**All Queries can be done in 2 ways.**

* Using the Query API
* Using the Lucene Query Parser & Syntax.

# **M2-Lab5 – Term Querying**

**Using API’s Approach**

**Test-1**

$> java -cp "Apache-Lucene-1.0.jar;lib/\*" m2\_lab5.LuceneTermQueryDemo "apples"



Check the file content & validate the output.

**Test-2**

$> java -cp "Apache-Lucene-1.0.jar;lib/\*" m2\_lab5.LuceneTermQueryDemo "apple"



Check the file content & validate the output.

**Test-3**

$> java -cp "Apache-Lucene-1.0.jar;lib/\*" m2\_lab5.LuceneTermQueryDemo "2012"



Check the file content & validate the output.

**Test-4**

$> java -cp "Apache-Lucene-1.0.jar;lib/\*" m2\_lab5.LuceneTermQueryDemo "french fries"



Check the file content & validate the output.

**Test-5**

$> java -cp "Apache-Lucene-1.0.jar;lib/\*" m2\_lab5.LuceneTermQueryDemo "french"



Check the file content & validate the output.

**Using Query Parser Approach**

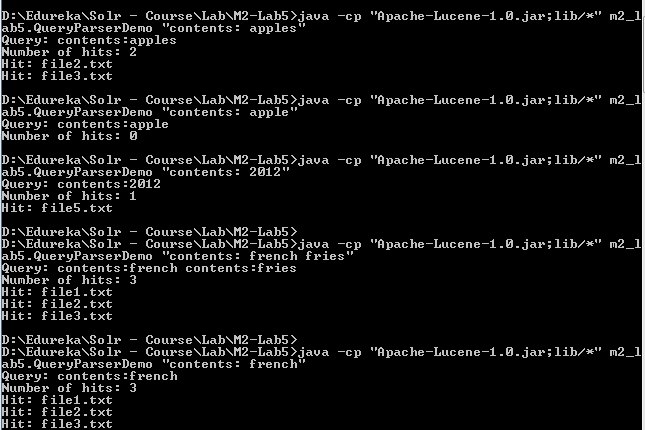
$> java -cp "Apache-Lucene-1.0.jar;lib/\*" m2\_lab5.QueryParserDemo "contents: apples"

$> java -cp "Apache-Lucene-1.0.jar;lib/\*" m2\_lab5.QueryParserDemo "contents: apple"

$> java -cp "Apache-Lucene-1.0.jar;lib/\*" m2\_lab5.QueryParserDemo "contents: 2012"

$> java -cp "Apache-Lucene-1.0.jar;lib/\*" m2\_lab5.QueryParserDemo "contents: french fries"

$> java -cp "Apache-Lucene-1.0.jar;lib/\*" m2\_lab5.QueryParserDemo "contents: french"



Check the file content & validate the output.

# **M2-Lab5 – Boolean Querying**

**Creating & Searching the Index using Boolean Queries**

**Using API’s Approach**

Occur.MUST -> Signifies the term should occur

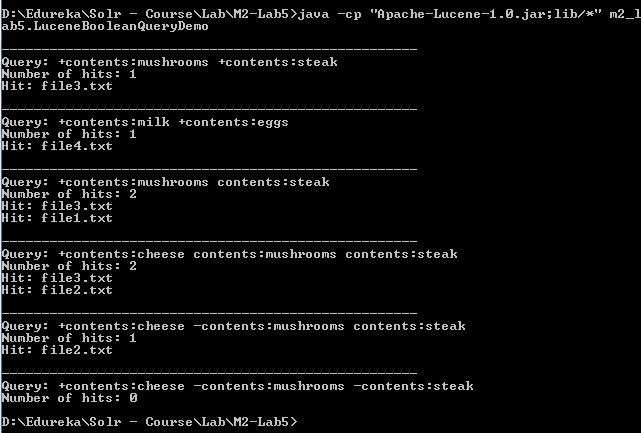
Occur.SHOULD -> Signifies the term is optional.

Occur.MUST\_NOT -> Signifies the term should not occur.

Execute the following command to create new index from the lucene source files for searching & should see something similar output after executing the command.

