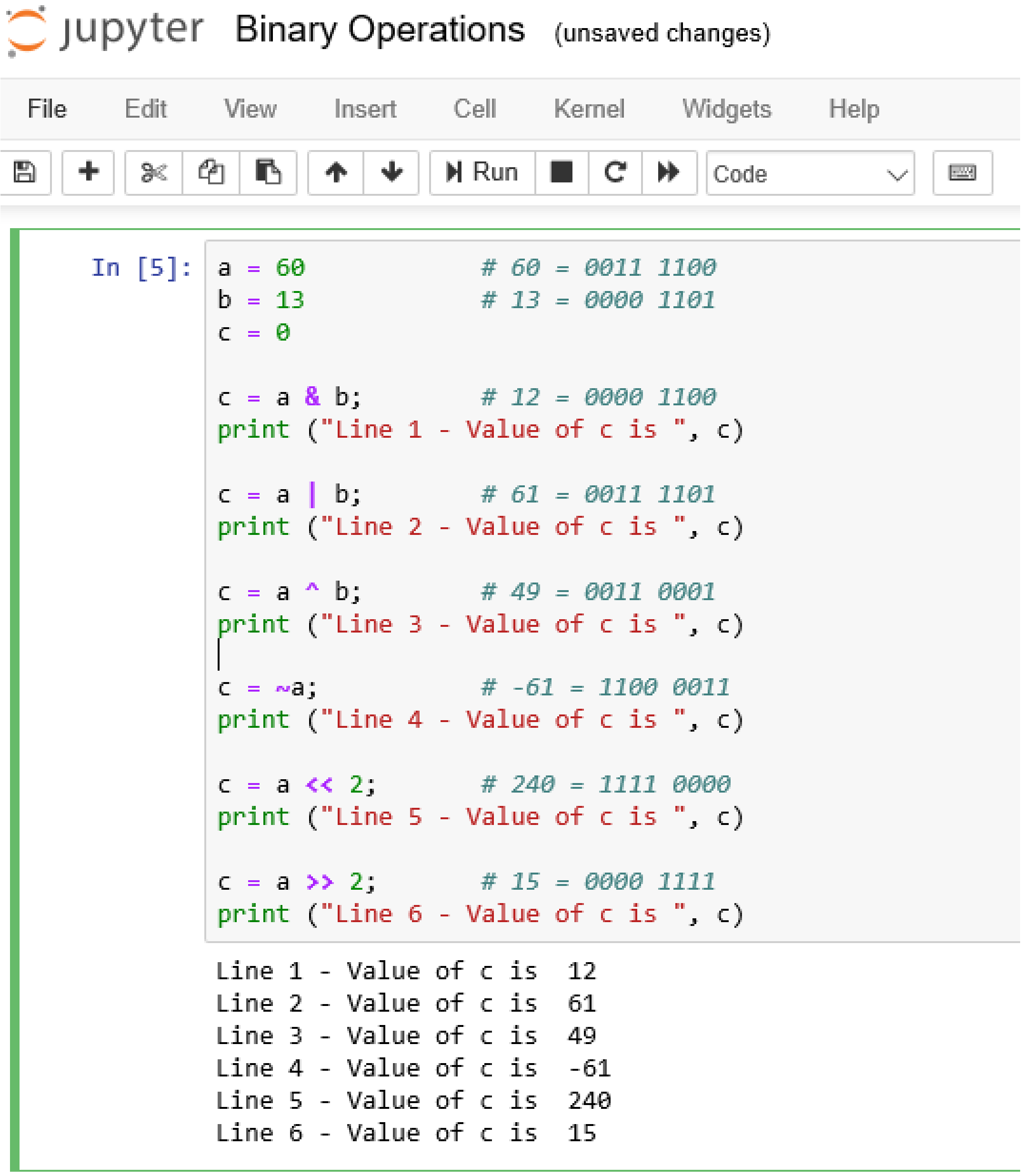
Sr.No	Practical Name	Date	Sign
1	Write a program to demonstrate bitwise operations.		
2	Implement Page Rank Algorithm.		
3	Implement Dynamic programming algorithm for computing the edit distance between string s1 and s2 (Levenshtein Distance).		
4	Write a program to Compute Similarity between two text documents.		
5	Implement a map reduce program to count words and ignore case in Hadoop environment		
6	Implement IR system using lucene		
7	Write a program for Pre-processing of a Text Document: stop word removal.		
8	Write a program to implement simple web crawler.		

Practical 1



jupyter Binary Operations (unsaved changes)
File Edit View Insert Cell Kernel Widgets Help
Run
Code

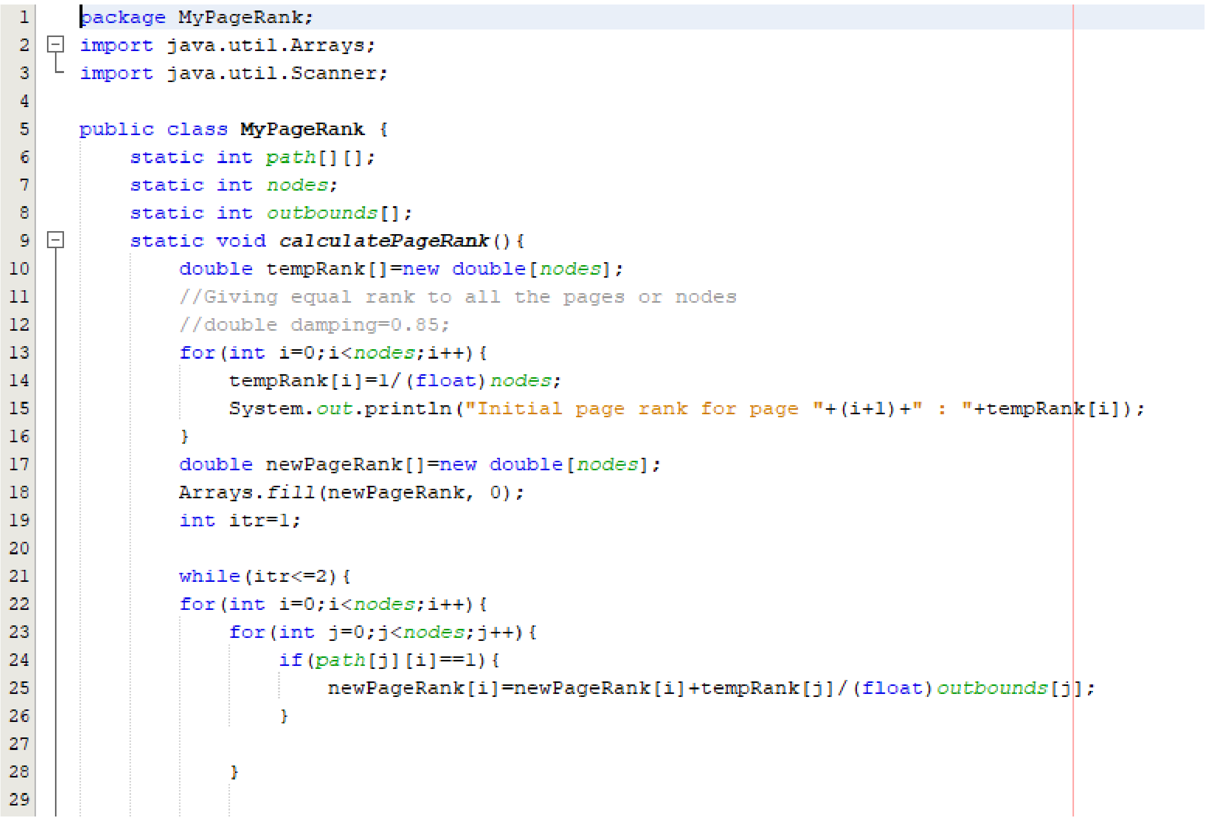
In [5]:

```
a = 60            # 60 = 0011 1100
b = 13            # 13 = 0000 1101
c = 0

c = a & b;        # 12 = 0000 1100
print ("Line 1 - Value of c is ", c)

c = a | b;        # 61 = 0011 1101
print ("Line 2 - Value of c is ", c)

c = a ^ b;        # 49 = 0011 0001
print ("Line 3 - Value of c is ", c)

c = ~a;           # -61 = 1100 0011
print ("Line 4 - Value of c is ", c)

c = a << 2;       # 240 = 1111 0000
print ("Line 5 - Value of c is ", c)

c = a >> 2;       # 15 = 0000 1111
print ("Line 6 - Value of c is ", c)
```

