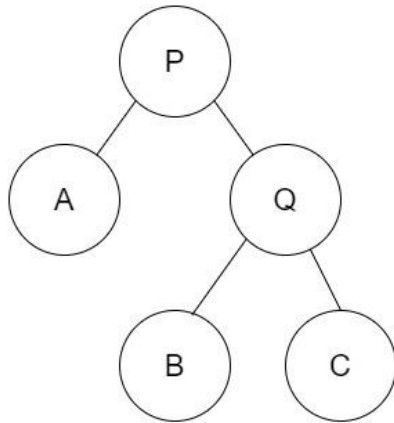
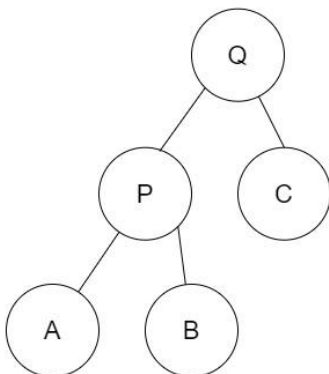


**Exercice 1 : Rotation gauche**



Réaliser la rotation gauche sur le nœud P



**Algorithme**

Arbre : A, B ;

Début

B <- A -> FD ;

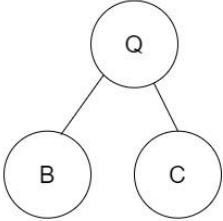
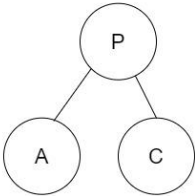
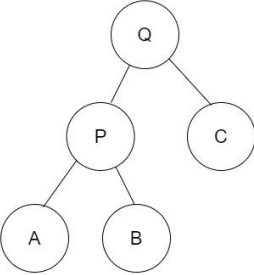
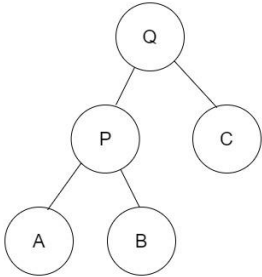
A -> FD <- B -> FG ;

B -> FG <- A ;

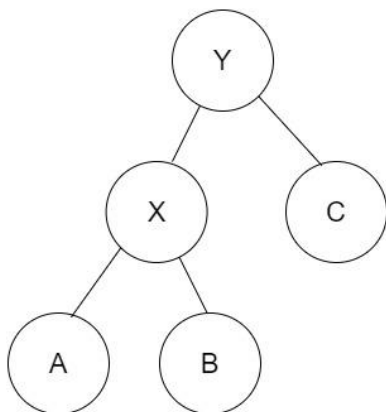
A <- B ;

Fin

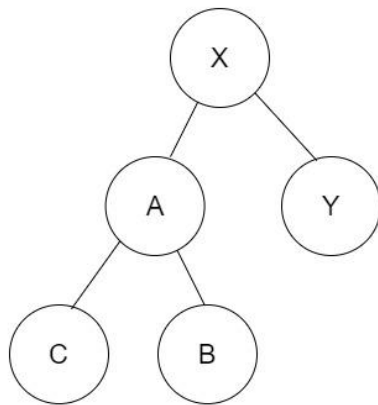
### Trace d'exécution :

<b>Instruction 1 :</b> B <- A -> FD ;	<b>Instruction 2 :</b> A -> FD <- B-> FG ;	<b>Instruction 3 :</b> B -> FG <- A ;	<b>Instruction 4 :</b> A <- B ;
			

### Exercice 2 : : Rotation droite



Réaliser la rotation gauche sur le nœud Y



### **Algorithme**

Arbre : A, B ;

Début

B <- A -> FG ;

A -> FG <- B -> FD ;

B -> FD <- A ;

A <- B ;

