

TD5 /TP 5 : Les Fonctions

Exercice 1 :

Écrire la fonction permettant de déterminer si un nombre (entier) passé en paramètre en impair ou non. Cette fonction doit retourner 1 si le nombre est impair et 0 s'il est pair.

Tester la fonction dans un programme interactif. C'est le programme principal qui affichera si le nombre est pair ou impair.

Exercice 2 :

- Écrire une fonction qui affichera le menu suivant:

```
*****MENU*****  
1---->Français  
2---->Anglais  
3---->Espagnol  
4---->Quitter  
*****  
Choisir votre langue:
```

- Écrire une fonction qui affichera le menu suivant (le Menu sera affiché selon la langue choisi): Exemple en Français

```
*****MENU*****  
1---->Racine carrée  
2---->carrée  
3---->cube  
4---->Inverse  
5---->Somme  
6---->Différence  
7---->Produit  
8---->Division entiere  
9---->Quitter  
*****  
Entrez votre choix (1-9) :
```

- Ecrire une fonction qui calculera la Racine carrée
- Ecrire une fonction qui calculera le carrée
- Ecrire une fonction qui calculera le cube
- Ecrire une fonction qui calculera l'inverse
- Ecrire une fonction qui calculera la somme
- Ecrire une fonction qui calculera la différence
- Ecrire une fonction qui calculera le produit
- Ecrire une fonction qui calculera la division entière
- Ecrire un programme utilisant les fonctions précédentes

Exercice 3 :

- Ecrire la fonction *NCHIFFRES* du type *int* qui obtient une valeur entière *N* (positive ou négative) du type *long* comme paramètre et qui fournit le nombre de chiffres de *N* comme résultat.
- Ecrire un petit programme qui teste la fonction *NCHIFFRES*.

Exemple :

Introduire un nombre entier : 6457392

Le nombre 6457392 à 7 chiffres.

Exercice 4 :

En mathématiques, on définit la fonction factorielle de la manière suivante:

$0! = 1$

$n! = n*(n-1)*(n-2)*... * 1$ (pour $n > 0$)

Ecrire une fonction *FACT* du type **double** qui reçoit la valeur *N* (type **int**) comme paramètre et qui fournit la factorielle de *N* comme résultat. Ecrire un petit programme qui teste la fonction *FACT*.

Exercice 5 :

- Ecrire une fonction *SMA* qui affichera la liste des modules de la filière *SMA*
- Ecrire une fonction *SMI* qui affichera la liste des modules de la filière *SMI*
- Ecrire une fonction *Condition* qui affichera les conditions d'accès à chaque filière
- Ecrire un programme utilisant les fonctions précédentes

Exercice 6 :

Pour faire cet exercice Utiliser les fonctions prédéfinies suivantes :

int rand(void): returns a pseudo-random number in the range of [0, RAND_MAX).

RAND_MAX: is a constant whose default value may vary \between implementations but it is granted to be at least 32767.

1. Créer une fonction *De*, qui prend en entrée 1 entier qui correspond au nombre de faces d'un dé et qui retourne aléatoirement un numéro de face.

Exemple *de(6)* correspond à un lancé de dé de 6, donc un retour entre 1 et 6.

2. Créer une fonction *Des*, qui prend en entrée 2 entiers - Nombre de dés à lancer. - Nombre de faces par dé.

Cette fonction retourne alors la somme des résultats de chaque dé.