

Série 4 : Programmation C++ (Structures)

Exercice 1 :

Soit le modèle (type) de structure suivant :

```
struct point
{ char c ;
  int x, y ;
} ;
```

Écrire une fonction qui reçoit en argument une structure de type `point` et qui en affiche le contenu sous la forme : `point B de coordonnées 10 12`

- en transmettant en argument la valeur de la structure concernée,
- en transmettant en argument l'adresse de la structure concernée,
- en transmettant la structure concernée par référence.

Dans les trois cas, on écrira un petit programme d'essai de la fonction ainsi réalisée.

Exercice 2 :

Soit le type structure `enreg` défini ainsi :

```
const int NMOIS = 12 ;
struct enreg
{ int stock ;
  float prix ;
  int ventes [NMOIS]
}
```

Écrire une fonction nommée `raz` qui « met à zéro » les champs `stock` et `ventes` d'une structure de ce type, transmise en argument. La fonction ne comportera pas de valeur de retour.

Écrire un petit programme d'essai qui affecte tout d'abord des valeurs aux différents champs d'une telle structure, avant de leur appliquer la fonction `raz`. On affichera les valeurs de la structure, avant et après appel (on pourra s'aider d'une fonction d'affichage).

Exercice 3 :

Soit la structure suivante, représentant un point d'un plan :

```
struct point
{ char c ; // nom du point
  int x, y ; // coordonnées
} ;
```

- Écrire la déclaration d'un tableau (nommé `courbe`) de `NP` points (`NP` supposé défini par une constante).
- Écrire une fonction (nommée `affiche`) qui affiche les valeurs des différents « points » du tableau `courbe`, transmis en argument, sous la forme :

```
point D de coordonnées 10 2
```

- Écrire un programme qui :

- lit en données des valeurs pour le tableau `courbe` ;
- fait appel à la fonction précédente pour les afficher.

Exercice 4 :

Soient les deux modèles de structure `date` et `personne` déclarés ainsi :

```
const int LG_NOM = 30 ;
struct date
{ int jour ;
  int mois ;
  int annee ;
} ;
struct personne
{ char nom [LG_NOM+1] ; // chaîne de caractères (de style C)
  // représentant le nom
  struct date date_emploi ;
  struct date date_poste ; } ;
```

Écrire une fonction qui reçoit en argument une structure de type `personne` et qui en remplit les différents champs avec un dialogue se présentant sous l'une des 2 formes suivantes :

```
nom : DUPONT
date emploi (jj mm aa) : 16 1 75
date poste = date emploi ? (O/N) : O
nom : DUPONT
date emploi (jj mm aa) : 10 3 81
date poste = date emploi ? (O/N) : N
date poste (jj mm aa) : 23 8 91
```