



Travaux Pratiques N° 5

Les fonctions en langage C

EXERCICE 1 : Analyse d'un programme

Que le résultat fournira ce programme :

```
#include <stdio.h>
int n=10, q=2 ;

main()
{
    int fct (int) ;
    void f (void) ;
    int n=0, p=5 ;
    n = fct(p) ;
    printf ("A : dans main, n = %d, p = %d, q = %d\n", n, p, q) ;
    f() ;
}

int fct (int p)
{
    int q ;
    q = 2 * p + n ;
    printf ("B : dans fct, n = %d, p = %d, q = %d\n", n, p, q) ;
    return q ;
}

void f (void)
{
    int p = q * n ;
    printf ("C : dans f, n = %d, p = %d, q = %d\n", n, p, q) ;
}
```

EXERCICE 2 : Conversion de température

En utilisant les fonctions, Ecrire un programme en C qui convertit les degrés Fahrenheit en degrés Celsius selon la formule : $TC = 5/9(TF - 32)$.

EXERCICE 3 : Conversion de devise

Écrivez un programme contenant une fonction qui fournit une valeur en dirhams à partir d'un prix entré en euros. On supposera que le taux de change est 1Euro = 11,2 Dirhams

Le résultat doit être sous la forme :

Entrez un prix en euros : 10



10.00 euros = 112.00 dirhams

EXERCICE 4 : Opérations sur les nombres

Écrire un programme permettant de saisir un entier n supérieur à zéro (par exemple : 5284).

Le programme calcule (par les fonctions) et affiche à l'écran :

- la somme des chiffres du nombre n (ici 19 car $5+2+8+4 = 19$);
- l'envers du nombre n (ici 4825).

EXERCICE 5 : Calcul de Prime

Écrire une fonction permettant de calculer le bonus dépendant du poste de travail :

poste 'A' (analyste)	: 234.50 \$ de bonus
poste 'P' (programmeur)	: 210.90 \$ de bonus
poste 'O' (opérateur)	: 189.00 \$ de bonus

Écrire quelques appels (utilisations) valides de cette fonction.

EXERCICE 6 : QCM

Toutes les semaines, un professeur donne à ces étudiants un QCM dont le nombre de questions varie d'une fois sur l'autre, mais dans lequel toutes les questions ont toujours 4 entrées (A, B, C et D). Un étudiant pensant qu'il n'a pas le temps de travailler cette matière (puisqu'il a, soit disant, trop à faire entre la programmation C, les maths discrètes, l'algèbre linéaire et/ou la microéconomie) décide d'écrire un programme qui génère aléatoirement sa feuille de réponse à ces QCM hebdomadaires... Une exécution de son programme affiche par exemple à l'écran :

Entrer la date:

jour: 10

mois: 2

annee: 2010

Nombre de Questions: 5

Date: 10-2-2010

Question 1: A

Question 2: C

Question 3: D

Question 4: B

Question 5: B

Voici les fonctions que cet étudiant a utilisées :

1. La fonction `date`, dont le prototype est `void date(int j, int m, int a)`; qui affiche à l'écran la date sous la forme `Date: j-m-a`.
2. La fonction `nombre_questions`, dont le prototype est le suivant : `int nombre_questions(void)`; qui demande à l'utilisateur de rentrer le nombre de questions du QCM et le renvoie en valeur de retour. Le nombre de questions peut varier entre 1 et 50. Si l'utilisateur entre un nombre n'étant pas dans cet intervalle, la question lui est reposée.



3. La fonction question, dont le prototype est le suivant : void question (int i, char c); qui affiche à l'écran Question i: c puis passe à la ligne suivante.
4. La fonction reponse, dont le prototype est le suivant : char reponse (void); qui renvoie de manière aléatoire A, B, C ou D.
Ecrire les différents corps des fonctions ci-dessus, puis le programme entier.

EXERCICE 7:

Ecrire la fonction int indiceval(double val, double tab[], int nb) qui recherche la valeur val dans le tableau tab de nb réels double précision. Cette fonction retourne l'indice du premier élément du tableau égal à val, ou -1 si cette valeur n'existe pas dans le tableau.

EXERCICE 8:

Ecrire la fonction : double moyenne(double ttab[], int nb, int debut, int fin) qui retourne la moyenne des éléments dont les indices sont compris entre debut et fin. Le tableau tab est un tableau contenant nb réels double précision.

EXERCICE 9 : Matrices

1. Ecrire une fonction saisir qui permet de saisir une matrice carrée
2. Ecrire une fonction afficher qui permet d'afficher les éléments d'une matrice
3. Ecrire une fonction additionner qui permet d'additionner deux matrices
4. Ecrire une fonction multiplier qui permet de multiplier deux matrices
5. Ecrire une fonction menu qui demande à l'utilisateur quelle est l'opération à effectuer
6. Ecrire le programme principal

EXERCICE 10 : Recherche des Points-Clos MaxLignes

Un Points-Clos MaxLignes est un élément d'une matrice qui est à la fois:
un maximum sur la ligne
et
un minimum sur la colonne

Exemples:

3	2	1
6	5	4
9	8	7

1	7	9	8
4	6	0	6

0	7
8	9
6	7
8	8

1. Ecrire une fonction *SaisirMatrice* qui permet de saisir :
 - le nombre de lignes
 - le nombre de colonnes



- les éléments d'une matrice MAT
2. Ecrire une fonction *AfficherMatrice* qui affiche les éléments d'une matrice.
 3. Ecrire une fonction *MaxLigne* qui crée une nouvelle matrice *Maxlig* à partir de la matrice saisie MAT tel que:

$$\text{MaxLig}[i,j] = \begin{cases} 1 & \text{si Mat}[i,j] \text{ est un maximum sur la ligne } i \\ 0 & \text{sinon} \end{cases}$$

4. Ecrire une fonction *MinColonne* qui crée une nouvelle matrice *MinCol* à partir de la matrice saisie MAT tel que:

$$\text{MinCol}[i,j] = \begin{cases} 1 & \text{si Mat}[i,j] \text{ est un minimum sur la colonne } j \\ 0 & \text{sinon} \end{cases}$$

5. Ecrire une fonction *TrouvePointsClos* permet trouver et afficher tous les Points-Clos MaxLigne d'une matrice Mat (c'est à dire: cette fonction affiche les valeurs des éléments ainsi que leurs numéros de lignes.