

*Exemple de corrigé*

**Exercice 1**

Définir une structure de données `Heure` permettant de représenter une heure au format hh/mm/ss, puis écrire les fonctions suivantes :

- 1) conversion d'un élément de type `Heure` en nombre de secondes (entier)
- 2) conversion d'un nombre de secondes (entier) en un élément de type `Heure`
- 3) addition de deux éléments de type `Heure`

```
→
typedef struct
{
    int hh;
    int mm;
    int ss;
} Heure;

int HeureEnSecondes(Heure h)
{
    return (h.hh*3600 + h.mm*60 + h.ss);
}

Heure SecondesEnHeure(int sec)
{
    Heure h;
    h.hh = sec/3600;
    sec = sec%3600;
    h.mm = sec/60;
    sec = sec%60;
    h.ss = sec;
    return h;
}

Heure AddHeures(Heure h1, Heure h2)
{
    Heure h3;
    return (SecondesEnHeure(HeureEnSecondes(h1)+HeureEnSecondes(h2)));
}
```

Autre solution :

```
Heure AddHeures2(Heure h1, Heure h2)
{
    Heure h3;
    h3.hh=h1.hh+h2.hh;
    h3.mm=h1.mm+h2.mm;
    h3.ss=h1.ss+h2.ss;
    if (h3.ss > 59)
    {
        h3.ss -= 60;
        h3.mm += 1;
    }
    if (h3.mm > 59)
    {
        h3.mm -= 60;
        h3.hh +=1;
    }
    return h3;
}
```

## **Exercice 2**

On souhaite compléter le programme suivant :

```
main()
{
    float a = 7.8, b = 3.6;
    float* c;
    c = calcul( # );
    printf("%f\n", # );
    free c;
}
```

- 1) Ecrire une fonction `calcul` adaptée à ce `main`, qui permet de renvoyer le résultat du calcul  $x+5y-2$ , et dont le profil est le suivant :

```
float* calcul(float* x, float* y)
```

- 2) Que doit-on mettre à la place des "#" dans le `main` afin que le programme affiche le résultat du calcul ? Expliquez.

```

→
float* calcul(float* x, float* y)
{
    float *ptr = (float*) malloc(sizeof(float));
    *ptr = ((*x) + 5 * (*y) - 2);
    return ptr;
}

main()
{
    float a = 7.8, b = 3.6;
    float* c;
    c = calcul( &a, &b );
    printf("%f\n", *c );
    free (c);
}

```

### Exercice 3

Dire ce que calcule la fonction f selon les différentes valeurs des arguments x et y :

```

int f(int x, int y)
{
    if (x==0)
        return y;
    else
        return (f(x-1, x+y));
}

```

→

renvoie :

- ( si  $x = 0$  :  $y$  ) affirmation incluse dans la suivante :
- (si  $x \geq 0$ , ou) si  $x > 0$  :  $\sum_{i=1}^{i \leq x} i + y$
- si  $x < 0$  ne termine jamais

### Exercice 4

On souhaite gérer une base de données d'inscriptions pour l'organisation d'un congrès qui dure une journée. Les organisateurs proposent aux participants de s'inscrire pour des repas, ainsi que pour l'hébergement en hôtel.

Un participant peut s'inscrire indépendamment aux 2 repas proposés : déjeuner (15 euros) et/ou dîner (35 euros) ou aucun. Il n'est pas obligé de prendre un hôtel. S'il en prend un, il peut choisir parmi 2 types d'hôtels différents : 2 étoiles (75 euros) ou 3 étoiles (100 euros).

Un participant peut venir accompagné de son conjoint. Dans ce cas, la réservation d'hôtel est identique mais lorsqu'un repas est sélectionné alors il faut en compter 2.

- 1) Créer un nouveau type `Participant` qui inclut son nom (chaîne de caractères : tableau de 20 caractères), son prénom (chaîne de caractères : tableau de 30 caractères), ainsi que

toutes les autres informations nécessaires à son inscription selon les critères définis ci-dessus. On privilégiera une structure contenant un nombre minimal de champs.

- 2) Créer un nouveau type `TabPart` qui est un tableau de 100 éléments de type `Participant`.
- 3) Créer une fonction `Nb2Etoiles` qui, pour un argument de type `TabPart` donné, affiche le nom et le prénom des personnes qui ont choisi de réserver un hôtel 2 étoiles. On pourra utiliser la fonction `printf` avec pour format `%s` pour l'affichage des chaînes de caractères nom et prénom.
- 4) Créer une fonction `NbDej` qui, pour un argument de type `TabPart` donné, retourne le nombre de déjeuners à prévoir.
- 5) Créer une fonction `Montant` qui calcule, pour un `Participant` donné en argument, le montant de sa facture.

→

```
typedef struct
{
    char nom[20];
    char prenom[30];
    int dejeuner; // 0=non, 1=oui
    int diner; // 0=non, 1=oui
    int hotel; // 1=pas d'hotel, 2=2etoiles, 3=3etoiles
    int seul; // 0=non, 1=oui
} Participant;

typedef Participant TabPart[100];

void Nb2Etoiles(TabPart tab)
{
    int i;
    for (i=0; i<100; i++)
    {
        if (tab[i].hotel==2)
        {
            printf("nom: %s, prenom:%s\n", tab[i].nom,
tab[i].prenom);
        }
    }
}

int NbDej(TabPart tab)
{
    int nb=0, i;
    for (i=0; i<100; i++)
    {
        if (tab[i].dejeuner==1)
        {
            nb+=1;
            if (tab[i].seul==0)
                nb+=1;
        }
    }
    return nb;
}
```

```
int Montant(Participant p)
{
    int total=0;
    if (p.dejeuner==1)
        total += 15;
    if (p.diner==1)
        total += 35;
    if (p.seul==0)
        total *= 2;
    if (p.hotel==2)
        total += 75;
    if (p.hotel==3)
        total += 100;
    return total;
}
```