**Step 1** cd M2-Lab5

**Step 2** java -cp "Apache-Lucene-1.0.jar;lib/\*" m2\_lab5.LuceneBooleanQueryDemo

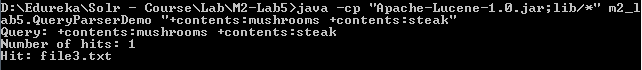


**Using Query Parser Approach**

|  |  |  |
| --- | --- | --- |
| **API** | **Query Parser Syntax** | **Description** |
| Occur.MUST | **+** | Signifies the term should occur. |
| Occur.MUST\_NOT | **-** | Signifies the term should not occur. |
| Occur.SHOULD |  | Signifies the term is optional. |

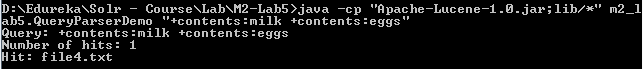
**Test-1**

$> java -cp "Apache-Lucene-1.0.jar;lib/\*" m2\_lab5.QueryParserDemo "+contents:mushrooms +contents:steak"



**Test-2**

$> java -cp "Apache-Lucene-1.0.jar;lib/\*" m2\_lab5.QueryParserDemo "+contents:milk +contents:eggs"



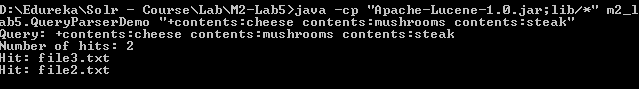
**Test-3**

$> java -cp "Apache-Lucene-1.0.jar;lib/\*" m2\_lab5.QueryParserDemo "+contents:mushrooms contents:steak"



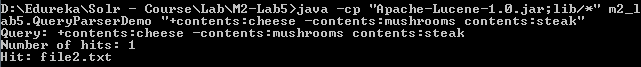
**Test-4**

$> java -cp "Apache-Lucene-1.0.jar;lib/\*" m2\_lab5.QueryParserDemo "+contents:cheese contents:mushrooms contents:steak"



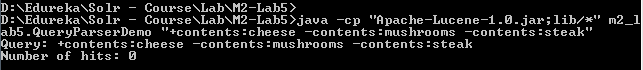
**Test-5**

$> java -cp "Apache-Lucene-1.0.jar;lib/\*" m2\_lab5.QueryParserDemo "+contents:cheese -contents:mushrooms contents:steak"



**Test-6**

$> java -cp "Apache-Lucene-1.0.jar;lib/\*" m2\_lab5.QueryParserDemo "+contents:cheese -contents:mushrooms -contents:steak"



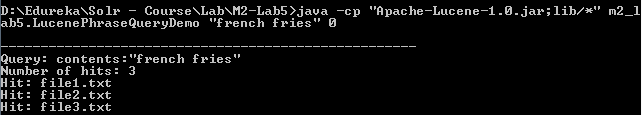
# **M2-Lab5 – Phrase Querying**

**Using API’s Approach**

**$>** cd M2-Lab5

**Test-1**

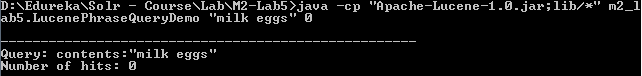
$> java -cp "Apache-Lucene-1.0.jar;lib/\*" m2\_lab5.LucenePhraseQueryDemo "french fries" 0



Check the file content & validate the output.

**Test-2**

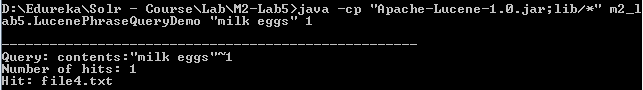
$> java -cp "Apache-Lucene-1.0.jar;lib/\*" m2\_lab5.LucenePhraseQueryDemo "milk eggs" 0



Check the file content & validate the output.

**Test-3**

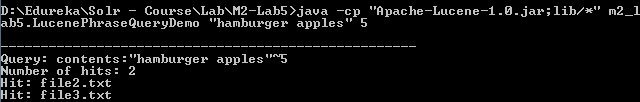
$> java -cp "Apache-Lucene-1.0.jar;lib/\*" m2\_lab5.LucenePhraseQueryDemo "milk eggs" 1



Check the file content & validate the output.

**Test-4**

$> java -cp "Apache-Lucene-1.0.jar;lib/\*" m2\_lab5.LucenePhraseQueryDemo "hamburger apples" 5



Check the file content & validate the output.

**Using Query Parser Approach**

**Test-1**

$> java -cp "Apache-Lucene-1.0.jar;lib/\*" m2\_lab5.QueryParserDemo "\"french fries\"~0"



Check the file content & validate the output.

**Test-2**

$> java -cp "Apache-Lucene-1.0.jar;lib/\*" m2\_lab5.QueryParserDemo "\"milk eggs\"~0"



Check the file content & validate the output.

