

TD 4 : Structures

IUT CAEN – Département Informatique

Module M1102

Exercice1

On considère les énumérations suivantes :

```
enum couleur_e COEUR, CARREAU, PIQUE, TREFLE ;
```

```
enum valeur_e SEPT, HUIT, NEUF, DIX , VALET, DAME, ROI , AS ;
```

Nous souhaitons définir une structure `carte` composée des champs `couleur` (de type `couleur_e`) et `valeur` (de type `valeur_e`) et un tableau de cartes (noté `jeu`) permettant de représenter un jeu complet de 32 cartes.

1. Écrivez un programme qui déclare les énumérations `couleur_e` et `valeur_e`, définit la structure `carte` et initialise le tableau `jeu`.
2. Complétez votre programme de façon à afficher un nombre indiqué de cartes d'un ensemble de cartes de jeu.

Remarque : pour simplifier l'affichage, vous utiliserez les deux tableaux suivants :

```
char ch_coul [] [8] = {"Coeur", "Carreau", "Pique", "Trefle"} ;
```

```
char ch_val [] [6] = {"7", "8", "9", "10", "Valet", "Dame", " Roi ", "As"} ;
```

3. Complétez votre programme pour qu'il mélange (de la manière qui vous paraît la plus simple) le jeu de carte.
4. On souhaite distribuer les cartes sur différents joueurs. Tous les joueurs doivent recevoir autant de cartes et s'il reste des cartes en trop, elles ne seront pas données (ainsi pour un jeu de 32 cartes distribué à 6 joueurs, chaque joueur recevra 5 cartes et 2 cartes ne seront pas utilisées).

La main de chaque joueur sera représentée par un tableau de cartes, et les mains obtenues par les différents joueurs par un tableau à deux dimensions. Complétez votre programme pour qu'il puisse distribuer les cartes de jeu. On supposera que le nombre de joueurs est une variable qui sera saisie par votre programme.

Exercice2

L'objet de base que nous considérons dans cet exercice est le point. Il est défini par ses coordonnées x et y (de type entier) et une couleur d'affichage (du type chaîne de caractères limitée à 10 caractères, n'oubliez pas de réserver la place pour le caractère `'\0'` qui signale la fin de chaîne).

1. Donnez le code de la structure `point`. Déclarez et initialisez ensuite un point nommé `mon_point`, situé aux coordonnées $x = 2$, $y = 3$ et de couleur rouge.
2. On veut maintenant pouvoir décrire des rectangles dont les côtés sont horizontaux et verticaux. Un rectangle est alors déterminé par les coordonnées de deux coins opposés (en bas à gauche et en haut à droite), la couleur d'affichage des traits et celle du fond. Écrivez une structure `rectangle` en utilisant la structure `point`.

3. Déclarez et initialisez une variable `un_rectangle` de type `rectangle` dont les deux sommets opposés sont respectivement aux coordonnées (100,200) et (300,2), la couleur d’affichage des traits et des points est le rouge et la couleur du fond est le blanc. Vous proposerez une solution dans laquelle aucune variable de type `point` n’est déclarée et une autre dans laquelle deux variables de type `point` sont déclarées et utilisées pour définir la variable `un_rectangle`.
4. Écrivez un programme qui, étant donné un point et un rectangle, affiche 1 (vrai) si le point est dans le rectangle (les bords du rectangle sont dans le rectangle) et 0 (faux) sinon.

Exercice3

On veut gérer une bibliothèque d’un laboratoire contenant au maximum 100 livres. Le nombre maximum de livres sera défini par la primitive `#define`. Chaque livre est identifié par :

- un titre (de type chaîne de caractères),
- une côte (ou numéro d’identification de type entier),
- un auteur (de type `Personne`),
- un booléen indiquant si le livre est emprunté.

1. Définissez :

- un type de structure `Personne` qui contient deux champs :
 - le nom d’une personne (de type chaîne de caractères),
 - le prénom d’une personne (de type chaîne de caractères).
- un type de structure `Livre` qui permet de représenter un livre.

2. Écrivez un programme qui permet de saisir une personne ainsi qu’un nouveau livre.