

Séance 2: Pointeurs et Tableaux

Les pointeurs

Un pointeur est une variable, il est destiné à contenir une adresse mémoire, c'est à dire une valeur identifiant un emplacement en mémoire. Pour différencier un pointeur d'une variable ordinaire, on fait précéder son nom du signe `**` lors de sa déclaration.

1. Type `*NomPointeur` déclare un **pointeur** `NomPointeur` qui peut recevoir des adresses de variables du type `Type`.

2. L'opérateur `&` désigne l'**adresse d'une variable** :

`&NomVariable` fournit l'adresse de la variable `NomVariable`.

3. L'opérateur `*` désigne le **contenu d'une adresse**.

`*NomPointeur` désigne le contenu de l'adresse référencée par le pointeur `NomPointeur`.

Exemple :

```
int a;           /* a est une variable de type int */
int *p;         /* p est un pointeur sur un entier */
p=&a;           /* p contient l'adresse de a */
a=10;           /* a vaut 10 */
*p=20;         /* on met 20 à l'adresse désignée par p */
               /* conséquence : a vaut 20 maintenant */
```

Exercice 1. Passage de paramètres à une fonction

Ecrire une procédure *echanger* (*int a*, *int b*) qui doit échanger les deux valeurs contenues dans les variables *a* et *b*.

Tester cette fonction dans la fonction *main()* : déclarer deux entiers *x* et *y* dans le *main()*, appeler la fonction *echanger* pour échanger leurs valeurs. **Ensuite**, afficher le contenu des deux variables : cela fonctionne-t-il ? Pourquoi ? Que faudrait-il faire pour que cela fonctionne ? Justifier.

Réécrire la procédure *echanger* pour qu'elle fonctionne, et modifier la fonction *main()* en conséquence.

Pointeurs et tableaux :

Le nom d'un tableau est considéré comme un pointeur sur son premier élément (Dans la réalité le nom d'un tableau comme n'importe quelle variable d'ailleurs référence un emplacement en mémoire).

Il y a équivalence entre `&tab[0]` et `tab` (adresse de `tab[0]`)

Il y a équivalence entre `tab[0]` et `*tab`

Il y a équivalence entre `tab[i]` et `*(tab+i)`

Si on incrémente le pointeur `tab` il ne contiendra pas l'adresse du premier élément du tableau + 1, mais l'adresse de l'élément suivant. La valeur de l'adresse qu'il contient sera donc incrémentée de la taille du type qu'il référence.

Exercice 2. Manipulations de base sur les tableaux

Ecrire un **programme** dans lequel :

- Vous déclarez un tableau de 10 entiers
- Vous saisissez 10 entiers au clavier en les rangeant dans le tableau
- Vous affichez le contenu du tableau en utilisant une boucle `for` et `printf`

Entraînez-vous à utiliser les différentes écritures équivalentes.

Exercice 3. Tableau d'entiers

1. Ecrire une **procédure** `afficheTableau` qui prend en arguments un tableau d'entiers (un pointeur vers un entier), et sa taille, puis affiche à l'écran son contenu, par exemple sous la forme : Entier 1 =

:

Entier 10 =

Vous testerez cette procédure dans le programme de la question 2.

2. Ecrire un **programme** dans lequel :
 - Vous déclarez un tableau de 10 entiers
 - Vous saisissez 10 entiers au clavier en les rangeant dans le tableau
 - Vous affichez le contenu du tableau en utilisant la procédure `afficheTableau`
 - Vous recherchez et affichez la **valeur** et la **position** du maximum et du minimum des entiers contenus dans le tableau (dans le cas de plusieurs maxima ou minima, on retiendra la **première occurrence** rencontrée).

Exemple : si on saisit les valeurs 4 3 5 6 8 4 3 8 6 5, on obtient

- le maximum est 8 et se trouve dans la case 4
- le minimum est 3 et se trouve dans la case 1

3. Au lieu de faire la recherche des extrema dans le `main`, écrire une **procédure** `maxmin` qui prend en arguments un tableau d'entiers (un pointeur vers un entier) et sa taille, et qui affiche à l'écran la **valeur** et la **position** du maximum et du minimum du tableau (dans le cas de plusieurs maxima ou minima, on retiendra cette fois-ci la **dernière occurrence** rencontrée). Tester cette procédure en l'appelant dans le `main`.

Exemple : si on saisit les valeurs 4 3 5 6 8 4 3 8 6 5, on obtient

- le maximum est 8 et se trouve dans la case 7
- le minimum est 3 et se trouve dans la case 6