Université Sidi Mohamed Ben Abdellah **Année Universitaire: 2015 / 2016**

Faculté des Sciences Dhar El Mahraz Fès **S3: SMI**

Département d'Informatique

Programmation I : Session Exceptionnelle Durée : 1 h ,30 mn

Exercice 1

1) Dans le programme C suivant, donnez les erreurs et les corrections à effectuer.

```
int addition(int a, int b);
{
int a, b, c;
c = a + b ;
return c;}
```

2) Soit le code suivant :

```
int i ;
printf("i ? ") ;
scanf("%d", &i) ;
switch (i) {
case 0 : printf(" NUL") ; break ;
case 1 : case 3 : case 5 : case 7 : case 9 :
printf(" IMPAIR") ; break ;
case 2 : case 4 : case 6 : case 8 :
printf(" PAIR") ; break ;
default : printf("NEGATIF OU PAS UN CHIFFRE") ; break ;}
```

Réécrivez l'instruction ***switch*** en n'utilisant que des instructions ***if***.

3) Ecrire une fonction ***MAX2*** en langage C qui à partir de deux entiers donnés, retourne le maximum.

4) Ecrire une fonction ***MAX3*** en langage C qui retourne le maximum de trois entiers en faisant appel à la fonction ***MAX2***.

5) Que produit le programme suivant ?

```
#include <stdio.h>
int main() {
int tab[] = {0, 1, 2, 3 };
int *i1 = tab + 1;
int *i2 = tab + 2;
int a = ++*i1 + *i2++;
int b = *++i1 + *i2--;
printf("%d#%d", a, b);
return 0;}
```

Exercices 2 :

Écrivez une fonction qui calcule la résistance équivalente d'un nombre quelconque de résistances en parallèle.

Rappel : $\frac{1}{R} = \sum_{i=1}^{n} \frac{1}{R_i}$

Prototype : *double resistance (double r[], int n);*

Exercices 3 :

Ecrire un programme en langage C permettant de :

1. Saisir les notes d'une matière pour une classe de 30 étudiants
2. Calculer et afficher la moyenne des notes
3. Chercher et afficher la meilleure et la plus mauvaise note
4. Calculer le nombre d'étudiants ayant une note supérieure à 10 (noté nbr_sup_10)
5. Tester si la moyenne de la classe, est inférieure à 8 alors il faut ajouter 1 point aux étudiants ayant une note supérieure à 8 et 0,5 aux étudiants qui ont une note entre 8 est 4.

NB : Utiliser le formalisme pointeur à chaque fois que cela est possible.

Exercices 4 :

Ecrire un programme à ***l'aide des fonctions*** qui lit les dimensions L et C d'un tableau T à deux dimensions du type ***int*** (dimensions maximales: 50 lignes et 50 colonnes). Remplir le tableau par des valeurs entrées au clavier et afficher le tableau ainsi que ***la somme des éléments pairs de chaque ligne et la somme des éléments impairs de chaque colonne.***

Université Sidi Mohamed Ben Abdellah **Année Universitaire: 2015 / 2016**

Faculté des Sciences Dhar El Mahraz Fès **S3: SMI**

Département d'Informatique

Programmation I : Session Exceptionnelle : CORRIGE

Corrigé 1

1)

```
int addition(int a, int b);
{
int a, b, c;
c = a + b ;
        return c;
}
```

2)

```
int i ;
printf("i ? ") ;
scanf("%d", &i) ;
if (i == 0) {
printf(" NUL") ;
}
else if (i == 1 || i ==3 || i == 5 || i == 7 || i == 9) {
printf(" IMPAIR") ;
}
else if (i == 2 || i ==4 || i == 6 || i == 8) {
printf(" PAIR") ;
}
else {
printf("NEGATIF OU PAS UN CHIFFRE") ;
}
```

3)

```
Fonction : max2 (a, b : entier) : entier
Var : max2 : entier
Début :          Si a<b alors max2 <= b
          Sinon max2 <= a
Retourner (max2)
Fin
```

En C

```
int max2(int a, int b)
{
int max2 ;
```

```
        if (a<b) {
        max2=b ;}
        else {
        max2=a;}
return(max2);
}
```

4)

Fonction : max3 (a, b, c : entier) : entier :
Var : max3 : entier
Début : max3 <= max2 [max2 (a, b), c)
Retourner (max3)
Fin

```
int max3(int a, int b, int c)
{ int max3 ;
max3=max2(max2(a,b),c);
return(max3);
}
```

5)

affiche 4#5

Corrigé 2

C'est un schéma classique d'accumulation (somme des inverses des éléments du tableau), puis retour de l'inverse de l'accumulateur. Un écueil à éviter est l'oubli du cas où un des éléments est nul, ce qui conduirait à une division par 0. La solution consiste à retourner directement 0 si l'un des éléments est nul.

```
double resistance (double r[], int n) {
double req=0.0;
int i;
for (i=0;i<n;++i) {
if (r[i]==0.0)
return 0.0;
req += 1/r[i];
}
return 1/req;
}
```

Corrigé 3 (voir TD)

Corrigé 4

a)

```
void LIRE_DIM (int *L, int LMAX, int *C, int CMAX)
        {
              /* Saisie des dimensions de la matrice */
              do
               {
                printf("Nombre de lignes de la matrice   (max.%d) : ",LMAX);
                scanf("%d", L);
               }
              while (*L<0 || *L>LMAX);
              do
               {
                printf("Nombre de colonnes de la matrice (max.%d) : ",CMAX);
                scanf("%d", C);
               }
              while (*C<0 || *C>CMAX);
         }
```

b) Ecrire la fonction LIRE_T à quatre paramètres T, L, C, et CMAX qui lit les composantes d'un tableau T du type int et de dimensions L et C.

```
void LIRE_T (int *T, int L, int C, int CMAX)
{
 /* Variables locales */
 int I,J;
 /* Saisie des composantes de la matrice */
 for (I=0; I<L; I++)
     for (J=0; J<C; J++)
          {
           printf("Elément[%d][%d] : ", I, J);
           scanf("%d", T + I*CMAX + J);
          }
}
```

c) Ecrire la fonction ***affiche_T*** à quatre paramètres T, L, C, et CMAX qui lit les composantes d'un tableau T du type int et de dimensions L et C.

```
void affiche_T (int *T, int L, int C, int CMAX)
{
 /* Variables locales */
 int I,J;
 /* Saisie des composantes de la matrice */
 for (I=0; I<L; I++)
     for (J=0; J<C; J++)

          printf("%d ", *(MAT + I*CMAX + J));
}
```

d) Ecrire la fonction ***somme_p_T*** à quatre paramètres T, L, C, et CMAX qui lit les composantes d'un tableau T du type int et de dimensions L et C.

```
int  somme_p_T (int *T, int L, int C, int CMAX)
{
 /* Variables locales */
 int I,J,S ;
 /* Saisie des composantes de la matrice */
 for (I=0; I<L; I+=2)
     for (S=0,J=0; J<C; J++)
         S+=T[I] [J];
Return(S);

 }
```

e) Ecrire la fonction ***somme_Imp_T*** à quatre paramètres T, L, C, et CMAX qui lit les composantes d'un tableau T du type int et de dimensions L et C.

```
int  somme_Imp_T (int *T, int L, int C, int CMAX)
{
 /* Variables locales */
 int I,J,S ;
 /* Saisie des composantes de la matrice */
 for (J=1; J<C; J+=2)
     for (S=0,I=0; I<L; I++)
         S+=T[I] [J];
Return(S);
 }
```