Chapitre 9: Les Pointeurs

I. Introduction

Jusqu'ici, nous avons utilisé des variables dites « **statiques** ».
Une ***variable statique*** est caractérisée par les propriétés suivantes :
- elle est déclarée en tête du bloc où elle est utilisée
- elle occupe un espace mémoire dont la taille est fixée dès le début pour qu'on y place ses valeurs
- l'accès à la valeur se fait par le nom de la variable.

Au contraire, une ***variable dynamique*** est caractérisée par les propriétés suivantes :
- elle peut être créée et détruite au cours de l'exécution du bloc dans lequel elle est déclarée
- l'espace mémoire rendu libre peut être récupéré
- l'accès à la valeur se fait par un **pointeur**.

II. Pointeurs

Un **pointeur** P est une variable statique dont les valeurs sont des adresses.
Une variable dynamique pointée par P sera notée **P^**. Cette variable dynamique est appelée **le contenue de la variable pointée par le pointeur P**

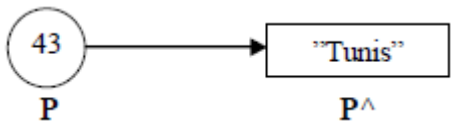
Selon le type de donnée contenu à l'adresse en question, on aura un **pointeur d'entier, de réel, de caractère**, ou de tout autre type. En accédant à cette adresse, on peut accéder indirectement à la variable et donc la modifier.

II.1 Déclaration de pointeur

Pour déclarer un variable pointeur on utilise l'opérateur ^ placé avant le type du pointeur**.**
Un pointeur sur entier est déclaré par **p : ^ entier**
Un pointeur sur réel **p : ^ réel**
Un pointeur sur caractère **p : ^ caractère**

Pointeur sur pointeur de type double **tab : ^^ double**
La variable *tab* est un pointeur pointant sur un pointeur qui pointe sur un flottant double !

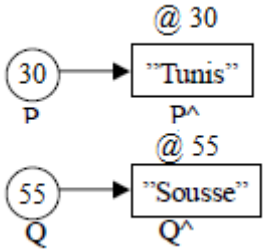
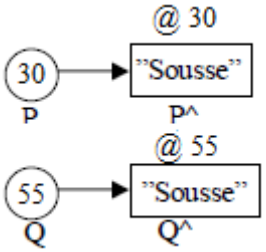
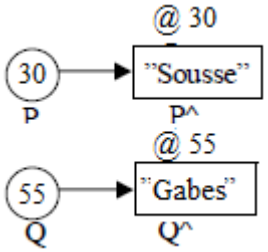
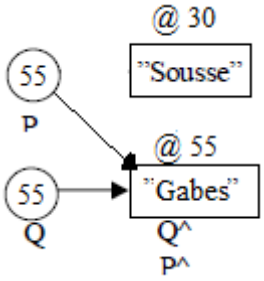
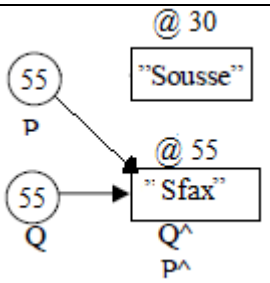
On peut déclarer un type de pointeur.
Exemple :

Algorithme Pteur **Types** Pointeur = ^Chaîne **Variables** P : Pointeur **Début** Allouer(P) *(* création du pointeur P *)* P^← ”Tunis” **Fin.**	

La variable pointeur **P** a pour valeur **43**. Elle pointe sur l'espace mémoire **P^** d'adresse **43** et dont le contenu est la chaîne **”Tunis”.**

II.2 Opérations sur les pointeurs

Il ne faut pas confondre les opérations sur les pointeurs avec les opérations sur les variables pointées

Algorithme Pteur **Types** Pointeur = ^Chaîne **Variables** P : Pointeur Q : Pointeur **Début**	
Allouer(P) *(* création du pointeur P *)* P^← ”Tunis” (*la variable pointée par P reçoit la valeur ”Tunis” *) Allouer(Q) *(* création du pointeur Q *)* Q^← ”Sousse” (*la variable pointée par Q reçoit la valeur ”Sousse” *)	
P^← Q^ (* le contenue de la variable pointée par P (noté P^) reçoit le contenue de la variable pointée par Q *)	
Q^←”Gabes” (* le contenue de la variable pointée par Q reçoit la chaîne ”Gabes” *) **(*Attention : le (* le contenue de la variable pointée par P ne change pas*)**	
P← Q (* P sera égale à Q donc *) (* P et Q pointent sur la même @ mémoire *) (*la variable pointée par P est celle pointée par Q*)	
P^← ”Sfax” (*équivalent à Q^← ”Sfax” *) (* le contenue de la variable pointée par Q (noté Q^) sera Sfax car P et Q pointent sur la même variable *)	

III. Pointeurs et arguments de fonctions

En raison de l'appel par valeur, une fonction ne peut pas modifier ses arguments. L'utilisation de pointeurs permet de tourner la difficulté.
Soit par exemple la fonction echange (x,y) qui est **censée** échanger ses arguments :

```
Procedure echange (x : entier, y :entier) /* INCORRECT */
Var
        aux :entier
debut
        aux← y
        y←x
        x ← aux
fin proc
```

```
Programme Exemple1
Var
        i :entier
        j :entier
début
        i←3
        j←4
        echange(i,j)
        écrire ("i=",i) /*affichei= 3*/
        écrire ("j=",j)/*affiche j=4*/
Fin
```

L'appel de `echange(i,j)` avec i et j entiers n'aura aucun effet car ces paramètres sont passés par valeur et ne sont pas modifiés en dehors de `echange()`.

```
Procedure echange (x : ^entier, y :^entier) /* CORRECT */
Var
        aux :entier
début
        aux← y^
        y^ ←x^
        x^ ← aux
fin proc
```

```
Programme Exemple2
Var     i :^entier
        j :^entier
début
        Allouer(i) ,  Allouer(j)
        i^←3
        j^←4
        echange (i,j)
        écrire ("i=",i^) /*affichei= 4*/
        écrire ("j=",j^)/*affiche j=3*/
Fin
```

Maintenant l'appel de `echange(i,j)` modifie les variables i^ et j^ car les adresses des paramètres sont passées à la fonction (on parlera de passage par VARIABLE ou par ADRESSE).