

TD 12 : Pointeurs et tableaux

Exercice 1 : Pour chacun des programmes suivants, trouver ce qui va s'afficher et représenter le mapping mémoire :

- a.

```
short Tab[]={8,5};
short *pTab=&Tab;
printf ("%d ",*pTab++);
printf ("%d ",*pTab);
```
- b.

```
char Text[]={ 'H','e','l','l','o',0};
char *pText;
for (pText=Text ; *pText!=0;pText++)
    printf ("%c",*pText);
```
- c.

```
#define NbEtudiants    100

void EffaceTableau (double *Tableau,int Taille);

void main(void)
{
    double Notes[NbEtudiants];
    EffaceTableau(Notes,NbEtudiants);
    printf ("%2.2f",Notes[3]);
}

void EffaceTableau (double *pTableau,int Taille)
{
    int i;
    for (i=0;i<Taille;i++) *pTableau++=0;
}
```
- d.

```
short Tab[99];
short *pTab=Tab;
printf ("%d\n",sizeof(Tab));
printf ("%d\n",sizeof(pTab));
```

Exercice 2 :

- a. Ecrire le prototype d'une fonction AffTab qui reçoit en paramètre l'adresse d'un tableau d'entiers et le nombre d'éléments.
- b. Ecrire le code de cette fonction qui affiche tous les éléments du tableau en utilisant le formalisme tableaux et pointeurs.
- c. Compléter le programme suivant afin que le contenu du tableau soit affiché :

```
void main(void)
{
    int Tab[]={1,2,3,4,5,6,7,8,9};

}
```

Exercice 3 :

Des étudiants sont évalués sur deux contrôles. A la fin du semestre, on ne conserve que la meilleure des deux notes pour faire la moyenne.

```
float Controle1[6]={10.4, 20, 13.4, 18.4, 7.8, 5.6};  
float Controle2[6]={17.2, 11.7, 12.7, 16.7, 18.7, 15.4};  
float Meilleur[6];
```

a. Ecrire le prototype et le code d'une fonction `CompareNote` qui reçoit en paramètre les données suivantes :

- l'adresse du tableau avec les notes du premier contrôle
- l'adresse du tableau avec les notes du second contrôle
- l'adresse d'un tableau pour y stocker la meilleure note
- le nombre de notes dans les tableaux

Ecrire le code permettant de remplir le tableau contenant la meilleure note en utilisant le formalisme tableaux et pointeurs.

b. Ecrire l'appel de la fonction.

Exercice 4 :

a. Ecrire le prototype et le code d'une fonction `PositifTab` qui reçoit un pointeur sur un tableau d'entiers et sa taille en paramètres. Cette fonction supprime tous les éléments négatifs et renvoie la nouvelle taille du tableau en utilisant le formalisme tableaux et pointeurs.