

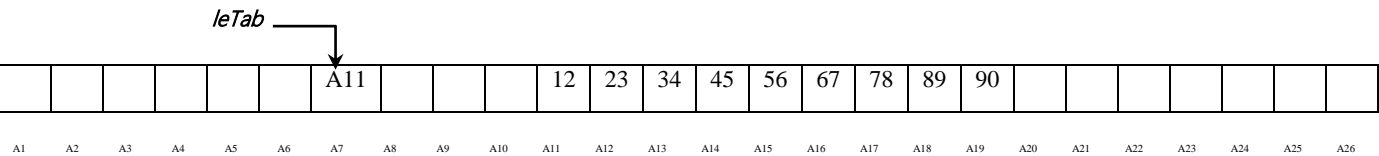
TD6 - Pointeurs

Tous les exercices de ce TD doivent utiliser la notation pointeur pour parcourir les tableaux.

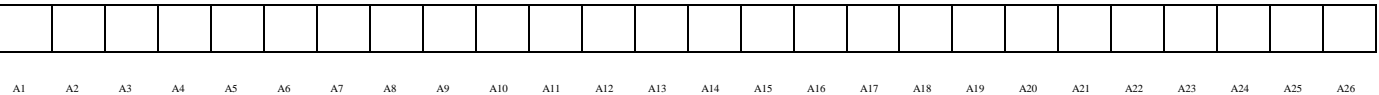
Exercice 1 :

Sans les programmer, qu'affichent les programmes suivants ?

```
void ExercicePointeurs1 ()
{
    int leTab[]={12,23,34,45,56,67,78,89,90};
    printf ("*leTab+2      = %d \n",*leTab+2);
    printf ("*(leTab+2)    = %d\n",*(leTab+2));
    printf ("%leTab       = %p\n",&leTab);
    printf ("*(&leTab[0]) = %d\n",*(&leTab[0]));
    printf ("%leTab+6     = %p\n",leTab+6);
    printf ("%*(leTab+6)  = %d\n",*(leTab+6));
}
```



```
void ExercicePointeurs2 (){
    long A, B, C, D;
    long *p1, *p2, *p3;
    long **pp;
    A = 10 ;
    B = 20 ;
    C = 30;
    D = 40;
    p1 = &A;
    p2 = &B;
    p3 = &B;
    *p1 = (*p1 + 1);
    *p3 = (*p2 + D);
    p3 = &C; *p3 = (*p2 + D);
    printf("A = %ld\nB = %ld\nC = %ld\nD = %ld\n", A, B, C, D);
    pp = &p3;
    printf("***pp = %ld\n", **pp);
}
```



Exercice 2 :

On dispose du programme principal et du prototype suivant :

```
void calculPuissance (int * nb, int puiss) ;

int main ()
{
    int a = 5, p = 2 ;
    calculPuissance (&a, p) ;
    printf ("Le resultat de la puissance vaut %d\n", a) ;
}
```

Écrivez la procédure **calculPuissance** qui modifie la valeur **nb** pour l'élever à la puissance **puiss**.

Une fois que vous aurez écrit votre procédure, utilisez les cases mémoires ci-dessous pour placer toutes les variables de votre programme **au moment de l'appel de la procédure**.

[illegible]

Exercice 3 :

Écrivez la procédure **auCarré** qui prend un nombre entier et qui l'élève au carré. Votre nombre doit être passé en paramètre sous la forme d'un pointeur. Écrivez également le programme principal qui :

- Déclare un entier N et l'initialise avec une valeur
- Appelle la procédure auCarré
- Affiche la nouvelle valeur de N.

Une fois que vous aurez écrit votre procédure, utilisez les cases mémoires ci-dessous pour placer toutes les variables de votre programme **au moment de l'appel de la procédure**.

[illegible]

Exercice 4 :

Écrivez la procédure **Echange** qui échange les valeurs de deux nombres réels passés en paramètre par référence, donc à l'aide de pointeurs. Écrivez également le programme principal qui :

- Déclare et initialise 2 réels
- Affiche leurs valeurs
- Appelle la procédure échange
- Affiche les nouvelles valeurs des 2 nombres.

Une fois que vous aurez écrit votre procédure, utilisez les cases mémoires ci-dessous pour placer toutes les variables de votre programme **au moment de l'appel de la procédure**.

[illegible]

Exercice 5 :

Écrivez la procédure **afficheTableau** qui affiche un tableau d'entiers de taille N. Vous devez utiliser la notation pointeur pour passer le tableau en paramètres (int *t) et pour parcourir le tableau. Écrivez un programme qui :

- Déclare un tableau de taille 10 en l'initialisant à la compilation (`int t[10] = {...}`)
- Déclare un pointeur sur un entier qui est initialisé avec la valeur de t (`int *p = t`) ;
- Appelle la procédure `afficheTableau` en lui passant p comme paramètre pour le tableau.

