

**Exercice 1 :**

1. — préfixe :  $\times - 12\ 8 + 7\ 9$   
 — infix :  $12 - 8 \times 7 + 9$   
 — postfix :  $12\ 8 - 7\ 9 + \times$
2. 64
3. Parcours infix

**Exercice 2 :**

1. — en largeur : 1 2 3 4 5 6 7 8 9 10 11 12 13  
 — préfixe : 1 2 4 8 5 3 6 9 10 12 13 7 11  
 — infix : 4 8 2 5 1 9 6 12 10 13 3 11 7  
 — postfix : 8 4 5 2 9 12 13 10 6 11 7 3 1
2. La hauteur est 4.
3. Cet arbre est équilibré car la hauteur de chaque sous-arbre gauche diffère au plus de 1 de chaque sous-arbre droit.
4. Cet arbre n'est pas complet car tous les niveaux ne sont pas remplis.

**Exercice 3 :**

1. Le numéro 17 est une femme (indice impair). Son père a pour indice 34 et sa mère 35. Son enfant a pour indice 8.
2. Quatrième génération :  $2^4 = 16$  personnes (la numérotation commence à 1).
3. Chaque niveau  $i$  contient  $2^{i-1}$  ascendants (la numérotation commence à 1). La somme de tous les niveaux correspond à la somme de termes d'une suite géométrique de raison 2 et de premier terme 1.

$$2^0 + 2^1 + 2^2 + 2^3 + 2^4 = \sum_{k=0}^4 2^k = \frac{1 - 2^{4+1}}{1 - 2} = 31$$

4. Représenter l'arbre généalogique sous forme d'un arbre binaire.
5. Ouvrir le fichier *arbre-genealogique.py*.

```

6.
1 # Pour respecter la numérotation de Sosa-Stradonitz la première cellule
   est laissée vide
2 macron = ["", "Emmanuel Macron", "Jean-Michel Macron", "Françoise Noguè
   s",
3         "André-Henri Macron", "Jacqueline Robert", "Jean Noguès",
4         "Germaine Arribet", "Henri Eugène Macron", "Marie Adèle Bosseur
   ",
5         "Georges William Robert", "Suzanne Leblond", "Fabien Noguès",
6         "Esther Mas", "Ernest Arribet", "Marie Madeleine Millet"]

```

Code 1 – Représentation en mémoire d'un arbre généalogique

Écrire la fonction `get_parents(tab : list, id_enfant : int) → tuple` qui renvoie le nom des parents de `id_enfant` sous forme de tuple. La fonction devra gérer les identifiants trop grands à l'aide d'une assertion.

7. Écrire la fonction `ascendant_homme(tab : list, hommes : list) → list` qui renvoie la liste des hommes de la personne dont on a réalisé l'arbre généalogique. La fonction devra parcourir l'arbre en profondeur. De plus une simple liste Python fera office de pile.

8. Écrire une version récursive de la fonction précédente.

nombre de feuilles (récursif)  
parcours profondeur de listes  
insertion, suppression  
binarytree  
tri par tas  
numérotation de Sosa-Stradonitz