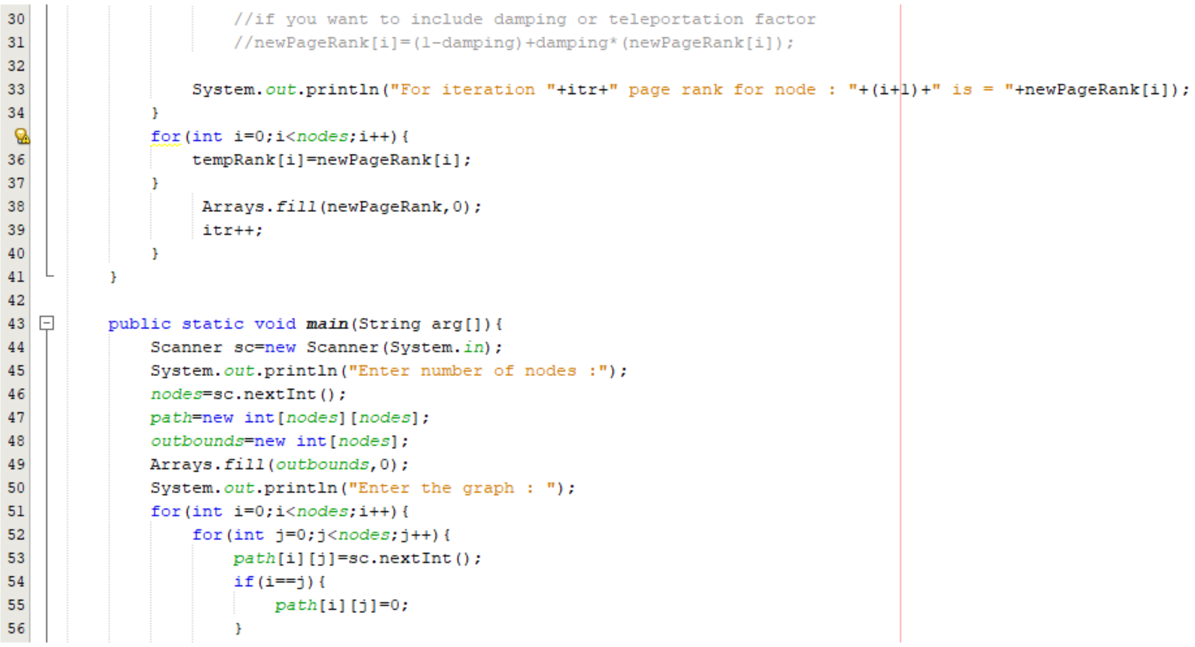
```
Line 1 - Value of c is  12
Line 2 - Value of c is  61
Line 3 - Value of c is  49
Line 4 - Value of c is  -61
Line 5 - Value of c is  240
Line 6 - Value of c is  15
```

Practical 2

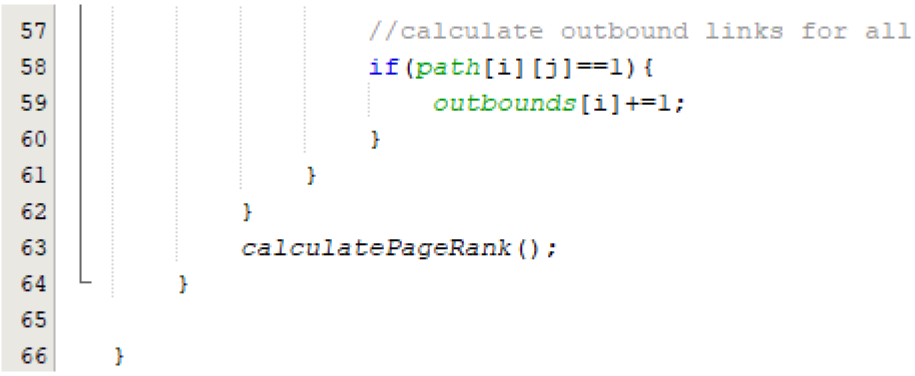
Simple Page Rank



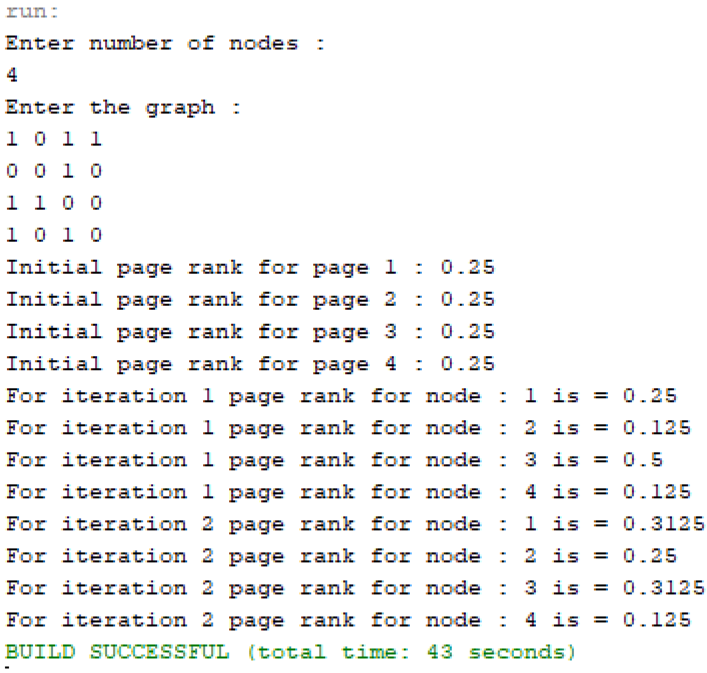
```java
package MyPageRank;
import java.util.Arrays;
import java.util.Scanner;

public class MyPageRank {
    static int path[][];
    static int nodes;
    static int outbounds[];
    static void calculatePageRank(){
        double tempRank[]=new double[nodes];
        //Giving equal rank to all the pages or nodes
        //double damping=0.85;
        for(int i=0;i<nodes;i++){
            tempRank[i]=1/(float)nodes;
            System.out.println("Initial page rank for page "+(i+1)+" : "+tempRank[i]);
        }
        double newPageRank[]=new double[nodes];
        Arrays.fill(newPageRank, 0);
        int itr=1;

        while(itr<=2){
        for(int i=0;i<nodes;i++){
            for(int j=0;j<nodes;j++){
                if(path[j][i]==1){
                    newPageRank[i]=newPageRank[i]+tempRank[j]/(float)outbounds[j];
                }

            }

```



```java
                //if you want to include damping or teleportation factor
                //newPageRank[i]=(1-damping)+damping*(newPageRank[i]);

            System.out.println("For iteration "+itr+" page rank for node : "+(i+1)+" is = "+newPageRank[i]);
        }
        for(int i=0;i<nodes;i++){
            tempRank[i]=newPageRank[i];
        }
            Arrays.fill(newPageRank,0);
            itr++;
        }
    }

    public static void main(String arg[]){
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter number of nodes :");
        nodes=sc.nextInt();
        path=new int[nodes][nodes];
        outbounds=new int[nodes];
        Arrays.fill(outbounds,0);
        System.out.println("Enter the graph : ");
        for(int i=0;i<nodes;i++){
            for(int j=0;j<nodes;j++){
                path[i][j]=sc.nextInt();
                if(i==j){
                    path[i][j]=0;
                }
```



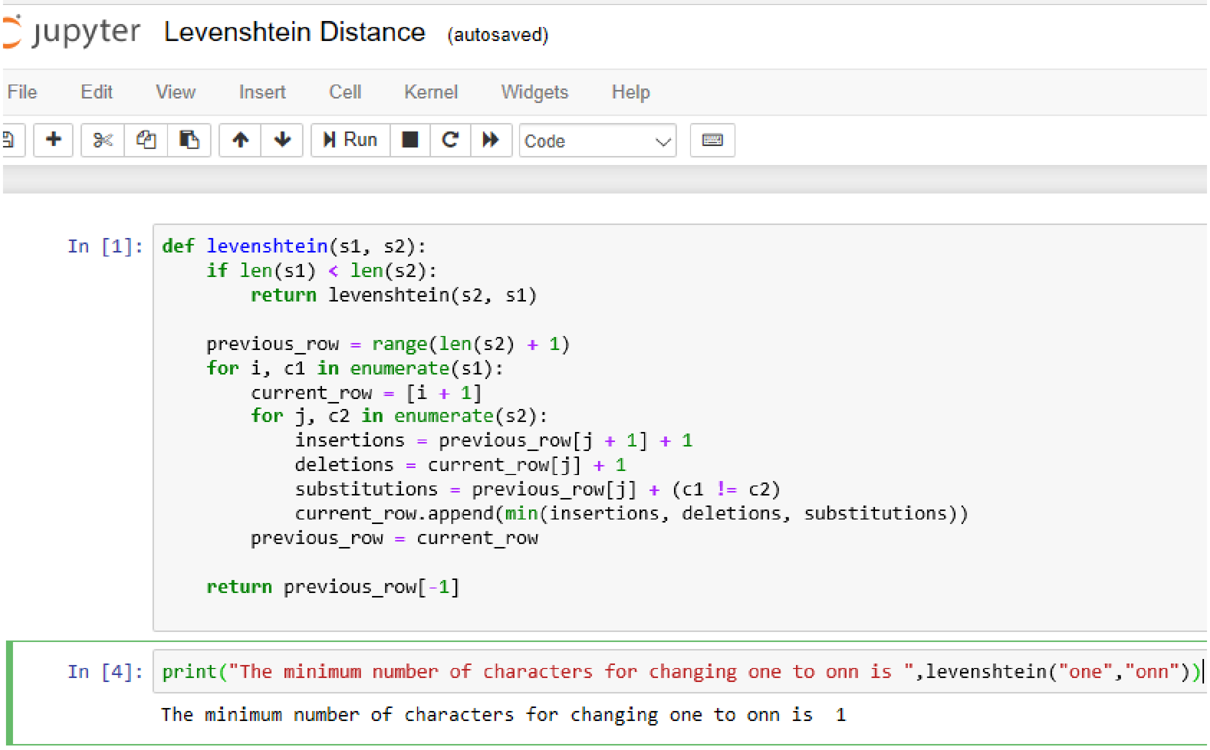
```
                //calculate outbound links for all
                if(path[i][j]==1){
                    outbounds[i]+=1;
                }
            }
        }
        calculatePageRank();
    }

}
```



```
run:
Enter number of nodes :
4
Enter the graph :
1 0 1 1
0 0 1 0
1 1 0 0
1 0 1 0
Initial page rank for page 1 : 0.25
Initial page rank for page 2 : 0.25
Initial page rank for page 3 : 0.25
Initial page rank for page 4 : 0.25
For iteration 1 page rank for node : 1 is = 0.25
For iteration 1 page rank for node : 2 is = 0.125
For iteration 1 page rank for node : 3 is = 0.5
For iteration 1 page rank for node : 4 is = 0.125
For iteration 2 page rank for node : 1 is = 0.3125
For iteration 2 page rank for node : 2 is = 0.25
For iteration 2 page rank for node : 3 is = 0.3125
For iteration 2 page rank for node : 4 is = 0.125
BUILD SUCCESSFUL (total time: 43 seconds)
```

