Travaux Pratiques d'algorithmique n°2

Cours d'Informatique de Deuxième Année

—DEUG Sciences S3-MI—

Représentation mémoire, permutations et structures

Au cours de cette séance, nous aborderons les points suivants : représentation des entiers en mémoire, manipulation de données par permutations circulaires et les structures sous divers aspects (rappels de syntaxe, opérateur `->`, tableaux).

▸ **Exercice 1.** ***(Représentation des entiers en mémoire)***

Écrivez un programme qui affiche la représentation binaire en mémoire d'un entier. Pour cela, vous utiliserez les opérateurs C de manipulation de bits.

Par exemple, $25(= 16 + 8 + 1)$ *donnera l'affichage suivant :*

```
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 1
```

Testez votre programme sur les valeurs suivantes : 25, 7, 1024, -1024. *En regardant ces deux derniers résultat, que concluez-vous quant au mode de représentation des entiers négatifs en mémoire sur votre ordinateur ?*

▸ **Exercice 2.** ***(Permutation circulaire)***

Écrivez une fonction permettant d'effectuer une permutation circulaire à gauche d'un élément sur quatre éléments réels. Le tableau suivant illustre le comportement de la fonction sur un exemple :

Variables	a	b	c	d
Avant	*1.8*	*23.1*	*10.3*	*-4.0*
Après	*23.1*	*10.3*	*-4.0*	*1.8*

Écrivez ensuite une fonction permettant d'effectuer une permutation circulaire à gauche d'un élément sur un tableau de taille quelconque de réels.

Écrivez enfin une fonction permettant d'effectuer une permutation circulaire à gauche ou à droite d'un nombre quelconque d'éléments sur un tableau de taille quelconque de réels. Cette fonction prendra en argument un entier spécifiant le sens et le nombre de décalages à effectuer ; ce nombre sera positif pour une permutation à droite et négatif pour une permutation à gauche.

▸ **Exercice 3.** ***(Mise à jour de structures)***

Écrire deux fonctions permettant de saisir et d'afficher le contenu d'une variable de type `Personne` *défini ci-dessous :*

```
struct personne
{ char          Nom[MAX];
  unsigned int  Annee_Naissance;
  double        Credit;
};
typedef struct personne Personne;
```

Dans le cadre de la mise à jour des bases de données en vue du passage à l'euro et à l'an 2000, écrire deux fonctions modifiant une `Personne` *:*

- *la fonction* `PassageEuro` *qui mettra à jour la valeur du crédit initialement en francs (1 euro = 6,55957 francs).*
- *la fonction* `PassageAn2000` *modifiera le champ* `Annee_Naissance` *au cas il où vaudrait entre 0 et 99 (pour éviter une confusion de siècle).*

Dans toutes les fonctions vous utiliserez la notation `->` *dès que vous en aurez la possibilité.*

▸ **Exercice 4.** ***(Tableau de structures)***

Pour simuler la gestion d'une banque, nous allons maintenant considérer des tableaux de `Personne` *qui vont donc correspondre à ses clients.*

Écrivez les fonctions suivantes :

- `MoyenneCredit` *qui calcule la moyenne des valeurs des crédits de tous les clients,*
- `AfficheNomsAnneeDonnee` *qui affiche le nom des clients nés une année donnée.*
- `AfficheRiches` *qui affiche le nom des clients ayant sur leur compte plus d'une certaine valeur donnée.*

▸ **Exercice 5.** ***(Tableau de structures, suite et fin)***

Ayant décider de se débarrasser de ses clients mauvais payeurs, la banque veut supprimer ceux dont le compte est négatif. Afin de vous rendre complice des ces agissements inqualifiables, écrivez une fonction qui supprime du tableau les personnes répondant à ce critère.

Veillez à implémenter cette fonction de façon la plus efficace possible.