Assignment #10

Instructor: Richard Frank (rfrank@sfu.ca)
TA: Ankit Gupta (aga53@sfu.ca)

Total Marks: 50 (5% of the Individual Assignments)
Due Date: Nov 25, 14:30 (via the Online Submission Server)

For this assignment, you have no tools to evaluate the XPath/XQuery you write. To get around this, I found a website which evaluates XPaths, and I wrote a little utility to evaluate XQueries. Please use them to ensure your assignment is correct. Include both the input XPath/XQuery and output result.

XPath Questions) Refer to: http://www.w3schools.com/xpath/xpath_examples.asp

For the below questions, use the XML from the above page, which contains an XML structure for a bookstore. Note that there is a'Try it yourself' link on the above page which allows you to modify the code and see the results. Please test your queries to ensure that your XPath works.

Q1) Select all books published after 2003.

```
path="/bookstore/book[year>'2003']/title";
```

Q2) Display the categories of books available.

```
path="/bookstore/book/@category";
```

Q3) Display the 2nd author in the database.

```
path="/bookstore/book[2]/author";
        or depending on browser
path="/bookstore/book[1]/author";
```

Q4) Display the language of the first book.

```
path="/bookstore/book[1]/title/@lang";
        or depending on browser
path="/bookstore/book[0]/title/@lang";
```

Q5) Display only the book titles that are categorized for the web.

```
path="//bookstore/book[@category='WEB']/title";
```

XQuery Questions) [7 marks each] Refer to the books.xml included as part of this assignment. You can also find it here: http://msdn.microsoft.com/en-us/library/ms762271%28VS.85%29.aspx

Q6) Calculate the number of books with price > 30$.

```
for $book in doc("mydoc.xml")/catalog
let $b :=count($book/book[price> 30])
return $b
```

Q7) Find all fantasy books.

```
for $book in doc("mydoc.xml")/catalog/book
where $book/genre = 'Fantasy'
return $book/title
```

Q8) Find all books with price greater than the average of all books.

```
for $book in doc("mydoc.xml")/catalog
let $b :=avg($book/book/price/text())
for $x in $book/book
where $x/price/text() > $b
return $x/title
```

Q9) Display all distinct authors, ordered alphabetically.

```
for $book in doc("mydoc.xml")/catalog,
$x in distinct-values($book/book/author/text())
order by $x
return <author>{$x}</author>
```

Q10) Display all distinct authors, and the books they wrote.

```
for $book in doc("mydoc.xml")/catalog,
$author in distinct-values($book/book/author/text())
order by $author
return<answer>
<author> {$author} </author>{
for $title in $book/book[author/text()=$author]/title
return $title }</answer>
```