Practical 3

Comupute levenshtein distance



jupyter Levenshtein Distance (autosaved)
File Edit View Insert Cell Kernel Widgets Help
Run Code

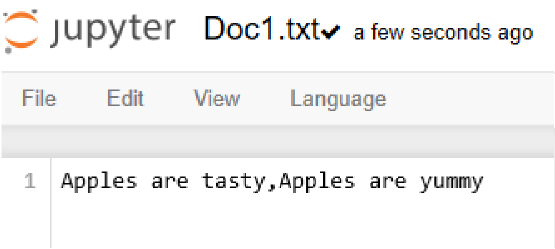
In [1]:
```
def levenshtein(s1, s2):
    if len(s1) < len(s2):
        return levenshtein(s2, s1)

    previous_row = range(len(s2) + 1)
    for i, c1 in enumerate(s1):
        current_row = [i + 1]
        for j, c2 in enumerate(s2):
            insertions = previous_row[j + 1] + 1
            deletions = current_row[j] + 1
            substitutions = previous_row[j] + (c1 != c2)
            current_row.append(min(insertions, deletions, substitutions))
        previous_row = current_row

    return previous_row[-1]
```

In [4]:
```
print("The minimum number of characters for changing one to onn is ",levenshtein("one","onn"))
```

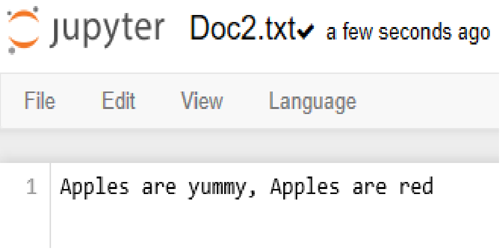
```
The minimum number of characters for changing one to onn is  1
```

Practical 4

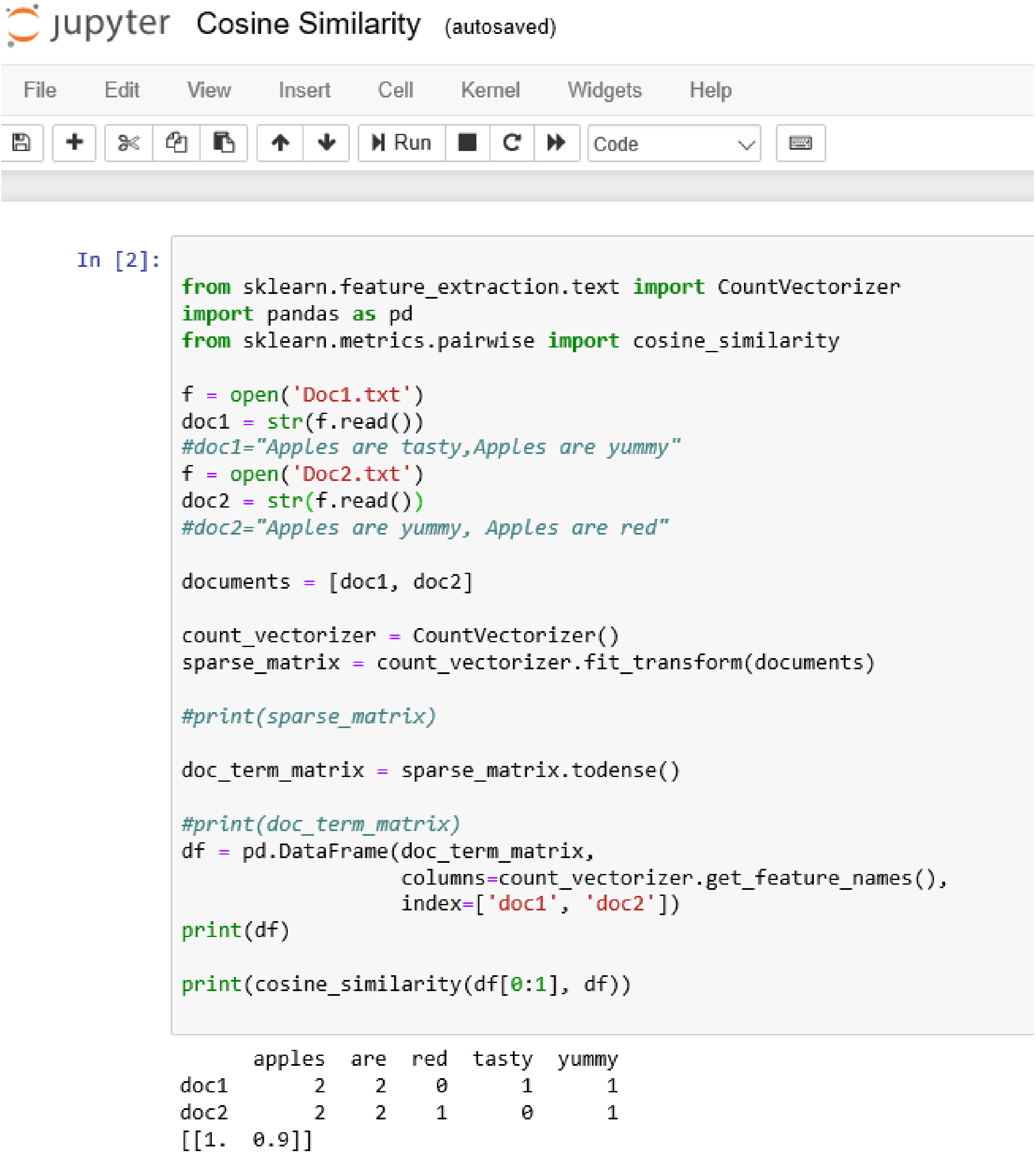
Coumpute Cosine Similarity



```
Apples are tasty,Apples are yummy
```



```
Apples are yummy, Apples are red
```

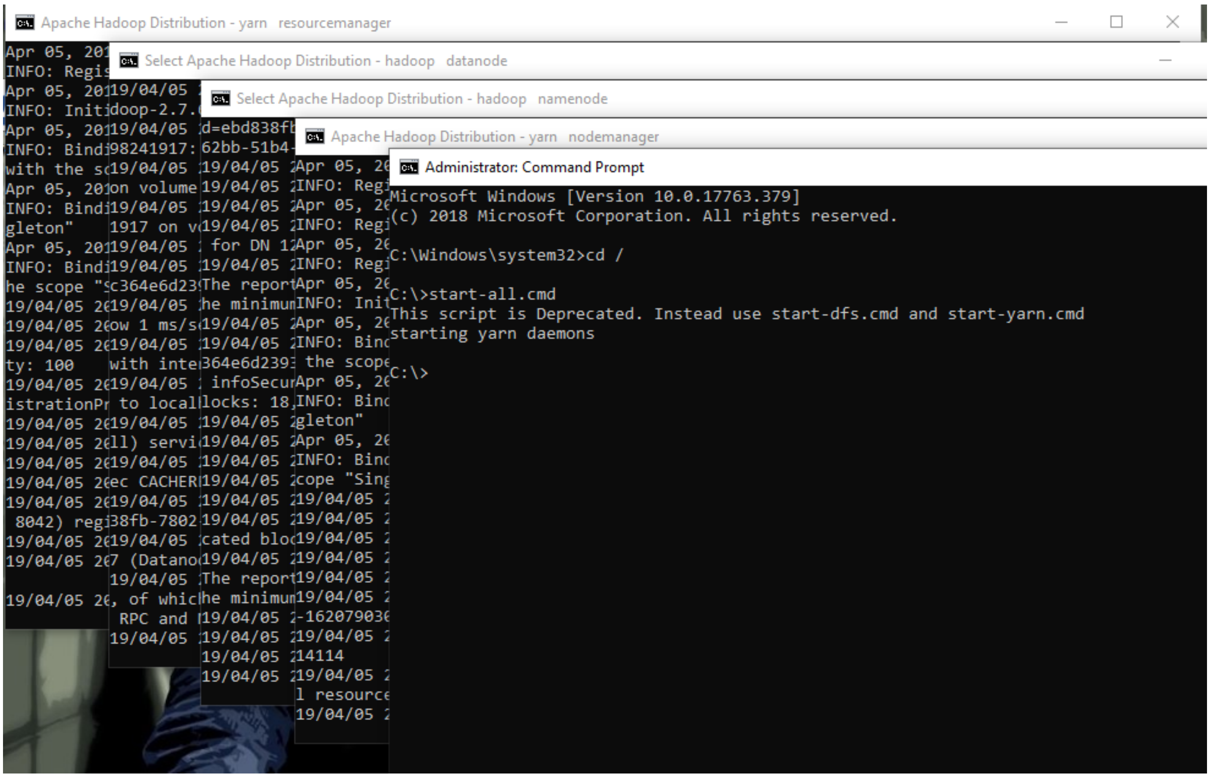


```python
from sklearn.feature_extraction.text import CountVectorizer
import pandas as pd
from sklearn.metrics.pairwise import cosine_similarity

f = open('Doc1.txt')
doc1 = str(f.read())
#doc1="Apples are tasty,Apples are yummy"
f = open('Doc2.txt')
doc2 = str(f.read())
#doc2="Apples are yummy, Apples are red"

documents = [doc1, doc2]

count_vectorizer = CountVectorizer()
sparse_matrix = count_vectorizer.fit_transform(documents)

#print(sparse_matrix)

doc_term_matrix = sparse_matrix.todense()

#print(doc_term_matrix)
df = pd.DataFrame(doc_term_matrix,
                  columns=count_vectorizer.get_feature_names(),
                  index=['doc1', 'doc2'])
print(df)

print(cosine_similarity(df[0:1], df))
```