Fin

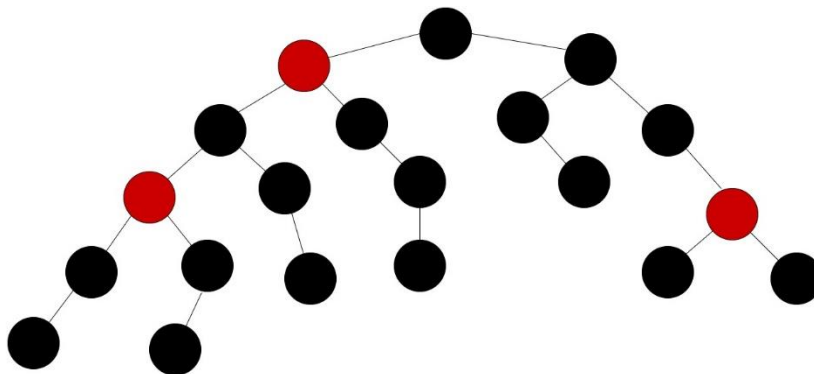
### **Trace d'exécution :**

<b><u>Instruction 1 :</u></b> B <- A -> FG ;	<b><u>Instruction 2 :</u></b> A -> FG <- B -> FD ;	<b><u>Instruction 3 :</u></b> B -> FD <- A ;	<b><u>Instruction 4 :</u></b> A <- B ;
<pre> graph TD     X((X)) --- A((A))     X --- Y((Y))   </pre>	<pre> graph TD     X((X)) --- A((A))     X --- B((B))   </pre>	<pre> graph TD     X((X)) --- A((A))     X --- Y((Y))     Y --- B((B))     Y --- C((C))   </pre>	<pre> graph TD     X((X)) --- A((A))     X --- Y((Y))     Y --- B((B))     Y --- C((C))   </pre>

### Exercice 3 : Arbre rouge est noir

- 1) Quels sont les propriétés que doit vérifier un arbre R&N
  - Un nœud est soit rouge soit noir ;
  - La racine est noire ;
  - Les enfants d'un nœud rouge sont noirs ;
  - Tous les nœuds ont 2 enfants. Ce sont d'autres nœuds ou des feuilles **NIL**, qui ne possèdent pas de valeur et qui sont les seuls nœuds sans enfants. Leur couleur est toujours **noire** et rentre donc en compte lors du calcul de la hauteur noire.
  - Le chemin de la racine à n'importe quelle feuille (**NIL**) contient le même nombre de nœuds noirs. On peut appeler ce nombre de nœuds noirs la **hauteur noire**.
  
- 2) Donner des illustrations du plus **grand** nombre possible de nœuds dans un arbres R&N de hauteur noire k

Quand on intercale des nœuds rouges et noirs.



- 3) Donner des illustrations du plus **petit** nombre possible de nœuds dans un arbres R&N de hauteur noire k

Quand tous les nœuds sont noir.

4) Peut-on remplacer 1 nœud rouge par 1 nœud noir ?

Non sauf si le rouge et le noir sont sur le même niveau.

5) Peut-on remplacer 2 nœuds rouge par 2 nœuds noir ? Généraliser

Oui si les deux rouges sont sur le même niveau et sur la hauteur 1 (la hauteur de l'arbre augmente de 1)

6) Donner un exemple d'un arbre R&N de hauteur noir = 3

7) Écrire un algorithme de reconnaissance d'un arbre R&N

### **Algorithme reconnaissance d'un arbre R&N**

Arbre :  $a$  ;

Entier :  $h, h_n$

Début

$h = 0$  ;

$h_n = -1$  ;

Si ( $a \rightarrow \text{couleur} = R$ ) ET ( $a \rightarrow \text{fg} = \text{NULL}$  OU  $a \rightarrow \text{fd} = \text{NULL}$  OU  $a \rightarrow \text{fg} \rightarrow \text{couleur} = R$ ) alors :

Retourner faux ;

Sinon si ( $a \rightarrow \text{fg} = \text{NULL}$ ) ET ( $a \rightarrow \text{fd} \neq \text{NULL}$  OU  $a \rightarrow \text{fd} = \text{NULL}$ ) ET ( $a \rightarrow \text{fg} \neq \text{NULL}$ ) alors :

Retourner faux ;

Sinon

Si ( $a \rightarrow \text{couleur} = N$ ) alors :

$h \leftarrow h+1$  ;

Fin si

Si ( $a \rightarrow \text{fg} = \text{NULL}$ ) ET ( $a \rightarrow \text{fd} = \text{NULL}$ ) alors :

Si ( $h_n = -1$ ) alors :

$h_n \leftarrow \text{hauteur}$  ;

Retourner vrai ;

Fin si

Sinon si ( $h \neq h_n$ ) alors :

Retourner faux ;

Fin si

Sinon