Question 1 :

La fonction **getVars** prend une formule logique en entrée et renvoie la liste des variables présentes dans la formule. La fonction utilise une fonction auxiliaire récursive **aux** pour parcourir la formule et ajouter les variables à une liste accumulatrice. La fonction **List.sort\_uniq** compare est ensuite utilisée pour trier la liste des variables sans doublons.

La fonction **string\_of\_var** convertit une variable **tformula** en sa représentation sous forme de chaîne de caractères. Elle est utilisée pour convertir les variables **p1, p2, q1, q2** en chaînes de caractères avant de les comparer dans l'assertion.

L'assertion vérifie si la liste des variables retournée par **getVars ex1** est égale à la **liste [string\_of\_var p1; string\_of\_var p2; string\_of\_var q1; string\_of\_var q2]**. Cela garantit que la fonction **getVars** fonctionne correctement en extrayant les variables de la formule.

Question 2 :

La fonction **evalFormula** prend en paramètres un environnement **env** et une formule logique **formula** et retourne la valeur booléenne résultante de l'évaluation de la formule dans cet environnement.

Dans le cas d'une constante **(Value b),** la fonction renvoie simplement la valeur booléenne **b**.

Pour une variable **(Var v),** la fonction recherche la valeur correspondante dans l'environnement en utilisant **List.assoc v env**.

Pour les autres opérations logiques (négation, conjonction, disjonction, implication, équivalence), la fonction évalue récursivement les sous-formules et applique l'opération logique correspondante.

Pour l'exemple donné, l'appel **evalFormula [("P1", false); ("P2", false); ("Q1", false); ("Q2", false)] ex1** renverra **true**.