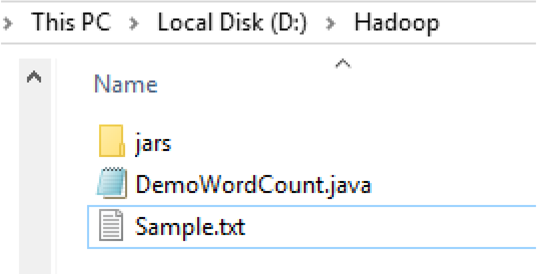
```
      apples  are  red  tasty  yummy
doc1       2    2    0      1      1
doc2       2    2    1      0      1
[[1.  0.9]]
```

Practical 5

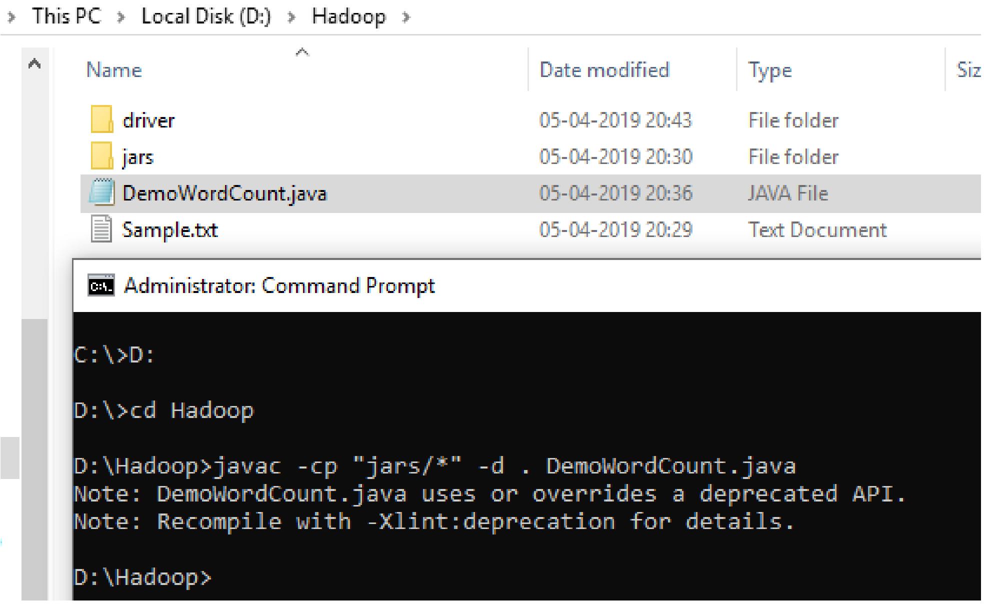
Simple map reduce program


Apache Hadoop Distribution - yarn resourcemanager
Select Apache Hadoop Distribution - hadoop datanode
Select Apache Hadoop Distribution - hadoop namenode
Apache Hadoop Distribution - yarn nodemanager
Administrator: Command Prompt

```
Microsoft Windows [Version 10.0.17763.379]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>cd /

C:\>start-all.cmd
This script is Deprecated. Instead use start-dfs.cmd and start-yarn.cmd
starting yarn daemons

C:\>
```



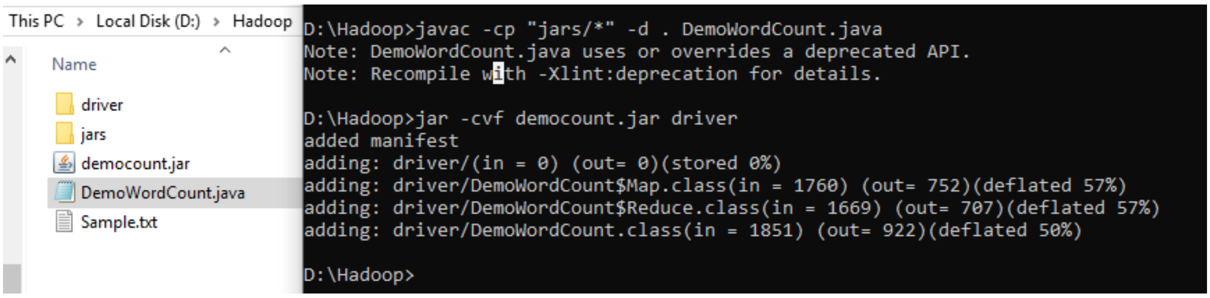
jars
DemoWordCount.java
Sample.txt


This PC > Local Disk (D:) > Hadoop >

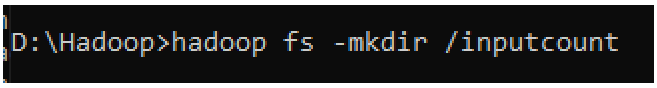
Name	Date modified	Type	Siz
driver	05-04-2019 20:43	File folder	
jars	05-04-2019 20:30	File folder	
DemoWordCount.java	05-04-2019 20:36	JAVA File	
Sample.txt	05-04-2019 20:29	Text Document	

Administrator: Command Prompt

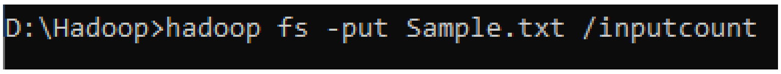
```
C:\>D:

D:\>cd Hadoop

D:\Hadoop>javac -cp "jars/*" -d . DemoWordCount.java
Note: DemoWordCount.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

D:\Hadoop>
```



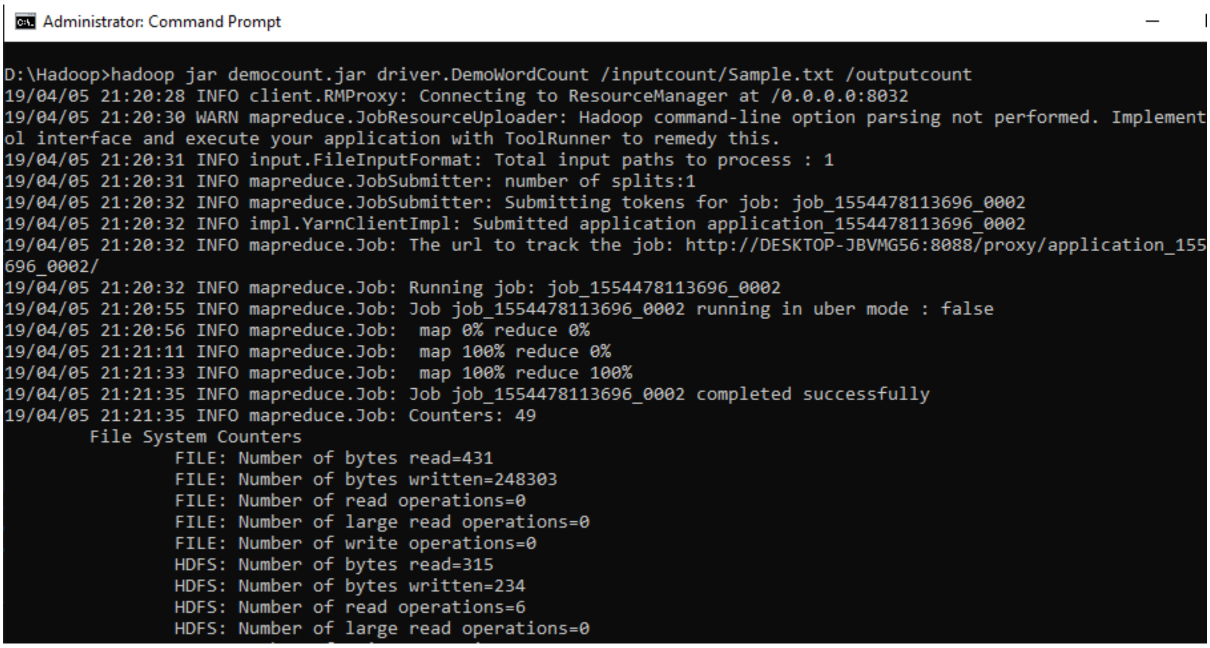
```
D:\Hadoop>javac -cp "jars/*" -d . DemoWordCount.java
Note: DemoWordCount.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

D:\Hadoop>jar -cvf democount.jar driver
added manifest
adding: driver/(in = 0) (out= 0)(stored 0%)
adding: driver/DemoWordCount$Map.class(in = 1760) (out= 752)(deflated 57%)
adding: driver/DemoWordCount$Reduce.class(in = 1669) (out= 707)(deflated 57%)
adding: driver/DemoWordCount.class(in = 1851) (out= 922)(deflated 50%)

D:\Hadoop>
```



