

TP N° 9 : Les pointeurs

Objectifs

Apprendre la manipulation de l'un des principaux aspects de programmations dans le langage C: les pointeurs ainsi que la forte liaison tableaux- pointeurs.

Exercice 1

Complétez-le pour chaque instruction du programme ci-dessus

	<u>A</u>	<u>B</u>	<u>C</u>	<u>P1</u>	<u>P2</u>
<code>int A = 1, B = 2, C = 3;</code> <code>int *P1, *P2;</code>	1	2	3	/	/
<code>P1=&A;</code>	1	2	3	&A	/
<code>P2=&C;</code>					
<code>*P1=(*P2)++ ;</code>					
<code>P1=P2 ;</code>					
<code>P2=&B ;</code>					
<code>*P1--*P2 ;</code>					
<code>++*P2 ;</code>					
<code>*P1*=*P2 ;</code>					
<code>A=*P2**P1 ;</code>					
<code>P1=&A ;</code>					
<code>*P2/=*P1 ;</code>					

Exercice 2

Soit P un pointeur qui 'pointe' sur un tableau A:

```
int A[] = {12, 23, 34, 45, 56, 67, 78, 89, 90};
int *P;
P = A;
```

Quelles valeurs ou adresses fournissent ces expressions:

- a) `*P+2`
- b) `*(P+2)`
- c) `&P+1`
- d) `&A[4]-3`
- e) `A+3`
- f) `&A[7]-P`
- g) `P+(*P-10)`

h) * (P+* (P+8) -A[7])

Exercice 3

Ecrire un programme qui lit deux tableaux A et B et leurs dimensions N et M au clavier et qui ajoute les éléments de B à la fin de A. Utiliser le formalisme pointeur à chaque fois que cela est possible.

Exercice 4

Ecrire un programme qui lit un entier X et un tableau A de type int au clavier et élimine toutes les occurrences de X dans A en tassant les éléments restants. Afficher le tableau A.

Le programme utilisera les pointeurs P1 et P2 pour parcourir le tableau.

Exercice 5

Ecrire un programme qui lit la dimension N d'un tableau **T** du type int (dimension maximale: 100), remplit le tableau par des valeurs entrées au clavier et affiche le tableau.

Copiez ensuite toutes les composantes strictement positives dans un deuxième tableau **TPOS** et toutes les valeurs strictement négatives dans un troisième tableau **TNEG**. Afficher les tableaux **TPOS** et **TNEG**.

NB : Utiliser des pointeurs.

Exercice 6

Ecrire un programme qui lit une chaîne de caractères CH et détermine la longueur de la chaîne à l'aide d'un pointeur PH. Le programme n'utilisera pas de variables numériques.

Exercice 7

Ecrire un programme qui lit 5 mots (une phrase de longueur maximale de 50 caractères) et les mémorise dans un tableau de chaînes de caractères TABCH. Inverser l'ordre des caractères à l'intérieur des 5 mots à l'aide de deux pointeurs P1 et P2. Afficher les mots.

Exercice 8

Soient deux tableaux d'entiers T1 (maximum 50) et T2 (max 100). Ecrire un programme C qui permet de vérifier que T2 existe dans T1.

Si T2 existe dans T1, alors supprimer les éléments de T2 dans T1.

NB : Utiliser le formalisme pointeur.

Exemple :

15	8	0	2	9
----	---	---	---	---

T2

T1

1	11	15	8	0	9	13	23	100
---	----	----	---	---	---	----	----	-----

⇒ T2 existe dans T1

⇒ T1

1	11	23	100
---	----	----	-----

Exercice 9

Ecrire un programme qui lit deux matrices A et B de dimensions N et M respectivement M et P au clavier et qui effectue la multiplication des deux matrices. Le résultat de la multiplication sera affecté à la matrice C, qui sera ensuite affichée. Utiliser le formalisme pointeur à chaque fois que cela est possible.

Exercice 10

Ecrire une fonction qui permet de rechercher dans un tableau d'entiers **tab** une valeur **A**.

```
void ChercherVal (int tab[], int n, int A, int *pos, int *nbOcc);
```

Dans **pos**, la fonction sauvegarde l'indice de la dernière apparition et -1 si la valeur n'a pas été trouvée. Dans **nbOcc**, elle sauvegarde le nombre d'occurrence de **A** dans **tab**.

Exercice 11

1- Écrire une fonction est **EstVoyelle** qui prend en paramètre un caractère et retourne 1 si ce caractère est une voyelle et 0 si non.

2- Écrire une fonction **NbVoyelle**:

```
void NbVoyelles(char ch[], int *v, int *s)
```

À travers **v**, la fonction sauvegarde le nombre de voyelles dans la chaîne **ch**. Dans **s**, la fonction sauvegarde le nombre de consonnes.

Exercice 12

Ecrire une fonction qui permet de rechercher dans un tableau d'entiers **tab** de taille **n** les deux plus grandes valeurs :

```
void deux_max (int T[], int n, int *max1, int *max2);
```

Exemple: si T=[3, 1, 5, 5, 3, 6, 2, 6] alors **max1=6** et **max2 = 5**.

Exercice 13

Ecrire une fonction ***SupprimerC*** qui à partir d'un texte TXT et d'un caractère C enlève tous les caractères C du texte TXT. Les modifications se feront dans la même variable TXT.

```
TXT="Cette ligne contient quelques lettres e"  
C='e'  
Alors TXT="Ctt lign contint qulqus lttrs"
```