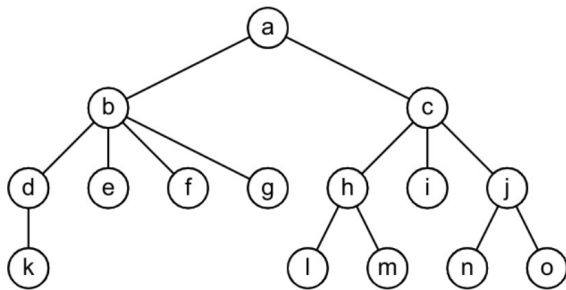


# EXERCICES SUR LES ARBRES

I) Soit l'arbre ci-dessous :



- 1) Quelle est la racine de cet arbre ?
- 2) Quelle est sa hauteur ?
- 3) Quelle est sa taille ?
- 4) Combien possède-t-il de feuilles ?
- 5) Quelle est la profondeur du nœud b ?
- 6) Quel est le nœud qui possède le plus d'enfants ?  
Le plus de descendants ?

II) Voici le code HTML d'une page web :

On appelle DOM (Document Object Model) l'arbre représentant l'organisation des balises dans un document HTML.

```
<html>
  <head>
    <meta charset="utf-8"/>
    <title>Ma super page</title>
  </head>
  <body>
    <header>
      <h1>Plantez des arbres !</h1>
      
    </header>
    <section>
      <h2>C'est bon pour la planète</h2>
      <p>Parce que ça capte le carbone.</p>
      <h2>C'est bon pour la complexité</h2>
      <p>Parce que le logarithme c'est mieux !</p>
    </section>
  </body>
</html>
```

- 1) Dessinez le DOM de cette page HTML. Chaque nœud contiendra une balise et on ne tiendra pas compte des attributs, textes, balises fermantes...
- 2) Quelle est la racine de cet arbre ?
- 3) Quelle est sa hauteur ?
- 4) Quelle est sa taille ?
- 5) Quelle est la profondeur de l'image « arbre.png » ?
- 1) Combien le nœud « section » a-t-il d'enfants ?

III) On considère la fonction « mystère » ci-dessous :

```
def mystère(x):
    if x>0:
        mystère(x-1)
        print("0", end="")
        mystère(x-1)
    else:
        print("1", end="")
```

- 1) On exécute dans la console : mystère(3)  
Dessinez l'arbre des appels récursifs et y inclure les chiffres que le programme va afficher.
- 2) En déduire le nombre affiché par mystère(3).
- 3) Quelle est la racine de l'arbre des appels récursifs ?
- 4) Quelle est sa hauteur ?
- 1) Quelle est sa taille ?

IV) On considère les arbres (non binaires) ci-dessous. Cherchez l'intrus :

