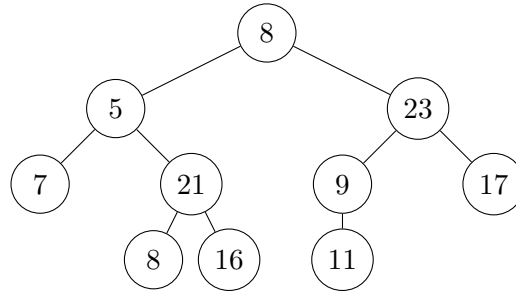


TP 5 – Arbres binaires

Exercice 1.

1. Créer une structure **Arbre** en C++ permettant de représenter un arbre binaire ayant des noeuds étiquetés par des entiers.
2. Créer l'arbre binaire T suivant :



Exercice 2.

1. A la main, faire le parcours préfixé de T . Ecrire une fonction permettant d'afficher les noeuds d'un **Arbre** lors d'un parcours préfixé.
2. A la main, faire le parcours infixé de T . Ecrire une fonction permettant d'afficher les noeuds d'un **Arbre** lors d'un parcours infixé.
3. A la main, faire le parcours postfixé de T . Ecrire une fonction permettant d'afficher les noeuds d'un **Arbre** lors d'un parcours postfixé.

Exercice 3.

1. Ecrire une fonction comptant le nombre de noeuds d'un **Arbre**.
2. Ecrire une fonction comptant le nombre de feuilles d'un **Arbre**.
3. Ecrire une fonction comptant le nombre de noeuds internes d'un **Arbre**.

Exercice 4.

1. Ecrire une fonction qui détermine la hauteur d'un **Arbre**.
2. Ecrire une fonction qui compte le nombre de noeuds à la profondeur p dans un **Arbre**.
3. Ecrire une fonction qui affiche les noeuds à la profondeur p d'un **Arbre**.

Exercice 5.

1. Ecrire une fonction qui teste si un **Arbre** est plein.
2. Ecrire une fonction qui teste si un **Arbre** est complet.

Un arbre est dit équilibré si en tout noeud la différence de hauteur entre ses sous-arbres est d'au plus 1.

3. Ecrire une fonction testant si un **Arbre** est équilibré.