CMPSCI 445 — Homework 3

60 Points

Due October 15th, at the beginning of class.
Typed solutions preferred. If hand-written, solutions must be **legible**.

XML Queries

In this exercise, you will write queries over an XML data set using the XQuery language. You will execute your queries using the open-source Saxon system, which we have installed in the class directory space on the EdLab machines.

You will use a simple XML file called bib.xml, shown in Figure 1 below. The DTD for this file, called bib.dtd, is shown in Figure 2. The DTD can be helpful in understanding the structure of the data and formulating your queries. Please see the *System Support* page for tips on running your first query, and location of the sample data.

Please turn in: (1) text of your queries, and (2) the query output.

1. List books that were published after 1991 and cost less than $100.

   The output should look like:

   ```
   <book ...>...</book>
   <book ...>...</book>
   ...
   ```

2. List books that were published after 1991 and cost less than $100, including their year, title, and price. The output format should be:

   ```
   <cheap-book year="">
   <title></title>
   <price></price>
   </cheap-book>
   <cheap-book year="">
   <title></title>
   <price></price>
   </cheap-book>
   ```

3. For each publisher, list its name and all books that it has published; each book listed should contain the following child elements in the specified order: year, title, and authors. You can use string() or data() to convert an attribute value to the text content of a new element, e.g., `<new-element> {string($b/@attribute)} </new-element>`

   The output should look like:

```
<publisher>
<name></name>
<book>
<year></year>
<title></title>
<author></author>
<author></author>

</book>
<book>

</book>
.
</publisher>
<publisher>

</publisher>
.
```

4. Create a flat list of all the (title, author) pairs; the title and author in each pair should belong to the same book. Enclose each pair in a "title-author" element. Hint: you can define multiple variables in the for clause, e.g., `for \$x in ..., \$y in ..., \$z in ... where ... return ...`. The output should look like:

```
<title-author>
<title></title>
<author><author>
</title-author>
<title-author>
<title></title>
<author><author>
</title-author>
```

Figure 1: Sample data file *bib.xml*. Available electronically in edlab.

```
<bib>
    <book year="1994">
        <title>TCP/IP Illustrated</title>
        <author><last>Stevens</last><first>W.</first></author>
        <publisher>Addison-Wesley</publisher>
        <price>65.95</price>
    </book>
    <book year="1992">
        <title>Advanced Programming in the Unix environment</title>
        <author><last>Stevens</last><first>W.</first></author>
        <publisher>Addison-Wesley</publisher>
        <price>65.95</price>
    </book>
    <book year="2000">
        <title>Data on the Web</title>
        <author><last>Abiteboul</last><first>Serge</first></author>
        <author><last>Buneman</last><first>Peter</first></author>
        <author><last>Suciu</last><first>Dan</first></author>
        <publisher>Morgan Kaufmann Publishers</publisher>
        <price>39.95</price>
    </book>
    <book year="1999">
        <title>The Economics of Technology and Content for Digital TV</title>
        <editor>
               <last>Gerbarg</last><first>Darcy</first>
               <affiliation>CITI</affiliation>
        </editor>
        <publisher>Kluwer Academic Publishers</publisher>
        <price>129.95</price>
    </book>
</bib>
```

Figure 2: DTD for sample data set.

```
<!ELEMENT bib  (book* )>
<!ELEMENT book  (title,  (author+ | editor+ ), publisher, price )>
<!ATTLIST book  year CDATA  #REQUIRED >
<!ELEMENT author  (last, first )>
<!ELEMENT editor  (last, first, affiliation )>
<!ELEMENT title  (#PCDATA )>
<!ELEMENT last  (#PCDATA )>
<!ELEMENT first  (#PCDATA )>
<!ELEMENT affiliation  (#PCDATA )>
<!ELEMENT publisher  (#PCDATA )>
<!ELEMENT price  (#PCDATA )>
```