**Information Retrieval (Lab Test)**

**10 January 2017**

Given the compressed XML file at

http://tinyurl.com/ir-lab-articles

containing a list of 40,000 scientific articles delimited by the tag <article>…</article>. The goal is to:

1. Create a Lucene index with one document per article, each including three fields:
	* The author name (XML element <author>)
	* The article title (XML element <title>)
	* The journal name (XML element <journal>)

 **Notes:**

* + You can assume that each <article> element has exactly one title, one journal and at least one author. In case of multiple authors, their names must be concatenated into a single field.
	+ Author names must be tokenized with a WhiteSpaceAnalyzer, all other fields must be tokenized with the ClassicAnalyzer.
1. Build a search engine that let a user search articles via a command line interface in which she can specify some keywords (possibly none) for the **author name** and the **title**. In order for a document to appear as a result, **both** the author names and the title must be matched by the correspondent keywords, if any.
The list of resulting documents is ranked by the standard Lucene ranker (as seen in class) and the top-10 results are printed, along with all fields together with the ranking score computed by Lucene for each of them.

Tips:

* We suggest the following procedure:
	1. Build a function that processes the XML file and gets the field values
	2. Test this function to make sure the fields are correctly extracted
	3. Create a script to build the index
	4. Create a script to search the index and run a few test queries
* Relative Xpath queries can be run on single nodes, e.g.
article\_node.xpath(“./author/text()”)
will get the list of text content of all direct children of article\_node having tag “author”