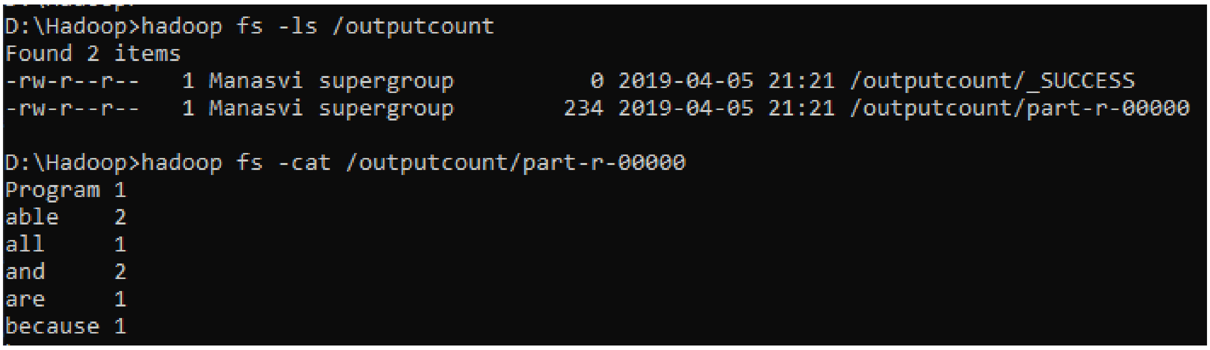
```
D:\Hadoop>hadoop fs -mkdir /inputcount
```



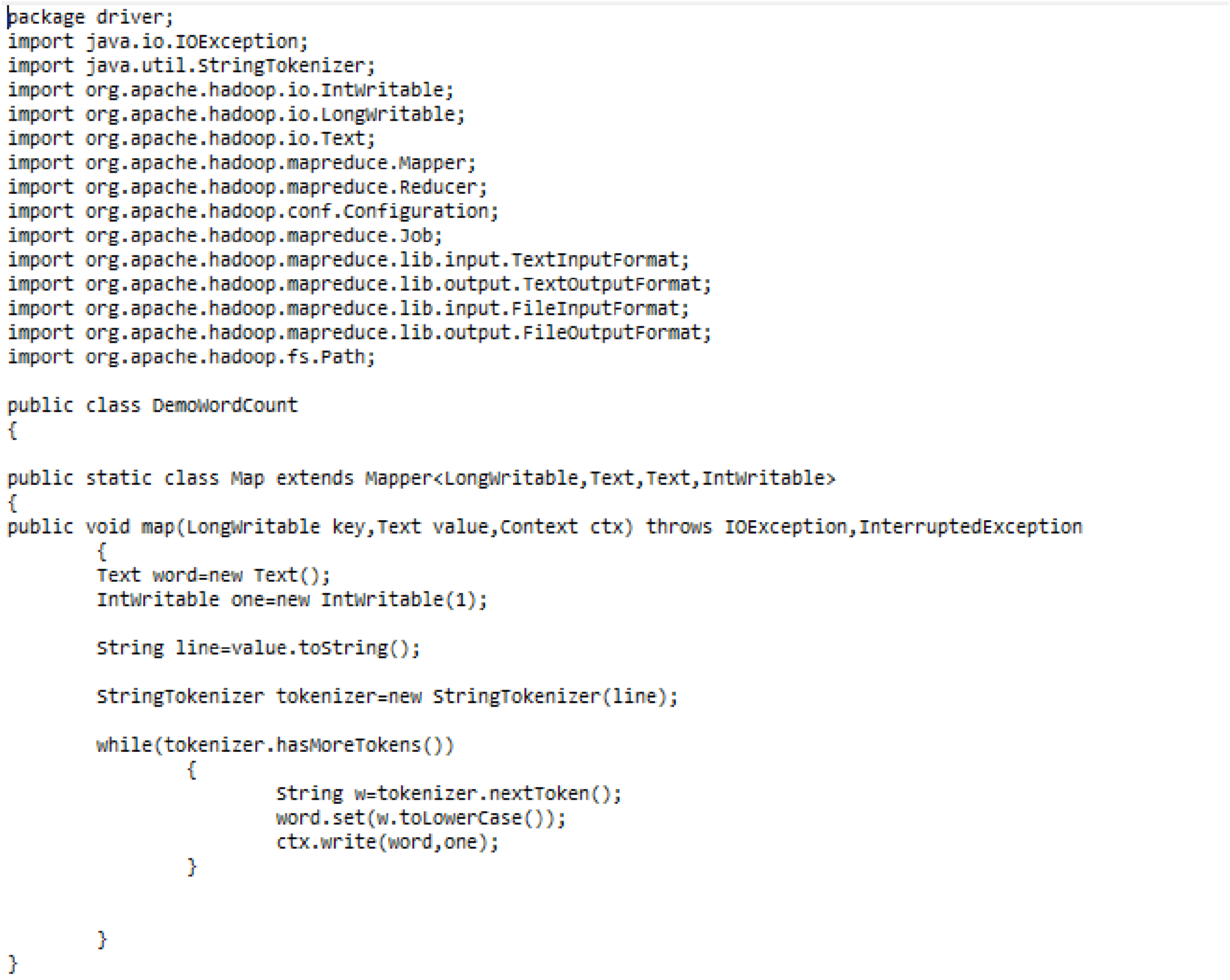
```
D:\Hadoop>hadoop fs -put Sample.txt /inputcount
```



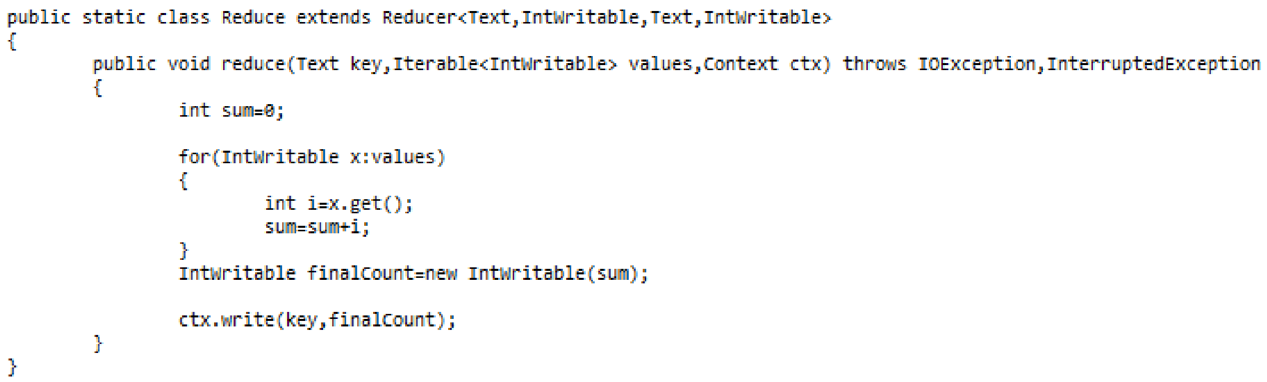
```
D:\Hadoop>hadoop jar democount.jar driver.DemoWordCount /inputcount/Sample.txt /outputcount
19/04/05 21:20:28 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
19/04/05 21:20:30 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement
ol interface and execute your application with ToolRunner to remedy this.
19/04/05 21:20:31 INFO input.FileInputFormat: Total input paths to process : 1
19/04/05 21:20:31 INFO mapreduce.JobSubmitter: number of splits:1
19/04/05 21:20:32 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1554478113696_0002
19/04/05 21:20:32 INFO impl.YarnClientImpl: Submitted application application_1554478113696_0002
19/04/05 21:20:32 INFO mapreduce.Job: The url to track the job: http://DESKTOP-JBVMG56:8088/proxy/application_155
696_0002/
19/04/05 21:20:32 INFO mapreduce.Job: Running job: job_1554478113696_0002
19/04/05 21:20:55 INFO mapreduce.Job: Job job_1554478113696_0002 running in uber mode : false
19/04/05 21:20:56 INFO mapreduce.Job:  map 0% reduce 0%
19/04/05 21:21:11 INFO mapreduce.Job:  map 100% reduce 0%
19/04/05 21:21:33 INFO mapreduce.Job:  map 100% reduce 100%
19/04/05 21:21:35 INFO mapreduce.Job: Job job_1554478113696_0002 completed successfully
19/04/05 21:21:35 INFO mapreduce.Job: Counters: 49
        File System Counters
                FILE: Number of bytes read=431
                FILE: Number of bytes written=248303
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=315
                HDFS: Number of bytes written=234
                HDFS: Number of read operations=6
                HDFS: Number of large read operations=0
```



```
D:\Hadoop>hadoop fs -ls /outputcount
Found 2 items
-rw-r--r--   1 Manasvi supergroup          0 2019-04-05 21:21 /outputcount/_SUCCESS
-rw-r--r--   1 Manasvi supergroup        234 2019-04-05 21:21 /outputcount/part-r-00000

D:\Hadoop>hadoop fs -cat /outputcount/part-r-00000
Program 1
able    2
all     1
and     2
are     1
because 1
```



```
package driver;
import java.io.IOException;
import java.util.StringTokenizer;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.lib.input.TextInputFormat;
import org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.fs.Path;

public class DemoWordCount
{

public static class Map extends Mapper<LongWritable,Text,Text,IntWritable>
{
public void map(LongWritable key,Text value,Context ctx) throws IOException,InterruptedException
        {
        Text word=new Text();
        IntWritable one=new IntWritable(1);

        String line=value.toString();

        StringTokenizer tokenizer=new StringTokenizer(line);

        while(tokenizer.hasMoreTokens())
                {
                        String w=tokenizer.nextToken();
                        word.set(w.toLowerCase());
                        ctx.write(word,one);
                }


        }
}
```

