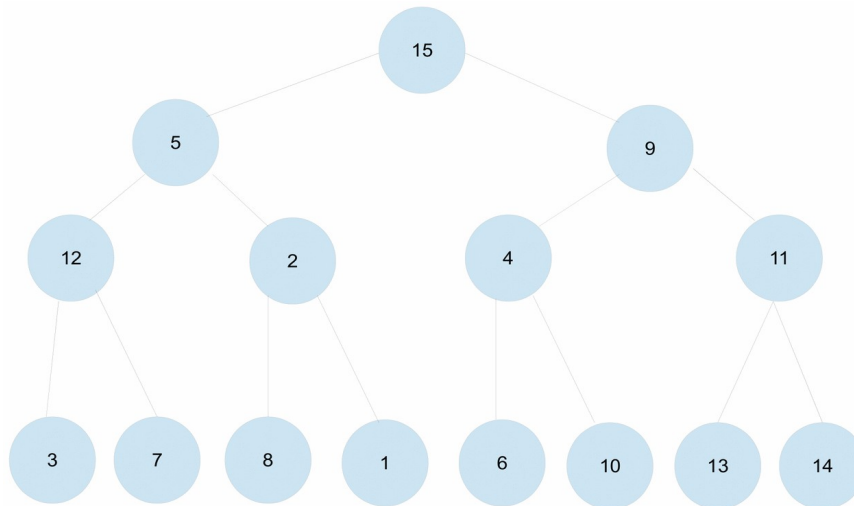


Un arbre est un graphe connexe où le nombre d'arêtes est égal au nombre de points diminués de 1

Un arbre dégénéré (appelé aussi filiforme) est un arbre dont les noeuds ne possède qu'un et un seul fils.

Exercice 13 :

1)

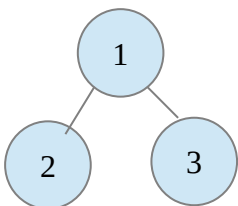


2) parcours infixe : 3 12 7 5 8 2 1 15 6 4 10 9 13 11 14, l'ordre infixe : on liste chaque sommet ayant un fils gauche la seconde fois qu'on le voit et chaque sommet sans fils gauche la première fois qu'on le voit.

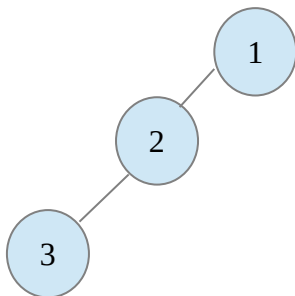
3) parcours préfixe : 15 5 12 3 7 2 8 1 9 4 6 10 11 13 14, l'ordre préfixe : on liste chaque sommet la première fois qu'on le rencontre dans la balade.

parcours postfixe : 3 7 12 8 1 2 5 6 10 4 13 14 11 9 15, l'ordre postfixe : on liste chaque sommet la dernière fois qu'on le rencontre.

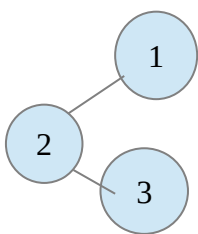
4) 1,2,3 => parcours préfixe



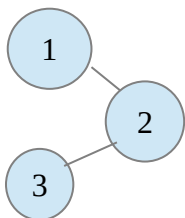
=> 1 2 3 en préfixe pour tout les cas ;
infixe : 2 1 3 postfixe : 2 3 1



