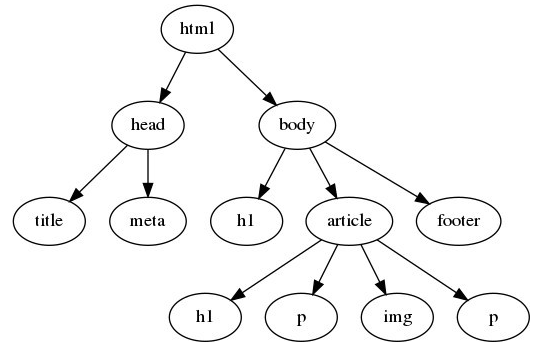


# CHAPITRE 6.1 : EXERCICES SUR LES ARBRES

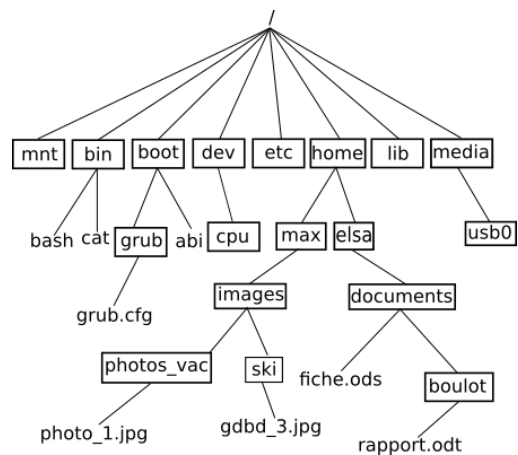
## Exercice 1 : Une page html

1. Combien de feuilles possède cet arbre ?
2. Quelle est l'étiquette de la racine de cet arbre ?
3. Quelle est sa taille ?
4. Quelle est sa hauteur ?
5. Quelle est la profondeur du nœud dont l'étiquette est image ?
6. Combien d'enfants possède cet arbre ?
7. Quel est le nœud qui possède le plus d'enfants ?



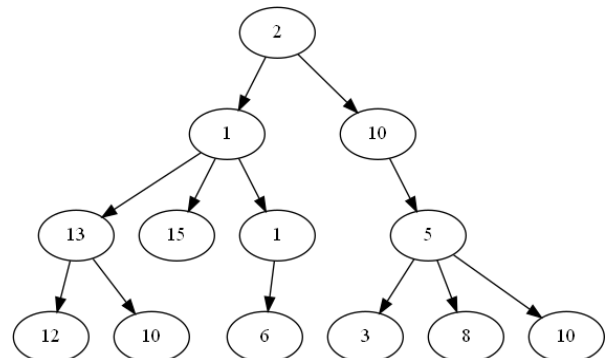
## Exercice 2 : Un système de fichiers

1. Combien de feuilles possède cet arbre ?
2. Quelle est l'étiquette de la racine de cet arbre ?
3. Quelle est sa taille ?
4. Quelle est sa hauteur ?
5. Quelle est la profondeur du nœud dont l'étiquette est image ?
6. Combien d'enfants possède cet arbre ?
7. Quel est le nœud qui possède le plus d'enfants ?



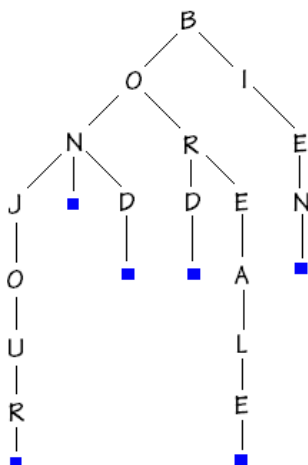
## Exercice 3

Ecrire le parcours en largeur de l'arbre ci-contre :



## Exercice 4

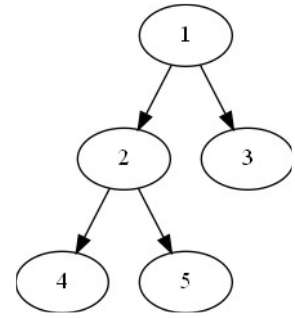
Ecrire le parcours des arbres ci-dessous en profondeur préfixe et postfixe.



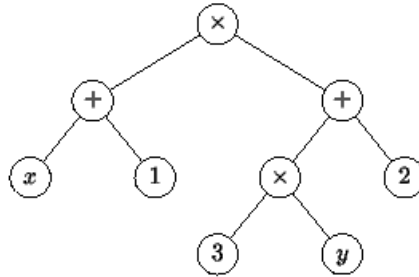
## **Exercice 5 : Construction d'arbres**

1. Dessiner l'arbre a2 construit par le code suivant (a1 est l'arbre vu en cours, que l'on redonne ci-contre) :

```
a2 = Nœud (0)
a2.ajoute(Nœud (5),a1)
```



2. Donner le code permettant de dessiner l'arbre a3 ci-contre



3. Dessiner l'arbre a4 construit par le code suivant, sachant que a4.enfants est la liste des enfants de a4 .

```
a4 = Nœud (1)
a4.ajoute(Nœud ('A'), Nœud ('B'), Nœud ('C'))
for i in range (3):
    a4.enfants[i].ajoute(Nœud (3*i), Nœud (i+5))
```

