

# TD N°4 : Les Fonctions

## 4.1 Hello world!

Écrire un programme qui définit et utilise :

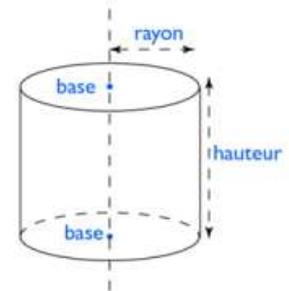
1. une fonction, nommée `un_salut()`, se contentant d'afficher "Hello world!" (elle ne possédera aucun paramètre ni valeur de retour),
2. une fonction, nommée `plusieurs_saluts()`, qui affiche "Hello world!" un nombre de fois égal à la valeur reçue en paramètre et qui ne renvoie aucune valeur, en utilisant la fonction précédente,
3. une fonction, nommée `trop_de_saluts()`, qui fait la même chose que `plusieurs_saluts()` et renvoie 1 si elle a affiché le message plus de 10 fois, 0 sinon, en utilisant la fonction précédente.

La fonction `main()` du programme utilisera `trop_de_saluts()` dans une instruction de test qui affichera "C'est beaucoup !" s'il y a plus de 10 messages affichés

*Remarque :* le résultat de `trop_de_saluts()` doit être considéré comme une valeur booléenne.

## 4.2 Cylindre

1. Écrire une fonction *Aire* qui permet de calculer l'aire d'un cylindre, sachant que l'aire d'un cylindre est égale à  $2\pi$  multiplié par le rayon de la base et par la hauteur.
2. Écrire une fonction *Volume* permettant de calculer le volume d'un cylindre, sachant que le volume d'un cylindre est égal à  $\pi$  multiplié par le rayon de la base au carré et par la hauteur.
3. Écrire un programme qui demande à l'utilisateur le rayon  $R$  de la base d'un cylindre et sa hauteur  $H$  et qui affiche son air et son volume en utilisant les deux fonctions ci-dessus.



## 4.3 PGCD

Le calcul du PGCD (Plus Grand Commun Diviseur) de deux entiers positifs  $a$  et  $b$  utilise l'algorithme d'Euclide suivant :

- Étape 1 : On réalise une division euclidienne du plus grand des deux nombres  $a$  par le second  $b$ , pour trouver un quotient  $q$  et un reste  $r$  ( $a = q.b + r, r < b$ ).
- Étape 2 : Répéter l'étape 1 (avec les nombres conservés :  $b$  devient le nouveau  $a$  et  $r$  devient le nouveau  $b$ ) jusqu'à arriver à un reste nul.
- Étape 3 : Le PGCD des nombres  $a$  et  $b$  de départ est égal au dernier reste non nul.

☞ **Exemple.** si  $a=96$  et  $b=81$ , les calculs sont les suivants :

a	b	r
96	= 1 * 81	+ 15
81	= 5 * 15	+ 6
15	= 2 * 6	+ 3
6	= 2 * 3	+ 0

Écrire une fonction qui calcule et renvoie le PGCD de deux entiers passés en paramètres.

## 4.4 Calculatrice

On souhaite écrire un programme simulant une calculatrice. On dispose de trois nombres réels. Selon un choix effectué à partir d'un menu affiché à l'écran, on désire calculer la somme, le produit, la moyenne, le minimum ou le maximum de ces trois nombres.

Pour cela, écrire :

1. une fonction pour calculer la somme de trois réels,
2. une fonction pour calculer le produit de trois réels,
3. une fonction pour calculer la moyenne de trois réels,
4. une fonction qui retourne le minimum entre trois réels,
5. une fonction qui renvoie le maximum entre trois réels,
6. une fonction qui permet d'afficher le menu ci-contre,
7. un programme qui permet à l'utilisateur d'utiliser la calculatrice autant de fois qu'il le souhaite (l'utilisateur doit taper 0 pour quitter),
8. une fonction qui retourne le minimum entre six réels, que l'on devra tester dans la fonction `main()`.

```
***** MENU *****
*
* 1 : ----> Somme ----- *
* 2 : ----> Produit ----- *
* 3 : ----> Moyenne ----- *
* 4 : ----> Minimum ----- *
* 5 : ----> Maximum ----- *
* ----- *
* 0 : ----> Quitter ----- *
*****
Donnez votre choix :
```

## 4.5 Réviser la multiplication

On veut écrire un programme qui demande à un utilisateur de réciter sa table de multiplication. L'utilisateur commence par entrer un nombre. Ensuite le programme affiche une à une les lignes de la table de multiplication de ce nombre, en laissant le résultat vide et en attendant que l'utilisateur entre le résultat. Si celui-ci est incorrect, on affiche un message d'erreur donnant la bonne valeur. Ensuite on passe à la ligne suivante. Si à la fin toutes les réponses sont correctes, on affiche un message de félicitations, sinon on affiche le nombre d'erreurs commises.

On représente ci-contre une exécution possible (les entrées de l'utilisateur sont affichées en italique)

Pour cela on écrira :

1) **une fonction lire\_n** qui permet de lire la valeur du nombre dont l'utilisateur veut réviser la table de multiplications. Ce nombre devra être compris entre MIN=2 et MAX=9 (MIN et MAX étant des constantes définies au début du programme). Si le nombre est incorrect, la fonction redemande un entier. Elle retournera la valeur entrée une fois qu'elle sera valide.

2) **une fonction une\_multiplication** qui affiche une ligne de la table et récupère la saisie de l'utilisateur puis vérifie si elle est correcte. Elle prendra en paramètres les nombres à multiplier. La valeur de retour permettra de savoir si la valeur saisie est correcte.

3) **une fonction reviser\_multiplications** qui appelle plusieurs fois la fonction précédente pour réviser les multiplications de chaque ligne de la table de multiplication du nombre qui lui est passé en paramètre. Elle renverra le nombre d'erreurs commises par l'utilisateur.

4) **la fonction principale (main)** utilise ces trois fonctions puis affiche le résultat (félicitations ou nombre d'erreurs commises).

5) **ajouter au programme une fonction** qui permet de réviser les **carrés** des nombres entiers compris dans l'intervalle donné par l'utilisateur. Elle prendra comme paramètres les bornes de l'intervalle et devra faire appel à la fonction `une_multiplication` de la question 2. Le choix de la révision sera fait grâce à un menu. Comme le montre l'exemple d'exécution ci-contre.

```
RÉVISION DES TABLES DE MULTIPLICATIONS
-----
Vous voulez réviser les multiplications par : 7
1 x 7 = 7
2 x 7 = 14
3 x 7 = 28
Erreur ! 3 x 7 = 21 et non 28
4 x 7 = 28
5 x 7 = 36
Erreur ! 5 x 7 = 35 et non 36
6 x 7 = 42
7 x 7 = 49
8 x 7 = 56
9 x 7 = 63
10 x 7 = 70
Vous avez fait 2 erreurs !
Vous n'avez pas encore appris votre table de
multiplications par 7
```

```
| RÉVISION DES MULTIPLICATIONS |
|-----|
| 1- Révision d'une table      |
| 2- Révision des carrés      |
|-----|
| 0- Quitter                   |
|-----|
|                               |
|      Votre choix ? 2        |
|                               |
| Révision des carrés :      |
| à partir de : 2            |
| jusqu'à : 6                |
| 2 x 2 = 4                  |
| 3 x 3 = 9                  |
| 4 x 4 = 16                 |
| 5 x 5 = 25                 |
| 6 x 6 = 36                 |
| Bravo ! Vous connaissez tous vos carrés de 2² à 6² |
| ...(réaffichage du menu...)...etc.
```