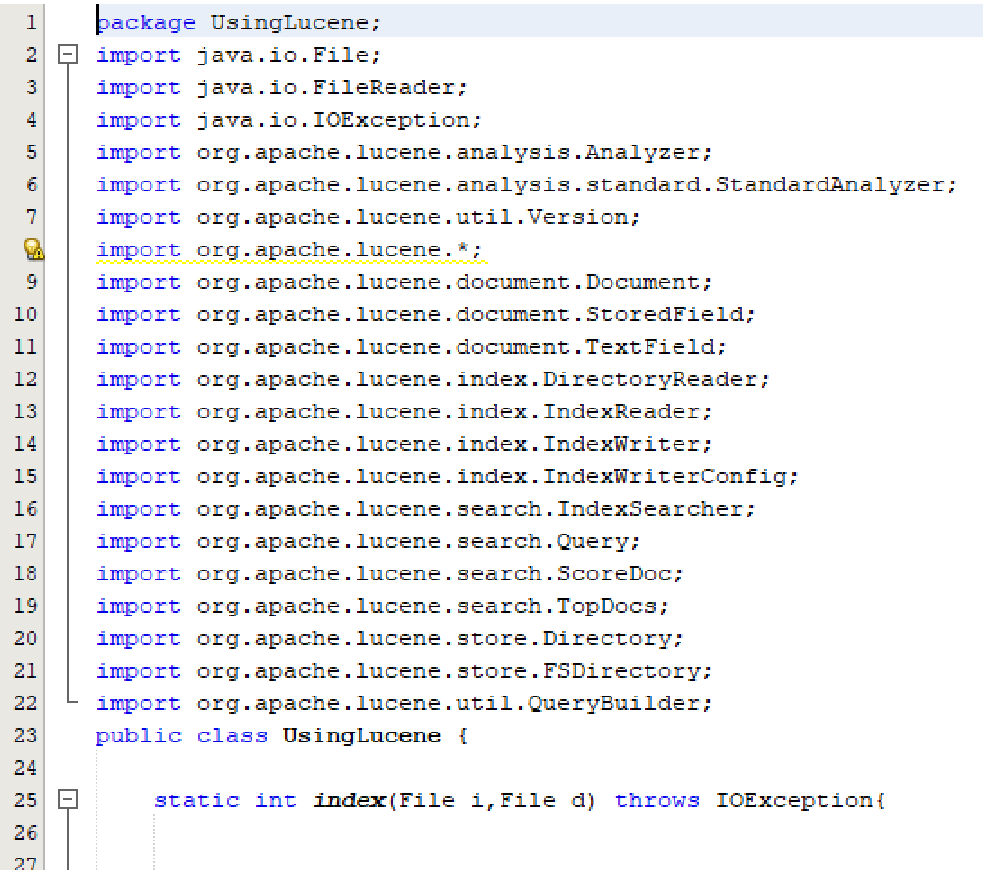


```
public static class Reduce extends Reducer<Text,IntWritable,Text,IntWritable>
{
        public void reduce(Text key,Iterable<IntWritable> values,Context ctx) throws IOException,InterruptedException
        {
                int sum=0;

                for(IntWritable x:values)
                {
                        int i=x.get();
                        sum=sum+i;
                }
                IntWritable finalCount=new IntWritable(sum);

                ctx.write(key,finalCount);
        }
}
```

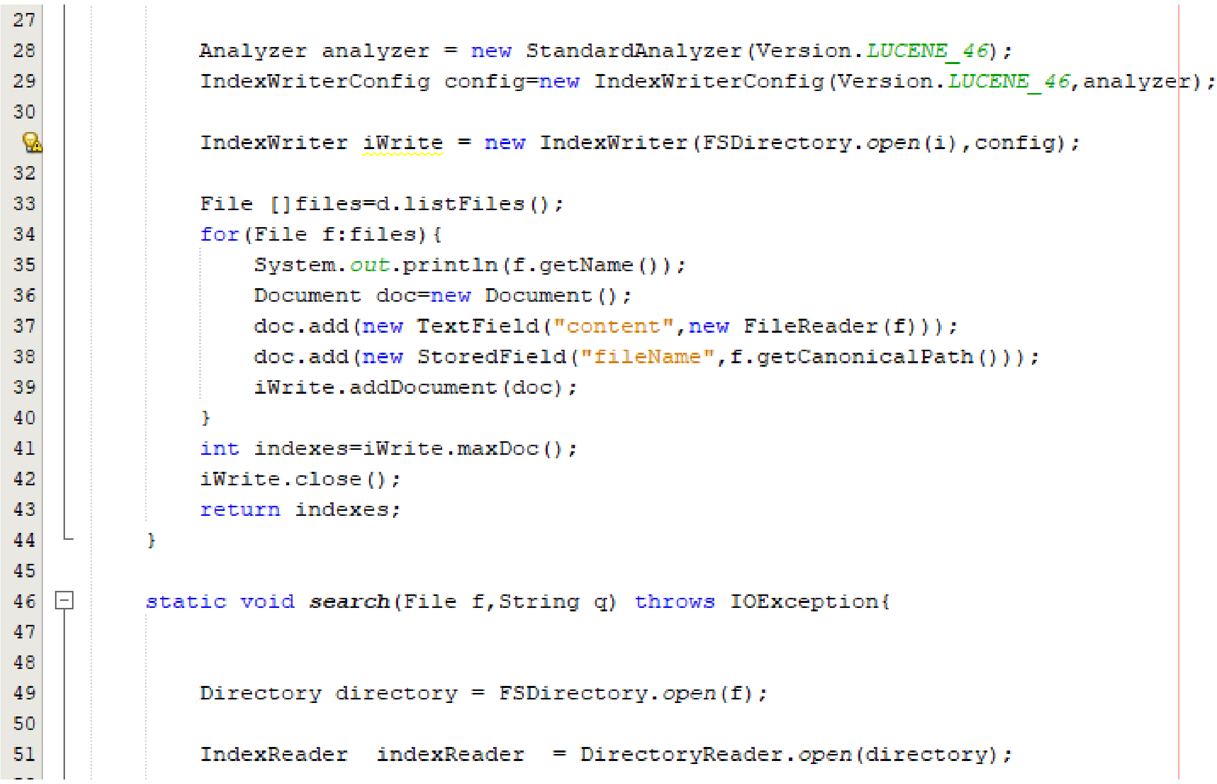
```
public static void main(String[] args) throws Exception
        {
                Configuration conf= new Configuration();

                Job job = new Job(conf,"My Word Count Program");
                job.setJarByClass(DemoWordCount.class);
                job.setMapperClass(Map.class);
                job.setReducerClass(Reduce.class);
                job.setOutputKeyClass(Text.class);
                job.setOutputValueClass(IntWritable.class);
                job.setInputFormatClass(TextInputFormat.class);
                job.setOutputFormatClass(TextOutputFormat.class);

                Path outputPath = new Path(args[1]);
                FileInputFormat.addInputPath(job, new Path(args[0]));
                FileOutputFormat.setOutputPath(job, new Path(args[1]));
                outputPath.getFileSystem(conf).delete(outputPath);

                //exiting the job only if the flag value becomes false
                System.exit(job.waitForCompletion(true) ? 0 : 1);
        }
}
```

