Travaux Pratiques d'algorithmique n°5

Cours d'Informatique de Deuxième Année

—DEUG Sciences S3-MI—

Manipulation de pointeurs

Dans cette séance, nous abordons la gestion des pointeurs sous divers aspects :

- quelques rappels de syntaxe ;
- l'allocation de la mémoire ;
- la manipulation des structures (opérateur `->`).

▸ **Exercice 1.** *On considère le programme suivant :*

```
#include <stdio.h>

void main(void)
{ int *ad1, *ad2, *ad3 ;
  int n = 10, p = 20 ;
  ad1 = &n ;
  ad2 = &p ;
  *ad1 = 3;
  *ad2 = *ad1 + *ad2 + p ;
  printf("n = %d, p =  %d\n",n,p);
  printf("*ad1 = %d, *ad2 =  %d\n",*ad1,*ad2);
  *ad3 = *ad2;
  printf("*ad2 = %d, *ad3 =  %d\n",*ad2,*ad3);
}
```

Ce programme est syntaxiquement correct mais produit un `core` *à l'exécution. Pourquoi ?*

▸ **Exercice 2.** ***(Permutation circulaire)*** *Écrire une fonction permettant d'effectuer une permutation circulaire sur quatre éléments réels. Le tableau suivant illustre le comportement de la fonction sur un exemple :*

Variables	*Avant*	*Après*
a	*3.2*	*1.3*
b	*1.3*	*-4.0*
c	*-4.0*	*2.8*
d	*2.8*	*3.2*

▸ **Exercice 3.** ***(Mise à jour de données)*** *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;
  float         Credit;
};

typedef struct personne Personne;
```

Dans le cadre de la mise à jour des bases de données informatiques en vue de l'introduction de la Monnaie Unique et du passage à l'an 2000 *:*

- *écrire une fonction* `void PassageEuro(???)` *qui met à jour la valeur du crédit (initialement en Francs) sur une variable de type* `Personne`*.*
  Indication : 1 Euro = 6.50 Francs ;
- *écrire une fonction* `void PassageAn2000(???)` *qui met à jour une variable de type* `Personne`*. En effet, dans le cas où le champ* `Annee_Naissance` *a une valeur comprise entre* 0 *et* 99*, il est nécessaire de le modifier pour éviter une confusion sur le siècle.*

▸ **Exercice 4.** ***(Triangle de Pascal)***

- *Construire à la main le triangle de Pascal de dimension* 4 *à l'aide des relations suivantes :*

$$\begin{cases} M_{1,1} & = & 1 \ , & \\ M_{i,j} & = & M_{i-1,j-1} + M_{i-1,j} & \text{pour } 0 < j \leq i \ , \\ & = & 0 & \textit{sinon.} \end{cases}$$

- *Écrire les fonctions* `AllocMat`*,* `TrianglePascal`*,* `AffichageTriangle` *et* `FreeMat` *pour respectivement allouer une matrice triangulaire, remplir le triangle, afficher le triangle et libérer l'espace mémoire précédemment alloué.*