

# TP - Premières fonctions, conditionnelles

## Objectif(s)

- ★ Programmer quelques fonctions simples en C.
- ★ Utiliser les conditions, les expressions logiques et l'imbrication de `if`
- ★ **Trouver les valeurs à utiliser pour tester vos fonctions. Pour chaque exercice, il est demandé de générer un jeu de test pour tester toutes les cas possibles. Ce n'est pas toujours facile, il faut bien réfléchir.**

## Exercices obligatoires

### Exercice 1 – Fonctions simples

#### Question 1

Écrivez une fonction `au_carre` qui prend en argument une valeur (`int`)  $x$  et retourne la valeur de  $x^2$ . Écrivez la fonction `main` obligatoire à un programme pour qu'elle affiche la valeur de  $72^2$  en utilisant la fonction précédente.

#### Question 2

Écrivez une fonction `pair` qui prend en argument une valeur (`int`)  $x$  et retourne la plus grande valeur (`int`) paire inférieure ou égale à  $x$ .

#### Question 3

Écrivez une fonction `test_pair()` qui vérifie que la fonction renvoie la bonne valeur pour des valeurs en exemple. La fonction affichera une alerte dans le cas où le test ne passe pas, ne fera rien sinon. Parmi les valeurs intéressantes à utiliser pour tester votre fonction `pair`, pensez bien à mettre aussi des tests sur les valeurs négatives et sur 0.

Écrivez aussi la fonction `main` qui appelle votre fonction de test<sup>1</sup>.

#### Question 4

Écrivez une fonction `difference` qui prend en argument deux valeurs (`double`)  $a$  et  $b$  et retourne la différence entre les deux, qui doit toujours être positive.

#### Question 5

Écrivez une fonction `test_difference` qui teste votre fonction sur différentes entrées. Pensez bien à tous les cas de test. Comme précédemment, la fonction restera muette si elle ne constate pas d'erreur, affichera un message dans le cas contraire.

### Exercice 2 – Conditionnelles

#### Question 1

Écrivez une fonction `verification` qui prend en argument un entier et retourne 1 si l'entier est pair, positif, comporte précisément 2 chiffres, et n'est pas divisible par 5, elle retourne 0 sinon.

Comme les booléens vrai et faux sont codés en C par les entiers 1 et 0 respectivement, on peut dire de votre fonction qu'elle vérifie que le paramètre en entrée satisfait la condition.

#### Question 2

Écrivez le test `test_verification` qui vérifie que votre fonction précédente est parfaitement opérationnelle.

### Exercice 3 – Médiane

1. Vous pouvez utiliser le `tp0` pour retrouver comment faire un `main` et des affichages

*Question 1*

Écrivez une fonction `max2` qui retourne le maximum de deux entiers.

*Question 2*

Écrivez une fonction `test_max` dans laquelle vous utiliserez des valeurs pertinentes pour tester votre fonction `max2`.

*Question 3*

Écrivez une fonction `mediane` qui retourne la valeur intermédiaire de trois entiers (la valeur qui a une autre valeur supérieure ou égale et une autre valeur inférieure ou égale).

*Question 4*

Écrivez une fonction `test_mediane`. Vous devriez avoir au moins 6 cas à tester.

## Renforcements

**Remarque:**

La section Renforcement permettra chaque semaine aux plus rapides de continuer de progresser.

### Exercice 4 – Des dés !

*Question 1*

Écrivez une fonction `quatre_des` qui prend un entier en paramètre et vérifie que cet entier est composé d'exactly quatre chiffres, chacun entre 1 et 6. Si tel est bien le cas, la fonction retourne la somme des quatre chiffres, sinon, elle retourne 0. Vous pourrez faire usage astucieusement de `/10` et `%10` afin de séparer les chiffres d'un nombre.

*Question 2*

Comme d'habitude, testez votre fonction avec une fonction `test_quatre_des`.

*Question 3*

Écrivez une fonction `quatre_des_croissant` qui prend un entier et l'interprète de la même façon, mais teste si les quatre chiffres sont de plus en plus grands. Si tel est bien le cas, la fonction retourne 1 (vrai), elle retourne 0 (faux) sinon.

*Question 4*

Comme d'habitude, testez votre fonction avec une fonction `test_quatre_des_croissant`.

### Exercice 5 – Date de naissance

*Question 1*

Écrivez une fonction `verifie_date_naissance` qui prend en paramètre trois entiers *jour*, *mois* et *annee* et vérifie qu'il s'agit d'une date de naissance valide. Une date de naissance est valide si c'est une véritable date et qu'elle se situe dans le passé. Vous pouvez utiliser des fonctions intermédiaires pour simplifier l'écriture.