Practical 6

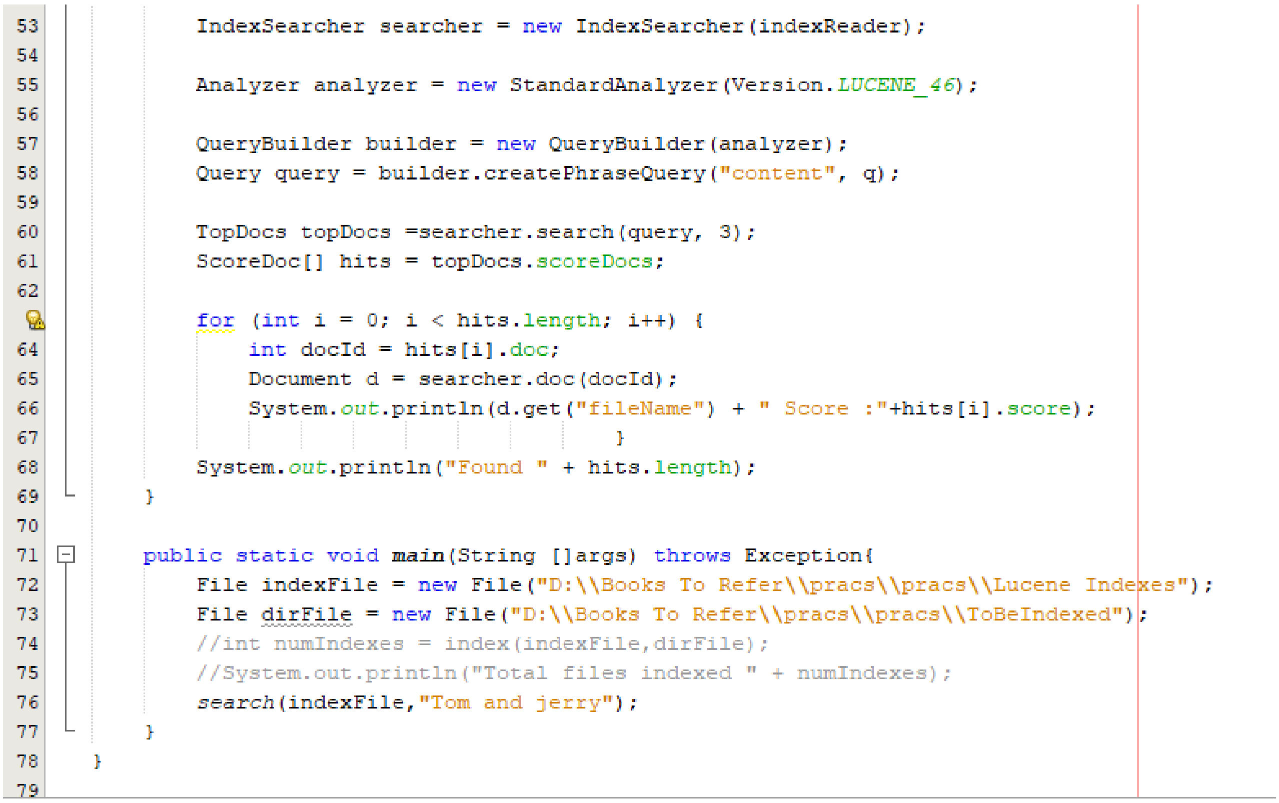
IR system using lucene



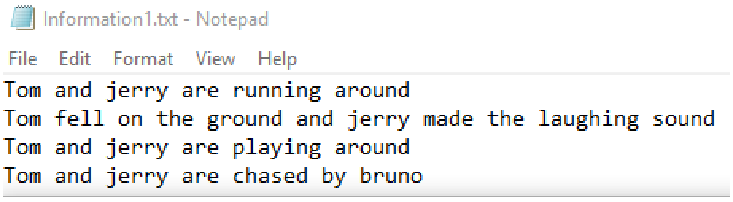
```java
package UsingLucene;
import java.io.File;
import java.io.FileReader;
import java.io.IOException;
import org.apache.lucene.analysis.Analyzer;
import org.apache.lucene.analysis.standard.StandardAnalyzer;
import org.apache.lucene.util.Version;
import org.apache.lucene.*;
import org.apache.lucene.document.Document;
import org.apache.lucene.document.StoredField;
import org.apache.lucene.document.TextField;
import org.apache.lucene.index.DirectoryReader;
import org.apache.lucene.index.IndexReader;
import org.apache.lucene.index.IndexWriter;
import org.apache.lucene.index.IndexWriterConfig;
import org.apache.lucene.search.IndexSearcher;
import org.apache.lucene.search.Query;
import org.apache.lucene.search.ScoreDoc;
import org.apache.lucene.search.TopDocs;
import org.apache.lucene.store.Directory;
import org.apache.lucene.store.FSDirectory;
import org.apache.lucene.util.QueryBuilder;
public class UsingLucene {

    static int index(File i,File d) throws IOException{

```



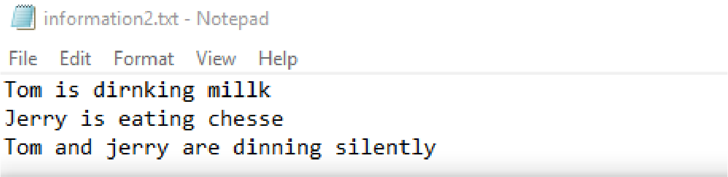
```java

        Analyzer analyzer = new StandardAnalyzer(Version.LUCENE_46);
        IndexWriterConfig config=new IndexWriterConfig(Version.LUCENE_46,analyzer);

        IndexWriter iWrite = new IndexWriter(FSDirectory.open(i),config);

        File []files=d.listFiles();
        for(File f:files){
            System.out.println(f.getName());
            Document doc=new Document();
            doc.add(new TextField("content",new FileReader(f)));
            doc.add(new StoredField("fileName",f.getCanonicalPath()));
            iWrite.addDocument(doc);
        }
        int indexes=iWrite.maxDoc();
        iWrite.close();
        return indexes;
    }

    static void search(File f,String q) throws IOException{


        Directory directory = FSDirectory.open(f);

        IndexReader  indexReader  = DirectoryReader.open(directory);
```



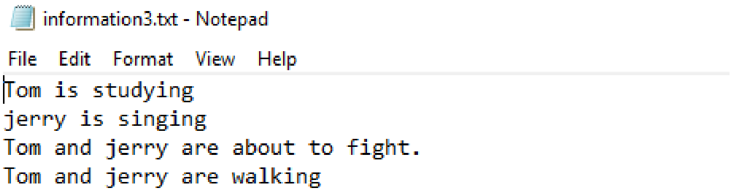
```java
            IndexSearcher searcher = new IndexSearcher(indexReader);

            Analyzer analyzer = new StandardAnalyzer(Version.LUCENE_46);

            QueryBuilder builder = new QueryBuilder(analyzer);
            Query query = builder.createPhraseQuery("content", q);

            TopDocs topDocs =searcher.search(query, 3);
            ScoreDoc[] hits = topDocs.scoreDocs;

            for (int i = 0; i < hits.length; i++) {
                int docId = hits[i].doc;
                Document d = searcher.doc(docId);
                System.out.println(d.get("fileName") + " Score :"+hits[i].score);
                                            }
            System.out.println("Found " + hits.length);
        }

        public static void main(String []args) throws Exception{
            File indexFile = new File("D:\\Books To Refer\\pracs\\pracs\\Lucene Indexes");
            File dirFile = new File("D:\\Books To Refer\\pracs\\pracs\\ToBeIndexed");
            //int numIndexes = index(indexFile,dirFile);
            //System.out.println("Total files indexed " + numIndexes);
            search(indexFile,"Tom and jerry");
        }
    }
```



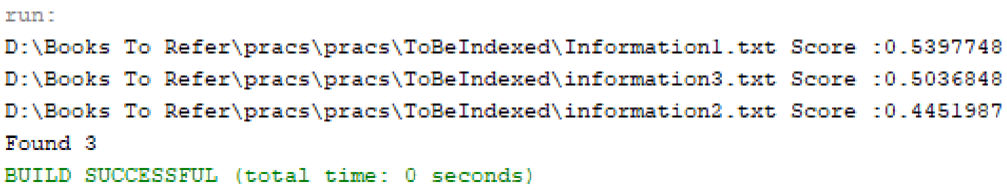
```
Tom and jerry are running around
Tom fell on the ground and jerry made the laughing sound
Tom and jerry are playing around
Tom and jerry are chased by bruno
```



```
Tom is dirnking millk
Jerry is eating chesse
Tom and jerry are dinning silently
```



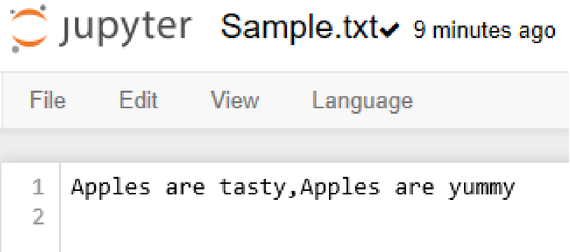
```
Tom is studying
jerry is singing
Tom and jerry are about to fight.
Tom and jerry are walking
```



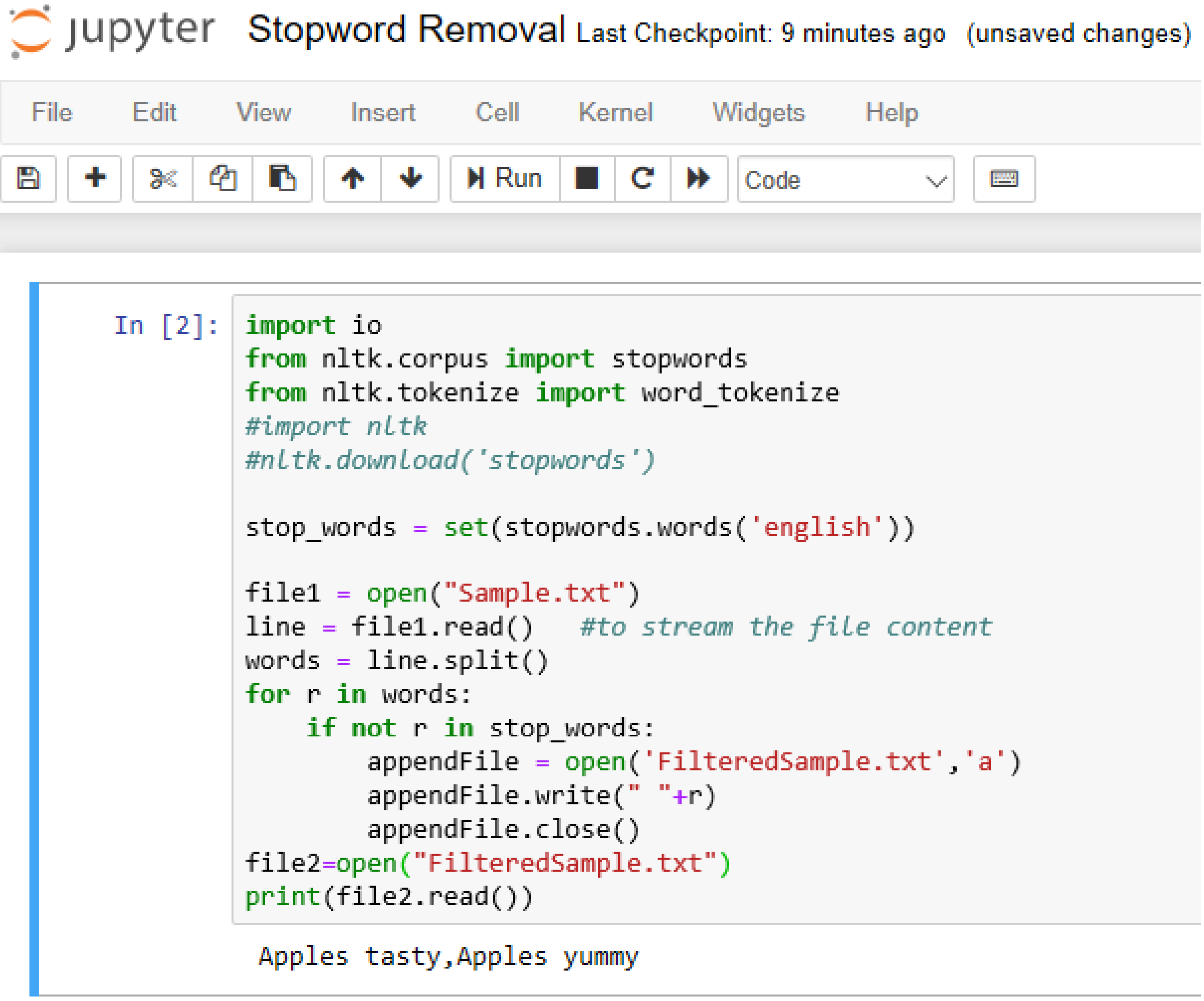
```
run:
D:\Books To Refer\pracs\pracs\ToBeIndexed\Information1.txt Score :0.5397748
D:\Books To Refer\pracs\pracs\ToBeIndexed\information3.txt Score :0.5036848
D:\Books To Refer\pracs\pracs\ToBeIndexed\information2.txt Score :0.4451987
Found 3
BUILD SUCCESSFUL (total time: 0 seconds)
```

