre deux valeurs

Écrivez la fonction `maximum` qui prend deux entiers pour arguments et qui renvoie la valeur du plus grand des deux.

Exercice 3 : signe d'une somme

Écrivez la fonction `signe_somme` qui prend deux entiers pour arguments et qui renvoie `1` si la somme des deux entiers est positive, `0` sinon.

Exercice 4 : nombre dans un intervalle

Écrivez la fonction `est_dans_intervalle` qui prend trois entiers comme arguments `nombre`, `i1` et `i2`, et qui renvoie `1` si `nombre` est dans l'intervalle $[i1, i2]$, `0` sinon.

Attention

La notation $i < j < k$ n'est pas équivalent à `i < j < k` dans la majorité des langages informatiques ! Ici vous aurez besoin d'utiliser l'opérateur `&&` quelque part...

Exercice 5 : maximum parmi trois valeurs

Écrivez une fonction `maximum_3_val` qui prend trois entiers et qui renvoie le plus grand.

Exercice 6 : signe somme version 2

Écrivez une fonction `signe_somme_v2` qui prend deux entiers, **sans calculer la somme**, et qui renvoie `0` si la somme est nulle, `-1` si elle est négative, et `1` sinon.

Exercice 7 : est-ce que j'ai le droit à une réduction ?

Un musée présente les tarifs suivants selon le profil :

- tarif normal : 20,5€ ;
- étudiant.e plus de 26 ans : 20€ ;
- étudiant.e moins de 26 ans : 10,8€ ;
- non-étudiant.e moins de 26 ans : 15,5€ ;
- tarif moins de 15 ans : 8,2€.

Les tarifs dépendent aussi du jour de la semaine :

- les weekends, tous les tarifs sont réduits de 2€ ;
- les jeudis, tout le monde paye au tarif moins de 15 ans.

Écrivez une fonction `tarif_musee` qui prend 3 arguments :

- un entier `age` ;
- un entier `est_etudiant` qui vaut `1` si la personne est étudiante, `0` sinon ;
- un entier `jour` qui vaut `1` si c'est un lundi, `2` si c'est un mardi,..., `7` si c'est un dimanche

et qui renvoie le prix d'un ticket d'entrée pour le musée selon les arguments

Rappel

Pour définir les constantes pour les tarifs en C, on va utiliser la directive `#define` comme ceci :

```
#define TARIF_NORMAL 20.5
#define TARIF_ETU_MOINS26 10.8 // étudiant moins de 26
#define TARIF_ETU_PLUS26 20 // étudiant plus de 26
#define TARIF_MOINS26 15.5 // non étudiant
#define TARIF_MOINS15 8.2
```

Exercices sur les boucles

Exercice 8 : somme des entiers jusqu'à `n`

Écrivez une fonction `somme_entiers` qui prend en argument un entier `n`, et qui fait la somme de tous les entiers jusqu'à `n`.

Par exemple si `n=5`, la fonction doit renvoyer 15, c'est-à-dire $(1 + 2 + 3 + 4 + 5)$.

Exercice 9 : dessins d'étoiles

Écrivez une fonction `dessiner_etoiles`, qui ne prend aucun argument, et qui ne renvoie aucune valeur, et qui dessine :

```
*
**
***
****
*****
*****
*****
*****
*****
*****
*****
```

Rappel

Pour afficher du texte sur la console on utilise la fonction `printf`.

Pour revenir à la ligne, on utilise le caractère `\n`.

Pour passer une variable dans la chaîne de caractère en premier argument de la fonction, on écrit `%d` si la variable est un entier, `%s` si c'est une chaîne de caractère... Par exemple :

```
char* texte = "Je suis du texte";
int a = 2;
// affiche Je suis du texte 2.
// puis retourne à la ligne
printf("%s %d.\n", texte, a);
```

Exercice 10 : décomposer une somme

Écrivez une fonction `decompose_pieces_billet` qui prend un entier et qui affiche le nombre de billets de 5€, et de pièces de 2€ et 1€, qu'il faut pour faire cette somme.

On veut qu'il y ait le plus possible de billets de 5€. Quand ce n'est plus possible, on fait pareil avec les pièces de 2€ jusqu'à que ça ne soit plus possible, et on complète avec des pièces de 1€.

La fonction ne renvoie rien, mais elle affiche le résultat avec ce `printf` :

```
printf("%d=%d*5 euros + %d*2 euros + %d*1 euro\n",somme, nb_5, nb_2, nb_1);
```

Rappel

L'opération de division `/` entre deux entiers donne un résultat entier (par ex `5/2` renvoie 2) et pour récupérer le reste d'une division, on utilise l'opérateur `%` (par ex `5%2` renvoie 1).