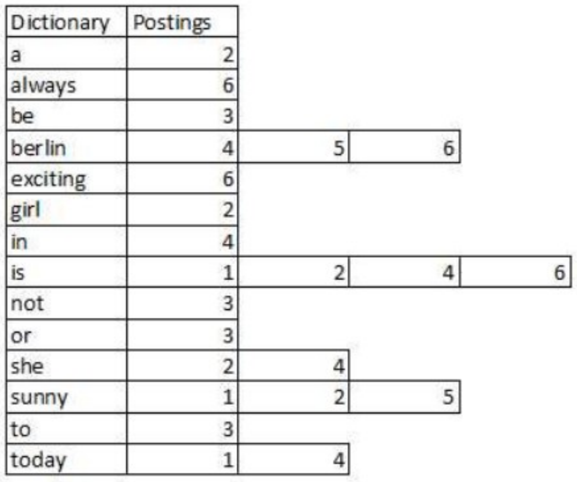
Programming Assignment 2

a. The inverted index looks like:



Dictionary	Postings			
a	2			
always	6			
be	3			
berlin	4	5	6	
exciting	6			
girl	2			
in	4			
is	1	2	4	6
not	3			
or	3			
she	2	4		
sunny	1	2	5	
to	3			
today	1	4		

b. To print a posting list of all the terms indexed above, we:

i. Created two directories – for index and data. Index directory will have indexes and the data directory has 6 documents containing the given sentences.

ii. customAnalyzer with StandardTokenizerFactory and LowerCaseFilterFactory to analyse the documents.

iii. An instance of indexWriter for indexing and added each document in the index with "contents" field.

iv. After indexing, search for each token using IndexSearcher and queryParser.

v. After finding the documents containing given term:

   a. Total term frequency and document frequency is taken from in-built functions ireader.totalTermFreq(term) and ireader.docFreq(term).

   b. To get frequency in the document and position, fetch the TermVector of the given term.

   c. Using postings() function and pointers, collect all the positions and their respective frequency and printed them in desired format.

Output:

```
Documents that contain 'sunny' and 'exciting':
0

Postings List:
[always:1:1]-->[5:1:[10]]
[a:1:1]-->[1:1:[7]]
[be:2:1]-->[2:2:[3,16]]
[or:1:1]-->[2:1:[6]]
[in:1:1]-->[3:1:[7]]
[is:4:4]-->[0:1:[6]]->[5:1:[7]]->[1:1:[4]]->[3:1:[4]]
[girl:1:1]-->[1:1:[15]]
[she:2:2]-->[1:1:[0]]->[3:1:[0]]
[not:1:1]-->[2:1:[9]]
[today:2:2]-->[0:1:[0]]->[3:1:[17]]
[exciting:1:1]-->[5:1:[17]]
[to:2:1]-->[2:2:[0,13]]
[sunny:3:3]-->[4:1:[0]]->[0:1:[9]]->[1:1:[9]]
[berlin:3:3]-->[4:1:[6]]->[5:1:[0]]->[3:1:[10]]
```