Récupérez les fichiers. Les fichiers `progListeSC.h` et `progListeSC.cpp` contiennent la définition du type ListeSC.

Le fichier `fichierTP5.cpp` contient la définition du type Dico (dictionnaire représenté par un arbre binaire étiqueté par des booléens) avec la fonction creerDico et des fonctions pour l'affichage d'un arbre. Il contient les entêtes d'autres fonctions sur les dictionnaires que vous aurez à compléter. L'affichage des arbres utilisera les langages et outils `dot` du logiciel `Graphviz` pour la visualisation des graphes.

Contrairement au dictionnaire vu en cours et TD, les mots ici sont construits sur l'alphabet $\{0, 1\}$ et non $\{a, b\}$. Un mot est représenté par une liste de 0 et 1. Par exemple le dictionnaire ci-dessous représente l'ensemble de mots : $\{001, 01, 1, 10\}$.

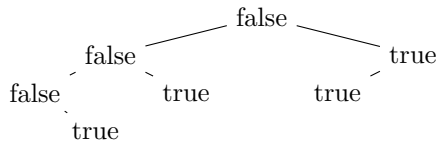
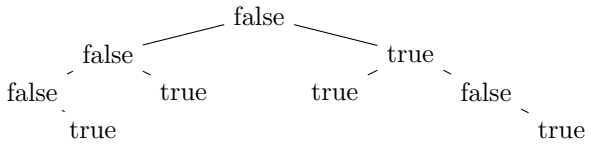
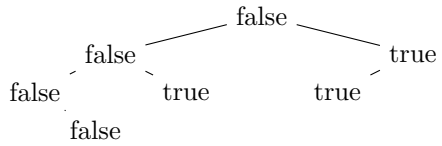


FIGURE 1 – arbre binaire codant le dictionnaire $\{001, 01, 1, 10\}$

1. Complétez la fonction `nbMots` qui calcule le nombre de mots d'un dictionnaire.
   Exemple : pour l'arbre `A` de la figure 1, `nbMots(A)` renvoie 4.
   Compilez et exécutez `tp5` avec l'option 1 pour tester votre fonction.
2. Complétez la fonction `appMot` qui teste si mot appartient à un dictionnaire.
   Exemple : avec l'arbre de la figure 1 et la liste `L=(0,1,1)` `appMot(A,L)` renvoie `false`.
   Compilez et exécutez `tp5` avec l'option 2 pour tester votre fonction.
3. Complétez la fonction `ajouterMot` qui ajoute un mot à un dictionnaire. Celui–ci est donc modifié. Exemple : si L est le mot (1,1,1) après l'exécution de `ajouterMot(A,L)`, l'arbre A de la figure 1 devient :



   Compilez et exécutez `tp5` avec l'option 3 pour tester votre fonction.
4. Complétez la fonction `supprimerMot` qui supprime un mot d'un dictionnaire. Celui–ci est donc modifié.
   Exemple : si L est le mot `(0,0,1)` après l'exécution de `supprimerMot(A,L)` ,l'arbre `A` de la figure 1 devient :



   Compilez et exécutez `tp5` avec l'option 4 pour tester votre fonction.
5. Complétez la fonction `lgMin` qui renvoie la longueur du plus petit mot d'un dictionnaire non vide.
   Exemple : pour l'arbre de la figure 1, `lgMin(A)` renvoie 1.
   Compilez et exécutez `tp5` avec l'option 5 pour tester votre fonction.
6. Complétez la fonction `contientPrefixe` qui teste si un dictionnaire contient un mot et l'un de ses préfixes.
   Exemple : le dictionnaire de la figure 1 contient le mot `(1,0)` et son préfixe `(1)`, `contientPrefixe(A)` renvoie true.
   Compilez et exécutez `tp5` avec l'option 6 pour tester votre fonction.
7. Complétez la fonction `motMin` qui calcule le plus petit mot, dans l'ordre lexicographique (alphabétique), d'un dictionnaire.
   Exemple : avec l'arbre de la figure 1, `motMin(A)` renvoie la liste (0,0,1).
   Compilez et exécutez `tp5` avec l'option 7 pour tester votre fonction.