Practical 7

Program to remove stop words

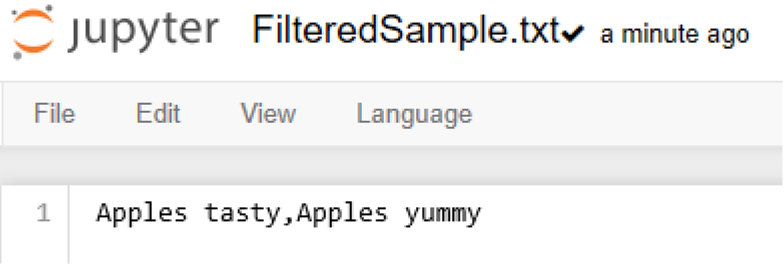


```
Apples are tasty,Apples are yummy
```


jupyter
Stopword Removal Last Checkpoint: 9 minutes ago (unsaved changes)
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Run
Code
In [2]:

```
import io
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
#import nltk
#nltk.download('stopwords')

stop_words = set(stopwords.words('english'))

file1 = open("Sample.txt")
line = file1.read()    #to stream the file content
words = line.split()
for r in words:
    if not r in stop_words:
        appendFile = open('FilteredSample.txt','a')
        appendFile.write(" "+r)
        appendFile.close()
file2=open("FilteredSample.txt")
print(file2.read())
```

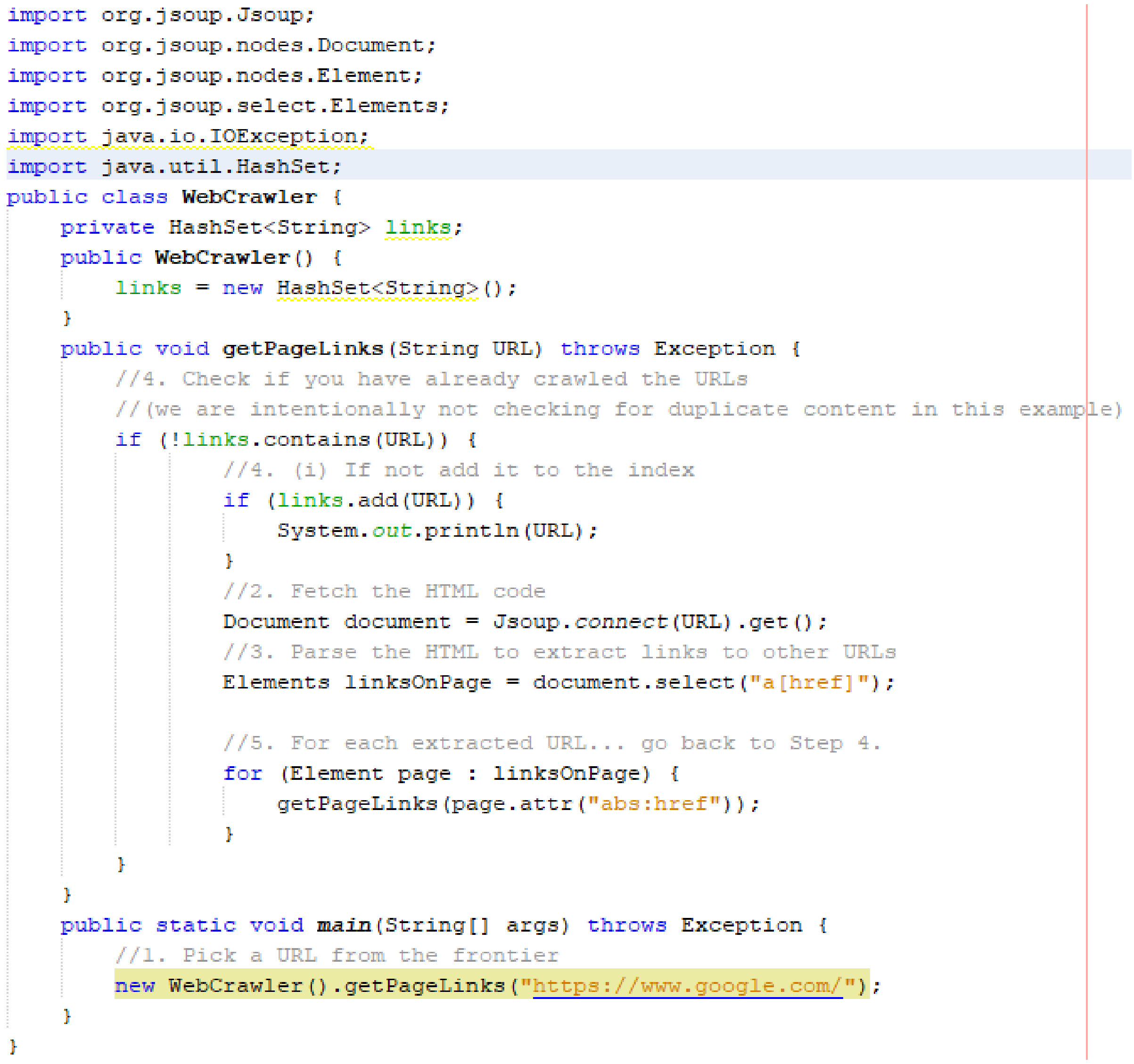
```
 Apples tasty,Apples yummy
```



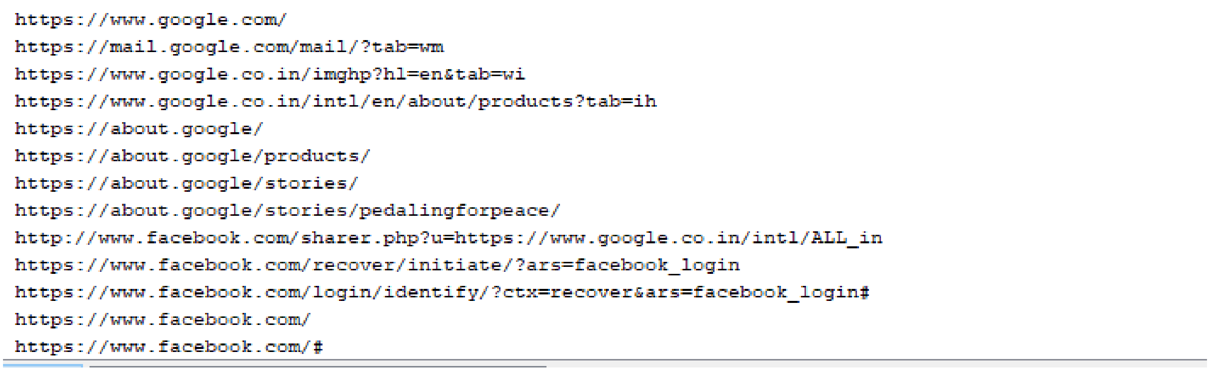
```
 Apples tasty,Apples yummy
```

Practical 8

Program for web crawler



```
import org.jsoup.Jsoup;
import org.jsoup.nodes.Document;
import org.jsoup.nodes.Element;
import org.jsoup.select.Elements;
import java.io.IOException;
import java.util.HashSet;
public class WebCrawler {
    private HashSet<String> links;
    public WebCrawler() {
        links = new HashSet<String>();
    }
    public void getPageLinks(String URL) throws Exception {
        //4. Check if you have already crawled the URLs
        //(we are intentionally not checking for duplicate content in this example)
        if (!links.contains(URL)) {
                //4. (i) If not add it to the index
                if (links.add(URL)) {
                    System.out.println(URL);
                }
                //2. Fetch the HTML code
                Document document = Jsoup.connect(URL).get();
                //3. Parse the HTML to extract links to other URLs
                Elements linksOnPage = document.select("a[href]");

                //5. For each extracted URL... go back to Step 4.
                for (Element page : linksOnPage) {
                    getPageLinks(page.attr("abs:href"));
                }
        }
    }
    public static void main(String[] args) throws Exception {
        //1. Pick a URL from the frontier
        new WebCrawler().getPageLinks("https://www.google.com/");
    }
}
```



```
https://www.google.com/
https://mail.google.com/mail/?tab=wm
https://www.google.co.in/imghp?hl=en&tab=wi
https://www.google.co.in/intl/en/about/products?tab=ih
https://about.google/
https://about.google/products/
https://about.google/stories/
https://about.google/stories/pedalingforpeace/
http://www.facebook.com/sharer.php?u=https://www.google.co.in/intl/ALL_in
https://www.facebook.com/recover/initiate/?ars=facebook_login
https://www.facebook.com/login/identify/?ctx=recover&ars=facebook_login#
https://www.facebook.com/
https://www.facebook.com/#
```