

Nom :

N° d'étudiant :

Examen XML

— Master d'Ingénierie Informatique —

Janvier 2017, durée 2h.

Les documents sont interdits à l'exception d'une feuille de memento (4 pages A4). L'examen est composé de quatre exercices indépendants. Tous les appareils électroniques y compris les montres sont interdits.

Les réponses aux questions doivent directement être écrites dans les cadres du sujet. Tout ce qui est écrit en dehors des cadres n'est pas pris en compte pour l'évaluation.

► **Exercice 1** On considère les deux DTD suivantes.

```
<!-- DTD 1 -->                <!-- DTD 2 -->
<!ELEMENT root ((a,b*)|(b+,a))>  <!ELEMENT root (a,b)+>
<!ELEMENT a (#PCDATA)>          <!ELEMENT a (#PCDATA)>
<!ELEMENT b (#PCDATA)>          <!ELEMENT b (#PCDATA)>
```

a) Donner un document qui soit valide pour la première DTD mais qui ne soit pas valide pour la seconde DTD.

b) Donner un document qui soit valide pour la seconde DTD mais qui ne soit pas valide pour la première DTD.

► **Exercice 2** On considère le schema XML suivant

```
<?xml version="1.0" encoding="utf-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="s">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="i" minOccurs="2" maxOccurs="3"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="i" type="xsd:string" abstract="true"/>
  <xsd:element name="p" type="xsd:string" substitutionGroup="i"/>
  <xsd:element name="n" type="xsd:string" substitutionGroup="i"/>
</xsd:schema>
```

a) Donner un document valide pour ce schéma

b) Donner une DTD équivalente qui valide les mêmes documents que ce schéma.

► **Exercice 3** On considère des documents valides pour la DTD donnée ci-dessous. Écrire une feuille de style XLST qui prend un de ces documents et donne le document obtenu en conservant uniquement les éléments avec un attribut `use` ayant la valeur `keep` ainsi que les ancêtres de ces éléments et en supprimant tous les autres éléments.

```
<!-- DTD -->
<!ELEMENT t (#PCDATA | t)*>
<!ATTLIST t use CDATA #IMPLIED>
```

Si on applique la feuille de style au document source (à gauche ci-dessous), on doit obtenir le document résultat (à droite ci-dessous).

```
<!-- Document source -->                <!-- Document résultat -->
<?xml version="1.0" encoding="utf-8"?>  <?xml version="1.0"?>
<t>                                        <t>
  <t use="keep"><t>a</t><t>a</t></t>        <t/>
  <t><t>c</t><t use="keep">c</t></t>        <t><t>c</t></t>
</t>                                        </t>
```

► **Exercice 4** On considère le document XML suivant.

```
<?xml version="1.0" encoding="utf-8"?>
<ts>
  <t id="t1">T1</t>
  <t id="t2" ref="t1">T2</t>
  <t id="t3" ref="t2">T3</t>
</ts>
```

a) Donner le résultat de l'application de la feuille de style XSLT suivante au document précédent.

```
<?xml version="1.0" encoding="utf-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:key name="key" match="t" use="@id"/>
  <xsl:template match="/*|text()">
    <xsl:copy>
      <xsl:apply-templates select="node()"/>
    </xsl:copy>
  </xsl:template>
</xsl:stylesheet>
```

b) Donner le résultat de l'application de la feuille de style XSLT suivante au document précédent.

```
<?xml version="1.0" encoding="utf-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:key name="key" match="t" use="@id"/>
  <xsl:template match="/*|text()">
    <xsl:copy>
      <xsl:apply-templates select="node()"/>
    </xsl:copy>
  </xsl:template>
  <xsl:template match="t[@ref]">
    <xsl:copy-of select="key('key',@ref)"/>
  </xsl:template>
</xsl:stylesheet>
```

c) Donner le résultat de l'application de la feuille de style XSLT suivante au document précédent.

```
<?xml version="1.0" encoding="utf-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:key name="key" match="t" use="@id"/>
  <xsl:template match="/*|text()">
    <xsl:copy>
      <xsl:apply-templates select="node()"/>
    </xsl:copy>
  </xsl:template>
  <xsl:template match="t[@ref]">
    <xsl:apply-templates select="key('key',@ref)"/>
  </xsl:template>
</xsl:stylesheet>
```

d) On considère le schéma suivant pour le document précédent. Compléter le pour que les deux attributs id et ref soient bien déclarés comme *identifiant* et *référence*.

```
<?xml version="1.0" encoding="utf-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="ts">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="t" minOccurs="0" maxOccurs="unbounded">
          <xsd:complexType>
            <xsd:simpleContent>
              <xsd:extension base="xsd:string">
                <xsd:attribute name="id" type="xsd:string"/>
                <xsd:attribute name="ref" type="xsd:string"/>
              </xsd:extension>
            </xsd:simpleContent>
          </xsd:complexType>
        </xsd:element>
      </xsd:sequence>
    </xsd:complexType>
    <!-- A compléter ICI -->
  </xsd:element>
</xsd:schema>
```