

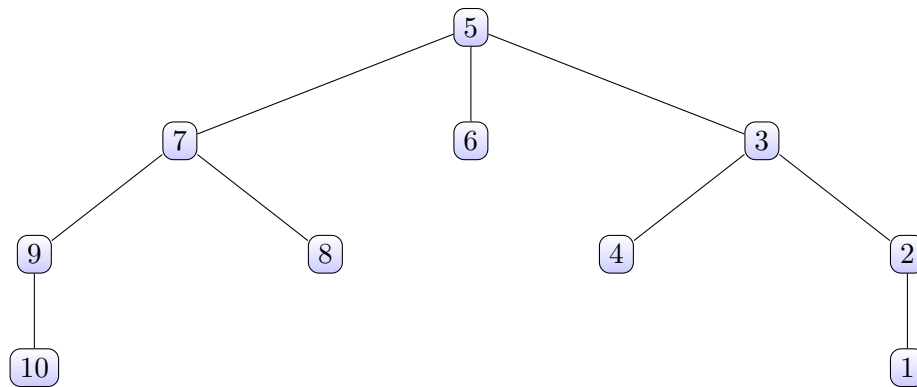
(\*) Exercices à faire à la maison.

**Exercice 1. Parcours.**

Dessiner l'arbre général dont le résultat des parcours préfixe et suffixe sont respectivement  $[1, 2, 4, 6, 3, 5]$  et  $[4, 6, 2, 5, 3, 1]$ .

**Exercice 2. Codage préfixe.**

- Donner le codage préfixe de l'arbre général suivant



- On considère un arbre général dont les sommets sont numérotés de  $\{1, 2, \dots, 14\}$  et dont le code de préfixe est donné par la liste suivante  
 $L = ((5, 4), (4, 3), (6, 0), (7, 2), (10, 0), (8, 0), (3, 1), (9, 0), (2, 0), (1, 2), (13, 1), (11, 1), (12, 0), (14, 0)).$   
 Dessiner l'arbre générale correspondant.

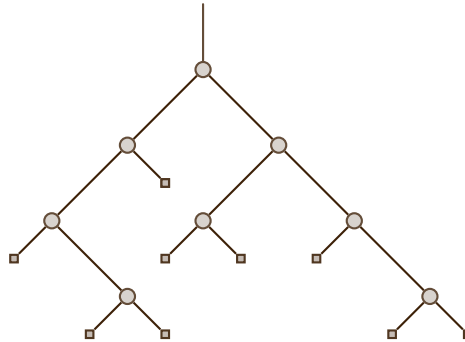
**Exercice 3. Vers les arbres binaires.**

1. Donner l'image des arbre généraux de l'exercice précédent par la fonction  $f$  ("arbres généraux  $\rightarrow$  arbres binaires sans sous-arbre droit") vue en cours.
2. Donner un exemple d'arbre binaire qui n'a pas d'antécédent par cette fonction.
3. Prouver par récurrence sur la hauteur d'un arbre général, que l'ordre préfixe est conservé pour tout arbre général. On peut pour ça étendre la fonction  $f$  aux forêts
4. Prouver de même que l'ordre suffixe (ou postfixe), devient l'ordre infixe.

**Exercice 4. Canopée d'un arbre binaire.**

Soit  $t$  un arbre binaire. La *canopée* de  $t$  est un mot binaire codant l'orientation des feuilles de  $t$  considérées de gauche à droite, tel que si la feuille est orientée vers la gauche (c'est-à-dire est fille gauche d'un nœud) on la code par 0 et si la feuille est orientée vers la droite (c'est-à-dire est fille droite d'un nœud) on la code par 1.

Par exemple, l'arbre binaire



a pour canopée 001101001.

Écrire une fonction *canop* paramétrée par un arbre binaire *t* qui renvoie la liste de 0 et de 1 correspondant à la canopée de *t*.

**Exercice 5.** Arbres binaires équilibrés.

Un arbre binaire *t* est *équilibré* si et seulement si pour tout nœud interne *x* de *t*,

$$|\text{Hauteur}(\text{SAG}(x)) - \text{Hauteur}(\text{SAD}(x))| \leq 1.$$

Écrire une fonction *est\_equilibre* paramétrée par un arbre binaire *t* qui teste si *t* est équilibré.

**Exercice 6.** \*.

Écrire une fonction *trasfBinGen* qui prend en parametre un arbre binaire sans sous-arbre droit et renvoie le code prefixe décrit en cours de l'arbre général correspondant. Testez cette fontion dans un main.