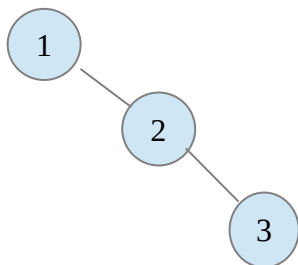
infixe : 3 2 1 postfixe : 3 2 1



infixe : 2 3 1 postfixe : 3 2 1



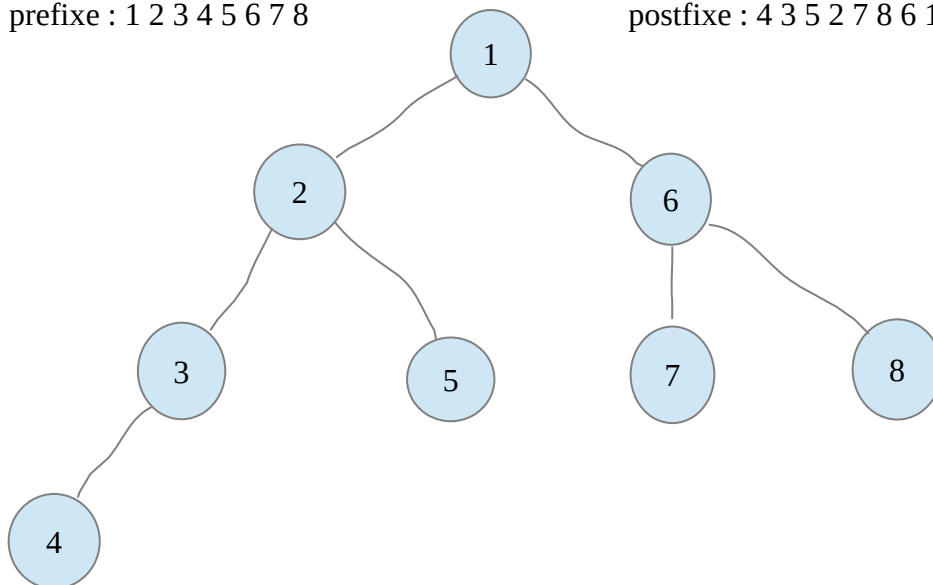
infixe : 1 3 2 postfixe : 3 2 1



infixe : 1 2 3 postfixe : 3 2 1

5) prefixe : 1 2 3 4 5 6 7 8

postfixe : 4 3 5 2 7 8 6 1



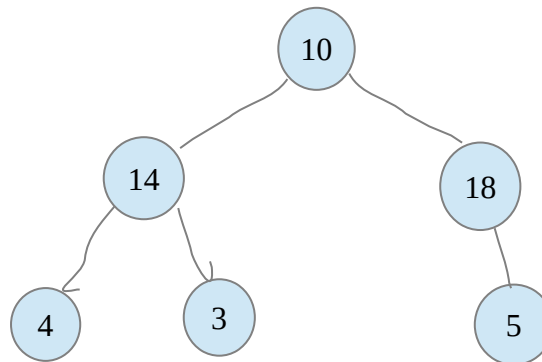
6) Non, il en existe pas

7)

Exercice 14 :

```
Enregistrement Noeud{  
    Val : entier  
    Sag :  $\hat{=}$  Noeud  
    Sad :  $\hat{=}$  Noeud  
}
```

Nombre nœuds d'un arbre :



Nombre noeuds(r:Noeud):entier

Début

Si r=NIL alors

retourner 0

Sinon

retourner (1+NombreNoeuds(r,sag)+NombreNoeuds(r,sad))

Fin si

Fin

Unfils(r:Noeud):entier

Début

Si r est vide alors retourner 0

Sinon Si (r.sag est vide et r.sad est vide)

retourner 0

Sinon Si (r.sag est vide et r.sad!= NIL)

retourner 1+Unfils(r.sad)

Sinon Si (r.sag!=NIL et r.sad=NIL)

retourner 1+Unfils(r.sag)

Sinon

retourner 0+Unfils(r.sag)+Unfils(r.sad)

Fin si

Fin

Feuilles(r:Noeud):entier

Début

Si r est vide retourner 0

Sinon Si r .sag est vide ET r.sad est vide

retourner 1

```

        Sinon
            retourner 0+Feuilles(r.sag)+Feuilles(r.sad)
        Fin si
    Fin

Descendants
    Enregistrement Noeud{
        Val:entier
        NbDesc:entier
        sag: ^ Noeud
        sad : ^ Noeud
    }

Descendants(r:Noeud):entier
Début
    Si r est vide alors retourner 0
    Sinon Si r.sag est vide et r.sad est vide alors
        retourner 0
    Sinon Si r.sag!=vide et r.sad!= vide alors
        retourner (1+Descendants(r.sag)+1+Descendants(r.sad))
    Sinon Si r.sag!= vide et r.sad=vide alors
        retourner (1+Descendants(r.sag))
    Sinon Si r.sag=vide et r.sad!=vide alors
        retourner (1+Descendants(r.sad))
    Fin si
Fin

```