**Exercices 1 :**

Soit le document XML suivant :

<livre titre="Mon livre">

<auteurs>

<auteur nom="nom1" prenom="prenom1"/>

<auteur nom="nom2" prenom="prenom2"/>

</auteurs>

<sections>

<section titre="Section1">

<chapitre titre="Chapitre1">

<paragraphe>Premier paragraphe</paragraphe>

<paragraphe>Deuxième paragraphe</paragraphe>

</chapitre>

</section>

<section titre="Section2">

<chapitre titre="Chapitre1">

<paragraphe>Premier paragraphe</paragraphe>

<paragraphe>Deuxième paragraphe</paragraphe>

</chapitre>

</section>

</sections>

</livre>

Écrivez les expressions XPath suivantes au format non abrégé :

– trouver la liste des chapitres de la première section ;

//section[position()=1]/child::chapitre

/child :: section/child :: section[1]/child ::chapitre

– trouver la liste des attributs du premier auteur ;

//auteur[position()=1]/@:\*

//auteur[1]/attribute :\*

– trouver la valeur (fonction string) de l’attribut nom du deuxième auteur ;

String(//auteur[position()=2]/@nom)

– trouver la liste des chapitres contenant deux paragraphes ;

/child ::livre/child ::sections/child ::section/child ::chapitre[count(Paragraphe)=2]

//chapitre[count(child ::Paragraphe)=2]

– trouver la liste des chapitres dont un paragraphe possède le mot Premier ;

//chapitre [contains (child ::paragraphe , ‘’Premier’')]

Child ::paragraphe :node --------- Premier parametre

– trouver la liste des sections ayant un chapitre ;

//section[count(child ::chapitre)=1]

– trouver la liste des éléments ayant un seul attribut ;

//\*[Count(attribute ::\*)=1]

/descendant ::\*[count(@\*)=1]

\*-🡪 Element

– trouver la liste des éléments ayant un ancêtre sections, sous deux formes ;

// sections/descendant ::\*

/descendant ::\*[ancestor ::sections]

– trouver la liste des attributs titre ;

//\*/@titre

– trouver la liste des éléments ayant deux fils et pas d’attributs ;

//\*[count(child ::\*)=2 and count (@\*)=0]

– trouver la liste des sections sans paragraphe ;

//section[count(child ::paragraphe)=0]

– trouver la liste des éléments dont le texte contient le mot paragraphe. //\*[contains(self ::text(), ‘’paragraphe’’)