**Test-3**

$> java -cp "Apache-Lucene-1.0.jar;lib/\*" m2\_lab5.QueryParserDemo "\"milk eggs\"~1"



Check the file content & validate the output.

**Test-4**

$> java -cp "Apache-Lucene-1.0.jar;lib/\*" m2\_lab5.QueryParserDemo "\"hamburger apples\"~5"



Check the file content & validate the output.

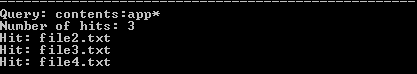
# **M2-Lab5 – Prefix Querying**

**Using API’s Approach**

**$>** cd M2-Lab5

**Test-1**

$> java -cp "Apache-Lucene-1.0.jar;lib/\*" m2\_lab5.LucenePrefixQueryDemo "app"



Check the file content & validate the output.

**Test-2**

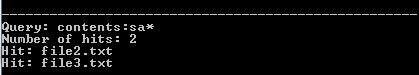
$> java -cp "Apache-Lucene-1.0.jar;lib/\*" m2\_lab5.LucenePrefixQueryDemo "a"



Check the file content & validate the output.

**Test-3**

$> java -cp "Apache-Lucene-1.0.jar;lib/\*" m2\_lab5.LucenePrefixQueryDemo "sa"



Check the file content & validate the output.

**Test-4**

$> java -cp "Apache-Lucene-1.0.jar;lib/\*" m2\_lab5.LucenePrefixQueryDemo "st"

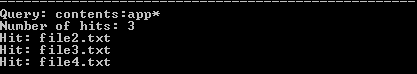


Check the file content & validate the output.

**Using Query Parser Approach**

**Test-1**

$> java -cp "Apache-Lucene-1.0.jar;lib/\*" m2\_lab5.QueryParserDemo "app\*"



Check the file content & validate the output.

**Test-2**

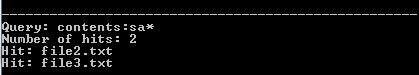
$> java -cp "Apache-Lucene-1.0.jar;lib/\*" m2\_lab5.QueryParserDemo "a\*"



Check the file content & validate the output.

**Test-3**

$> java -cp "Apache-Lucene-1.0.jar;lib/\*" m2\_lab5.QueryParserDemo "sa\*"



Check the file content & validate the output.

**Test-4**

$> java -cp "Apache-Lucene-1.0.jar;lib/\*" m2\_lab5.QueryParserDemo "st\*"



Check the file content & validate the output.

# **M2-Lab5 – Range Querying**

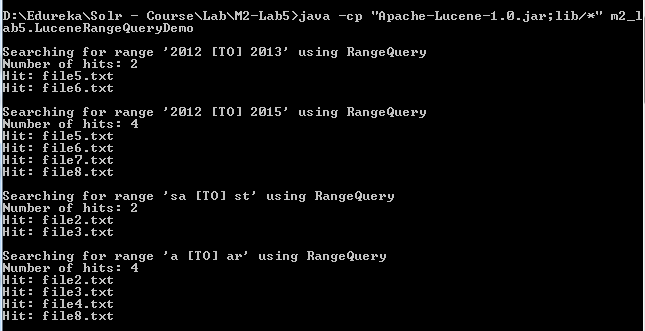
**Using API’s Approach**

* Range queries can be done on both numeric and character based terms.
* Range queries are searched based on the byte stream lexical order.

**$>** cd M2-Lab5

**Test-1**

$> java -cp "Apache-Lucene-1.0.jar;lib/\*" m2\_lab5.LuceneRangeQueryDemo

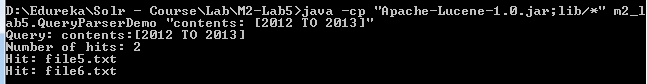


Check the file content & validate the output.

**Using Query Parser Approach**

**Test-1**

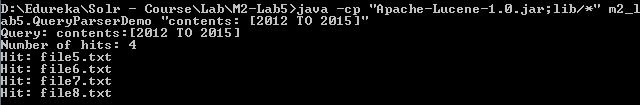
$> java -cp "Apache-Lucene-1.0.jar;lib/\*" m2\_lab5.QueryParserDemo "contents: [2012 TO 2013]"

****

Check the file content & validate the output.

**Test-2**

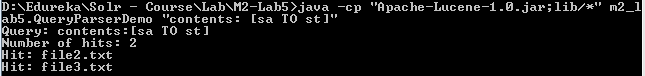
$> java -cp "Apache-Lucene-1.0.jar;lib/\*" m2\_lab5.QueryParserDemo "contents: [2012 TO 2015]"

****

Check the file content & validate the output.

