

TP 4 – Programmation C

Tableaux – Structures de données - Fonctions

Temps prévu (3h)

Exercice 1 :

Soit le modèle (type) de structure suivante :

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

1. Afficher le contenu d'une variable de type *struct point* sous la forme :
point B de coordonnées 10 12
2. Écrire la déclaration d'un tableau (nommé **courbe**) de NP points (NP supposé défini par une instruction (**#define**)).
3. Afficher les valeurs des différents "points" du tableau courbe, après sont remplissage, sous la forme précédente de 1.
4. Créer une structure *pixel* dont les champs sont les suivants :
 - a. **p** de type *struct point*
 - b. **coul** de type *char*
5. Écrire la déclaration d'un tableau (nommé **image**) de N pixel (N supposé défini par une instruction (**#define**)). Remplir et afficher le tableau image.

Exercice 2 : facturation par code

Réaliser un programme établissant une facture pouvant porter sur plusieurs articles. Pour chaque article à facturer, l'utilisateur ne fournira que la quantité et un numéro de code à partir duquel le programme devra retrouver à la fois le libellé et le prix unitaire.

Le programme devra refuser les codes inexistants. A la fin, il affichera un récapitulatif tenant lieu de facture.

Les informations relatives aux différents articles seront définies dans la source même du programme. Elles seront toutefois placées à un niveau global, de manière à pouvoir, le cas échéant, faire l'objet d'une source séparé, appellable par **#include**.

On prévoira deux fonctions :

- Une pour rechercher les informations relatives à un article, à partir de son numéro de code,
- Une pour afficher la facture récapitulative.

Exemple

Combien d'articles à facturer ? 3

Code article ? 25

Quantité de produitX au prix unitaire de 370 ? 33

Code article ? 7 ?

** article inexsistant – redonnez le code : 16

Quantité de produitY au prix unitaire de 199,50 ? 12

Code article ? 26

Quantité de produitZ au prix unitaire de 295,25 ? 6

Facture

Article	Nbre	PU	Montant
ProduitX	33	370	12210
ProduitY	12	199,50	2394
ProduitZ	6	295,25	1771,50

TP N° 5

Exercice 1 :

Ecrire un programme de cryptage (de chiffrement) "Rot 13". C'est un algorithme de chiffrement de texte. Il s'agit d'un décalage de 13 caractères de chaque lettre du texte à chiffrer. Entre autre c'est une rotation des lettres de 13 positions, c'est-à-dire qu'après la lettre Z on a la lettre A.

Exp. : correspondance entre les caractères en clair et chiffrés :

Caractère en clair	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Caractère chiffré	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M

On supposera que le texte ne contiendra pas des chiffres, des ponctuations, et des symboles.

On y prévoira :

- Une fonction pour la lecture du texte à chiffrer se fait par l'entrée standard
- Une fonction de chiffrement du texte.
- Une fonction de déchiffrement du texte

- Exercice 2

On veut écrire un programme qui applique le tri de sélection sur un tableau d'entiers à l'aide des fonctions.

- 1 - Ecrire la fonction REMPLIR qui permet le remplissage d'un tableau d'entiers.
 - 3 - Ecrire la fonction INSERER qui permet d'insérer un entier dans un tableau déjà trié.
 - 2 - Ecrire la fonction TRI_INSERTION qui permet de trier par ordre croissant les éléments d'un tableau à N entiers donné comme argument en utilisant la fonction INSERER.
- Ecrire la fonction FUSION qui construit un tableau trié par ordre croissant avec les éléments de deux tableaux A et B triés eux aussi par ordre croissant.
 - Ecrire la fonction AFFICHER qui permet l'affichage du tableau trié.
 - Ecrire un programme pour tester toutes les fonctions créées.

NB:

- Utiliser les pointeurs.
- Trier le tableau de gauche à droite en insérant à chaque fois l'élément $I+1$ dans le tableau (déjà trié) des I premiers éléments.

