TEI XPath Exercises

All the XPath exercises will be based on extracting data from the eXist native XML database. When you have booted your machine off a Knoppix CD (this may have already been done for you), a web browser will load. In the browser window click on "eXist XQuery interface". This should take you to a web-form which allows you to test out XPath and XQuery queries against the eXist database that is running in memory. We shall be using this interface for both our XPath and our XQuery exercises.

As a sample basic XPath query, in the XQuery box type:

```
(: All Person Elements :)
declare namespace tei="http://www.tei-c.org/ns/1.0";
collection('/db/pc')//tei:person
```

Make sure to notice the following things:

- We have started the query with an XQuery comment. This is comprised of any text between (: and :) so `(: This is a comment :)`. The reason we have done this is so your can distinguish between queries in the Query History at the bottom of the form which stores queries you have made.
- This is followed by a namespace declaration since the XML we are querying is in the TEI namespace
- Moreover, we have specified as part of our query the collection that we are looking at. While this isn't strictly necessary it pays to be specific, especially when you have multiple collections in your database.
- Finally, notice that in our search for any `<person>` elements we have had to specify what namespace each element is in. Hence our basic XPath is `//tei:person` rather than `//person`.

Press the 'Submit Query' button and your query should retrieve a list of all the `<person>` elements (in the TEI namespace) in the Protestant Cemetery collection. If your query has retrieved the expected results clicking on the 'New Query' link on the Query Results page will save your query to the 'Query History' drop-down list at the bottom of the form. If your query did not work properly and you want to edit it, simply click on the back button of your browser.

To examine the full entry for some of the gravestones search instead for `//tei:TEI`, perhaps limiting it to the first 3 results:

```
(: First 3 Stones :)
declare namespace tei="http://www.tei-c.org/ns/1.0";
collection('/db/pc')//tei:TEI[1 to 3]
```

Do as many of the exercises below as you can, writing in the answeres to record your progress if you wish.

XPath Exercises

1. List `<person>` elements of all women.

   ANSWER: The number of hits is _____

2. List the `<person>` elements of all British women. (Hint: Used an attribute of `<nationality>`.)

   ANSWER: The number of hits is _____

3. List only the `<persName>` elements of all British women.

   ANSWER: The third woman's foreName is _____

4. List the `<person>` elements of all British women whose final age was less than or equal to 25

   ANSWER: The number of hits is _____

5. List the `<persName>` elements of all British women whose final age was less than or equal to 25

   ANSWER: The first woman's foreName is _____

6. List the birthplace (`<settlement>`) of elements of all British women

   ANSWER: The eighth woman's birthplace is _____

7. Use the XPath function count() to output the number of record of British women have birthplace (`<settlement>`) of elements.

   ANSWER: The count is _____

8. Now count how many of these were born in London?

   ANSWER: The count is _____

9. Find all `<body>` elements which contain the word 'LOVE' using the XPath function contains(., 'LOVE').

   ANSWER: The number of hits is _____

10. Do the same, except search for 'love' (lowercase).

    ANSWER: The number of hits is _____

11. Now do the same but using the eXist extension for case-insensitive searching such as: `[. &='love']` and although this is case-insensitive, why did it not retrieve as many as `contains(., 'LOVE')`?

    ANSWER: The number of hits is _____

12. Similarly, find all `<body>` elements which contain any word which has 'love' in any part. (Hint: use * as a wildcard)

    ANSWER: The number of hits is _____

13. Use the eXist extension to find all the <body> elements which have both the words love and death.

    ANSWER: The number of hits is _____

14. Use a different eXist extension |= to find all the <body> elements which have *either* the word love *or* death.

    ANSWER: The number of hits is _____

15. Similarly, find all <body> elements which contain any of 'love', 'loved', 'loves' without using a wildcard.

    ANSWER: The number of hits is _____

16. Find all <body> elements where the word 'loving' is near the word 'memory'. (Hint: use the near() function.)

    ANSWER: The number of hits is _____

17. Find all the gravestone records (<TEI> elements) which mention 'Oxford'.

    ANSWER: The number of hits is _____

18. Find all the gravestone records which mention 'Oxford' where the person died in the 1960s.

    ANSWER: The number of hits is _____