## **Exercice 6**

1. Représenter les arbres A, B et D construits par le code ci-dessous :

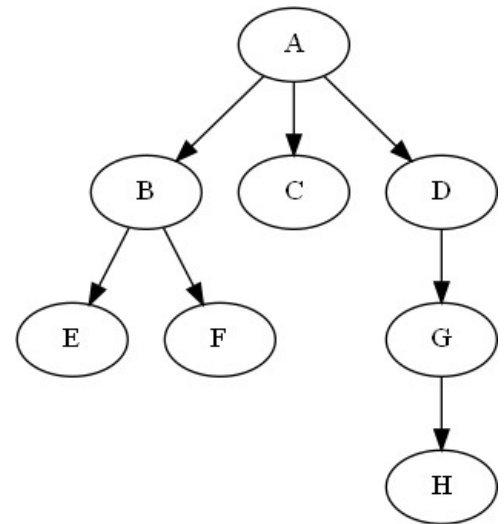
```
A = Nœud ('A')
B = Nœud ('B')
C = Nœud ('C')
D = Nœud ('D')
E = Nœud ('E')
F = Nœud ('F')
G = Nœud ('G')
H = Nœud ('H')
B.ajoute(E, F)
D.ajoute(G)
A.ajoute(B, C, H)
G.ajoute(H)
```

2. Représenter l'arbre A après l'instruction A.enfants[2] = D

## Exercice 7

On admet que si A est un arbre :

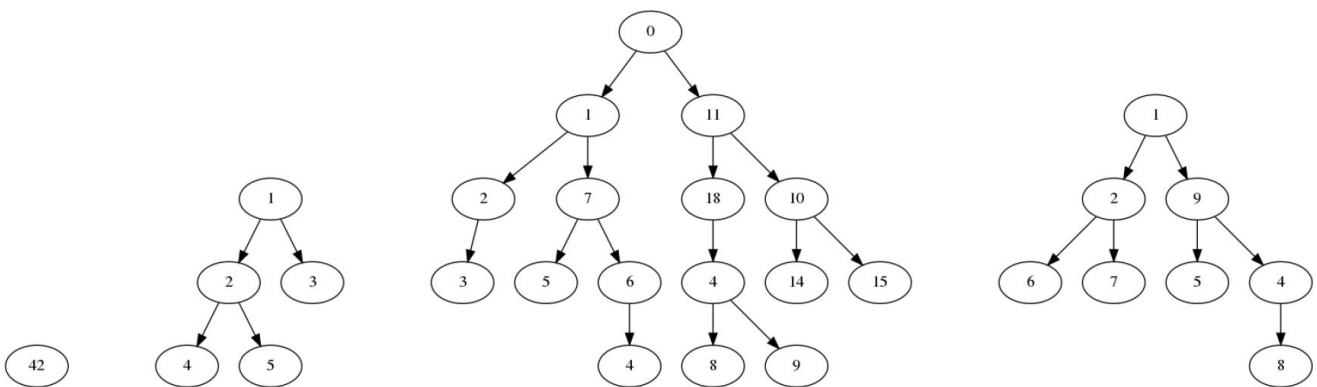
- A.enfants est la liste des enfants de A .
- A.label est l'étiquette de A .



1. Quels sont le type et la valeur de A.label ?
2. Quels sont le type et la valeur de A.enfants ?
3. Quels sont le type et la valeur de A.enfants[0] ?
4. Quels sont le type et la valeur de A.enfants[0].label ?
5. Quels sont le type et la valeur de A.enfants[0].enfants[1].label ?

## Exercice 8

Dans cet exercice, on considère les arbres a0, a1, a5 et a6 respectivement affichés ci-dessous :



1. Compléter le tableau ci-dessous :

	a0	a1	a5	a6
Taille				
Hauteur				
Nombre de feuilles				
Plus grande étiquette				

2. Voici une première fonction qui prend un arbre en paramètre :

```

def est_une_feuille(arbre):
    if arbre.enfants==[]:
        return True
    else:
        return False
  
```

- Est-ce une fonction récursive ?

• Quelles sont les valeurs de f1 , f2 , f3 et f4 suivantes ?

- f1 = est\_une\_feuille(a0)
- f2 = est\_une\_feuille(a1)
- f3 = est\_une\_feuille(a1.enfants[0])
- f4 = est\_une\_feuille(a1.enfants[1])

3. Voici une autre fonction qui prend également un arbre en paramètre :

```
def mystere(arbre):  
    if est_une_feuille(arbre):  
        total = 1  
    else:  
        total = 0  
        for sous_arbre in arbres.enfants:  
            total=total+mystere(sous_arbre)  
    return total
```

- Est-ce une fonction récursive ?
- Donner les valeurs de mystere(a0), mystere(a1), mystere(a5) et mystere(a6).
- Proposer un nom plus adapté à cette fonction.

4. On veut écrire une fonction somme(arbre) qui prend en paramètre un arbre et qui renvoie la somme des étiquettes de tous les nœuds de l'arbre.

- Quelles seront les valeurs de somme(a0) et somme(a1) ?
- Proposer un code pour cette fonction.

5. Ecrire le code d'une fonction e\_max(arbre) qui prend en paramètre un arbre et qui renvoie la valeur de l'étiquette la plus grande.