**Test-3**

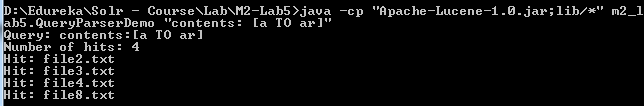
$> java -cp "Apache-Lucene-1.0.jar;lib/\*" m2\_lab5.QueryParserDemo "contents: [sa TO st]"

****

Check the file content & validate the output.

**Test-4**

$> java -cp "Apache-Lucene-1.0.jar;lib/\*" m2\_lab5.QueryParserDemo "contents: [a TO ar]"

****

Check the file content & validate the output.

# **M2-Lab5 – Wild Card Querying**

**Using API’s Approach**

Execute the following commands.

$> cd m2-lab5

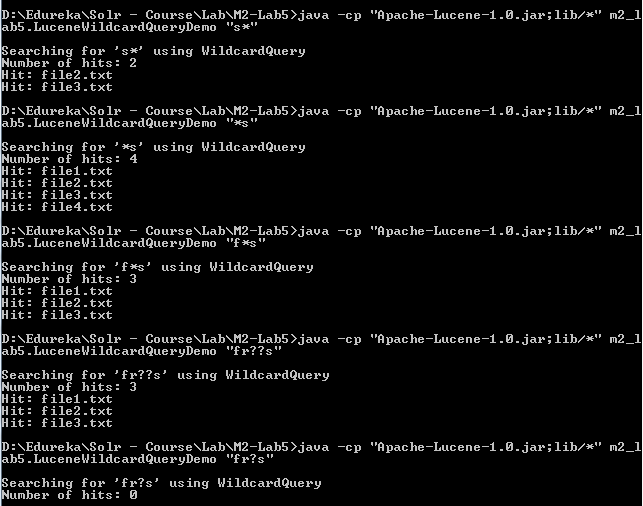
$> java -cp "Apache-Lucene-1.0.jar;lib/\*" m2\_lab5.LuceneWildcardQueryDemo "s\*"

$> java -cp "Apache-Lucene-1.0.jar;lib/\*" m2\_lab5.LuceneWildcardQueryDemo "\*s"

$> java -cp "Apache-Lucene-1.0.jar;lib/\*" m2\_lab5.LuceneWildcardQueryDemo "f\*s"

$> java -cp "Apache-Lucene-1.0.jar;lib/\*" m2\_lab5.LuceneWildcardQueryDemo "fr??s"

$> java -cp "Apache-Lucene-1.0.jar;lib/\*" m2\_lab5.LuceneWildcardQueryDemo "fr?s"

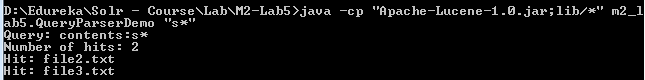


Check the file content & validate the output.

**Using Query Parser Approach**

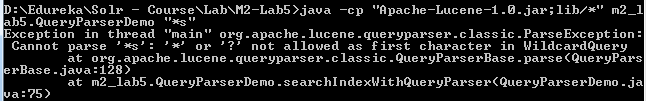
Execute the following commands & validate the file and output.

$> java -cp "Apache-Lucene-1.0.jar;lib/\*" m2\_lab5.QueryParserDemo "s\*"

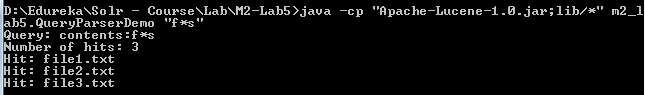


Lead wild cards are not allowed & would through exception.

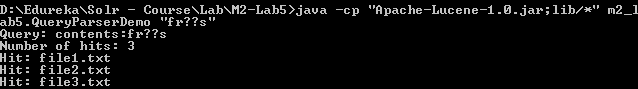
$> java -cp "Apache-Lucene-1.0.jar;lib/\*" m2\_lab5.QueryParserDemo "\*s"



$> java -cp "Apache-Lucene-1.0.jar;lib/\*" m2\_lab5.QueryParserDemo "f\*s"



$> java -cp "Apache-Lucene-1.0.jar;lib/\*" m2\_lab5.QueryParserDemo "fr??s"



$> java -cp "Apache-Lucene-1.0.jar;lib/\*" m2\_lab5.QueryParserDemo "fr?s"

