

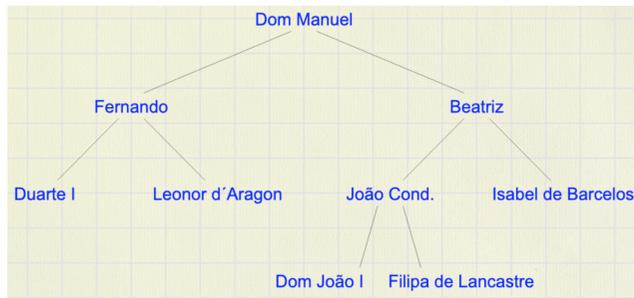
Algorithmes sur les arbres – exercices

1. HAUTEUR D'UN ARBRE

La hauteur d'un arbre correspond au *nombre de niveaux* de cet arbre. Après avoir revu le fonctionnement de la fonction qui compte les nœuds d'un arbre, écrire la fonction récursive `hauteur` qui calcule la hauteur d'un arbre.

2. LES AVIS

Dom Manuel fut l'avant dernier roi de la dynastie des *Avis* qui régna durant la prestigieuse période des Grandes découvertes. Sa généalogie simplifiée est représentée par l'arbre binaire :



- Créer l'instance `arbre1` de la classe `Node` qui permet d'implémenter cet arbre puis affichez-le dans la console.
- Affichez le nom d'une arrière-grand-mère maternelle de *Dom manuel*.
- Affichez la généalogie de *Beatriz de Portugal*, Duchesse de Viséu.
- Affichez le nombre de générations représentées par cet arbre.

3. UN ARBRE EN DEBRANCHE

On crée un nouvel arbre binaire à l'aide du code ci-dessous :

```
arbre2=Node(1)
arbre2.gauche=Node(2)
arbre2.gauche.gauche=Node(3)
arbre2.gauche.droit=Node(4)
arbre2.gauche.droit.gauche=Node(5)
arbre2.gauche.droit.droit=Node(6)
arbre2.droit=Node(7)
arbre2.droit.gauche=Node(8)
arbre2.droit.droit=Node(9)
```

- Dessinez une représentation de `arbre2`.
- Donner le résultat obtenu par son parcours en largeur, puis par ses parcours en profondeur (préfixé, infixé et postfixé).

4. RETOUR SUR LA RPN

Dans notre chapitre sur les piles, nous avons évoqué la *notation polonaise inversée* (RPN) qui permet d'évaluer une expression arithmétique sans ambiguïté vis-à-vis des priorités. Dans le fichier Python, vous trouverez la fonction `rpn` qui permet d'évaluer une expression arithmétique en *notation polonaise inversée*. Elle prend en paramètre une liste contenant l'expression arithmétique.

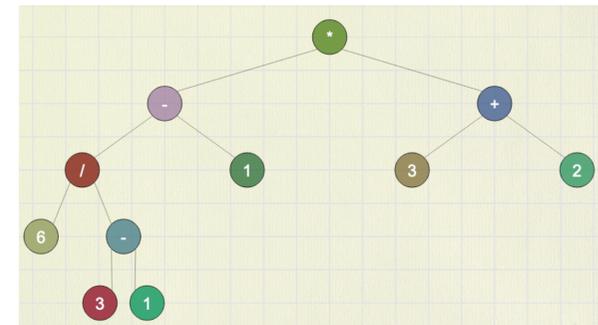
L'expression en RPN est dite *postfixée*. Nous allons comprendre pourquoi.

- Que va retourner la fonction `rpn` en passant la variable `expression` en paramètre ?
`expression=['6','2','/','1','2','+','*']`
Faites le test avec le programme Python.

On peut stocker une expression arithmétique dans un arbre binaire dans lequel chaque feuille est un *opérande*. Les autres nœuds sont des *opérateurs*.

Le parcours postfixé de cet arbre donne l'expression à évaluer en RPN.

Prenons l'exemple de cet arbre :



- Construire cet arbre (`arbre3`) à l'aide de la classe `Node`. Vérifiez qu'il est correct dans la console.
- Créer la fonction `postfixe` qui remplit récursivement une liste initialement vide avec tous les éléments de l'arbre dans l'ordre postfixé. Testez votre fonction.
- Évaluez enfin cette expression à l'aide la fonction `rpn`.

5. CONVERSION EN ITERATIF

Tout algorithme *récurif* peut se traduire par un algorithme *itératif* équivalent. L'objectif de cet exercice est de créer la fonction `prefixeIter` qui affiche le parcours préfixé d'un arbre binaire.

Écrire le code de cette fonction en utilisant la structure de donnée adaptée. Vérifiez son fonctionnement avec `arbre1` et `arbre2`.