Une fois que vous aurez écrit votre procédure, utilisez les cases mémoires ci-dessous pour placer toutes les variables de votre programme au moment de l'appel de la procédure.

[illegible]

Exercice 6 :

Écrivez la **procédure** qui ajoute un entier à la fin d'un tableau d'entiers. Votre procédure doit vérifier qu'il est possible de rajouter un élément dans le tableau (que le tableau n'est pas plein).

Écrivez également le programme principal qui :

- Déclare un tableau de taille MAXTAB (MAXTAB = 20) en l'initialisant à la compilation avec 10 valeurs ($\text{int } t[\text{MAXTAB}] = \{\dots\}$)
- Déclare une variable taille qui contient le nombre de valeurs contenues dans le tableau (soit 10)
- Déclare un pointeur sur un entier qui est initialisé avec la valeur de t ($\text{int } *p = t$)
- Affiche le tableau en utilisant la variable p
- Demande une valeur à l'utilisateur
- Appelle la procédure qui rajoute cet élément dans le tableau en lui passant p comme paramètre pour le tableau
- Affiche de nouveau le tableau
- Teste avec assert si le résultat obtenu est correct.

Une fois que vous aurez écrit votre procédure, utilisez les cases mémoires ci-dessous pour placer toutes les variables de votre programme **au moment de l'appel de la procédure**.

[illegible]

Exercice 7 :

Écrivez la procédure **MinMax** qui parcourt un tableau d'entiers et qui calcule les valeurs minimum et maximum contenues dans le tableau. Vous devez utiliser la notation pointeur pour passer votre tableau en paramètre et pour le parcourir. Écrivez également le programme principal qui :

- Déclare un tableau de taille MAXTAB (MAXTAB = 20) en l'initialisant à la compilation avec 10 valeurs ($\text{int } t[\text{MAXTAB}] = \{\dots\}$)
- Déclare une variable taille qui contient le nombre de valeurs contenues dans le tableau (soit 10)
- Déclare un pointeur sur un entier qui est initialisé avec la valeur de t ($\text{int } *p = t$) ;
- Déclare deux variables entières qui contiendront le minimum et le maximum
- Appelle la procédure **MinMax** en lui passant p comme paramètre pour le tableau
- Affiche les valeurs du minimum et du maximum
- Teste avec assert si le résultat obtenu est correct.

Une fois que vous aurez écrit votre procédure, utilisez les cases mémoires ci-dessous pour placer toutes les variables de votre programme **au moment de l'appel de la procédure**.

[illegible]

Exercice 8 :

Écrivez la procédure **nbOccurrences** qui compte le nombre de fois qu'une valeur entière est présente dans un tableau d'entiers. Vous devez utiliser la notation pointeur pour passer votre tableau en paramètre et pour le parcourir. Écrivez également le programme principal qui :

- Déclare un tableau de taille MAXTAB (MAXTAB = 20) en l'initialisant à la compilation avec 10 valeurs (int t[MAXTAB] = {...})
- Déclare une variable taille qui contient le nombre de valeurs contenues dans le tableau (soit 10)
- Déclare un pointeur sur un entier qui est initialisé avec la valeur de t (int *p = t)

- Affiche le tableau
- Demande une valeur à l'utilisateur
- Appelle la procédure **nbOccurrences** en lui passant p comme paramètre pour le tableau
- Affiche le résultat obtenu
- Teste avec assert si le résultat obtenu est correct.

Une fois que vous aurez écrit votre procédure, utilisez les cases mémoires ci-dessous pour placer toutes les variables de votre programme **au moment de l'appel de la procédure**.

A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	A16	A17	A18	A19	A20	A21	A22	A23	A24	A25	A26

Exercice 9 :

Écrire une procédure **saisirTableau** permettant de saisir N valeurs entières dans un tableau en utilisant la notation pointeur. Pour la saisie des valeurs on utilisera la fonction qui génère un nombre aléatoire que nous avons déjà vu dans un des TD précédents.

Écrivez le programme principal qui

- Déclare un tableau de taille MAXTAB (MAXTAB = 20) en l'initialisant à la compilation avec 10 valeurs (`int t[MAXTAB] = {...}`)
- Déclare une variable `taille` qui contient le nombre de valeurs contenues dans le tableau (soit 10)
- Déclare un pointeur sur un entier qui est initialisé avec la valeur de `t` (`int *p = t`) ;
- Appelle les procédures **afficheTableau** et **saisirTableau** en lui passant `p` comme paramètre pour le tableau

Une fois que vous aurez écrit votre procédure, utilisez les cases mémoires ci-dessous pour placer toutes les variables de votre programme **au moment de l'appel de la procédure**.

[illegible]

Exercice 10 :

Écrire une procédure **calculSomme** permettant de renvoyer la somme des nombres positifs ainsi que la somme des nombres négatifs d'un tableau d'entiers de N éléments en utilisant la notation pointeur. Écrivez également le programme principal qui :

