

1. Consider the following sample XML database:

```
<moviedb>
  <movies>
    <movie id="airforceone">
      <title>Air Force One</title>
      <year>1998</year>
      <length units="minutes">110</length>
      <film>Color</film>
      <actors>
        <actor idref="harrisonford"/>
        <actor idref="glennclose"/>
        ...
      </actors>
      <ownedby idref="mgm"/>
    </movie>
    ...
  </movies>

  <stars>
    <star id="harrisonford">
      <name>Harrison Ford</name>
      <sex>Male</sex>
      <address>
        <street>789 Palm St.</street>
        <city>Beverly Hills</city>
      </address>
      <starredIn>
        <movie idref="airforceone">
        <movie idref="starwars1">
        ...
      </starredIn>
    </star>
    ...
  </stars>
```

```

<studios>
  <studio id="mgm">
    <sname>Metro Goldwyn Mayer</sname>
    <address>Hollywood Boulevard, CA</address>
    <owns>
      <movie idref="airforceone"/>
      <movie idref="starwars1"/>
      ...
    </owns>
  </studio>
  ...
</studios>
</moviedb>

```

Write Lorel queries for the following:

- (a) Get names of all male actors who play a role in "Air Force One".
- (b) Get the names and addresses of studios which own a movie in which "Harrison Ford" has acted.
- (c) For all movies whose title has "War" as a substring (e.g. Star Wars I), get the movie name and the names of the actors who act in it.
- (d) Write a valid DTD for this XML data.

2. Consider the following XML data:

```
<state>
  <sname>Alabama</sname>
  <scode>AL</scode>
  <capital>Montgomery</capital>
  <population>4040587</population>
  <nickname>Yellow Hammer State</nickname>
  <majorcities>
    <cname>Montgomery</cname>
    <cname>Birmingham</cname>
    <cname>Mobile</cname>
  </majorcities>
</state>
```

Assume that there is a variable number of <cname> elements within the <majorcities> element. Write an XSL program that will produce the following HTML code when applied to the above XML data.

```
<html>
<head><title>State Details</title></head>
<body>
<center><BR>
<h2>State Details</h2><BR>
<h2>Alabama</h2>
<table border=0>
<tr>
<td><b>State Code:</b></td><td align="left">AL</td>
</tr>
<tr><td><b>State Capital:</b></td><td align="left">Montgomery</td>
</tr>
<tr><td><b>Nickname:</b></td><td align="left">Yellow Hammer State</td></tr>
<tr><td><b>Population:</b></td><td>4040587</td></tr>
<tr><td><b>Major Cities:</b></td><td>Montgomery, Birmingham, Mobile</td></tr>
</table>
</center>
</body>
</html>
```

3. Consider the hierarchical structure of articles published in various computer science journals. At the top level, we have many journals. Each journal has a journal name and a collection of volumes. Each volume has a volume number, a year and a collection of numbers. Each number has a number number, a date and a collection of articles. Each article has a start page, an end page, a title and a collection of authors. Each author has a name. We have decided to record the information about articles in the following XML format:

```

<Journals>
  <Journal>
    <JournalName>ACM Transactions on Database Systems</JournalName>
    <Volumes>
      <VolumeEntry>
        <Volume>1</Volume>
        <Year>1976</Year>
        <Numbers>
          <NumberEntry>
            <Number>1</Number>
            <Date>March 1976</Date>
            <Articles>
              <Article>
                <Authors>
                  <Author>M. Stonebraker</Author>
                  <Author>D. Maier</Author>
                </Authors>
                <Title>Ingres: A Database Management System</Title>
                <StartPage>1</StartPage>
                <EndPage>18</EndPage>
              </Article>
              ...
            </Articles>
          </NumberEntry>
          ...
        </Numbers>
      </VolumeEntry>
      ...
    </Volumes>
  </Journal>
  ...
</Journals>

```

Write a DTD for this XML data (i.e. the XML data should be valid with respect to the DTD you write).