**Task 1:**

**Write a Java program, to take an HDFS Path as input and display all the files and sub-directories in thatHDFS path.**

package mapreduce.demo;  
  
import java.io.\*;  
import org.apache.hadoop.conf.Configuration;  
import org.apache.hadoop.fs.FileSystem;  
import org.apache.hadoop.fs.Path;  
import org.apache.hadoop.fs.FileStatus;  
  
public class FileListing {  
    public static void main(String[] args) {  
        if (args.length != 1) {  
            System.out.println("Pass one argument");  
            System.exit(1);  
        }  
          
        Path path = new Path(args[0]);  
          
        try  
        {  
            Configuration conf = new Configuration();  
            FileSystem fileSystem = FileSystem.get(path.toUri(), conf);  
            FileStatus[] fileStatus=fileSystem.listStatus(path);  
              
            for (FileStatus fStat : fileStatus) {  
                if (fStat.isDirectory()) {  
                    System.out.println("Directory: " + fStat.getPath());  
                }  
                else if (fStat.isFile()) {  
                    System.out.println("File: " + fStat.getPath());  
                }  
                else if (fStat.isSymlink()) {  
                    System.out.println("Symlink: " + fStat.getPath());  
                }  
            }  
  
        }  
        catch (IOException e)  
        {  
            e.printStackTrace();  
        }  
    }  
}

OUTPUT:

[acadgild@localhost training]$ hadoop jar filelisting.jar /user/acadgild/company

File: hdfs://localhost:9000/user/acadgild/company/\_SUCCESS  
File: hdfs://localhost:9000/user/acadgild/company/part-m-00000

**Task 2:**

**Modify the previous program to list all the files and sub-directories in the HDFS path recursively.**

package list;  
  
import java.io.IOException;  
  
import java.net.URISyntaxException;  
  
  
import org.apache.hadoop.conf.Configuration;  
import org.apache.hadoop.fs.FileStatus;  
import org.apache.hadoop.fs.FileSystem;  
import org.apache.hadoop.fs.Path;  
  
public class aaa {  
  public static void main(String[] args) throws IOException, URISyntaxException  
  {  
        
      Path path = new Path(args[0]);  
      fileStatusFunc(path);  
      
    }  
    
  
public static void fileStatusFunc(Path path) throws IOException {  
    Configuration conf = new Configuration();  
      
    FileSystem filesystem = FileSystem.get(path.toUri(), conf);  
          
      
    FileStatus[] fileStatus = filesystem.listStatus(path);  
         
     
      
    for(FileStatus eachFileStatus : fileStatus){  
          
        if(eachFileStatus.isDirectory() == true)  
        {  
        System.out.println("\*\*\*\*\* Contents of the Directory \*\*\*\*\*");  
        System.out.println("\t Directory: \t" + eachFileStatus.getPath());  
        //recursive call  
        fileStatusFunc(eachFileStatus.getPath());  
        }  
        if(eachFileStatus.isFile() == true){  
        System.out.println("File: \t" + eachFileStatus.getPath());  
        }  
        if(eachFileStatus.isSymlink() == true){  
        System.out.println("Link: \t" + eachFileStatus.getPath());  
        }  
    }  
    }  
  
      
}

OUTPUT:

[acadgild@localhost training]$ hadoop jar aaa1.jar /user/acadgild/hadoop

\*\*\*\*\* Contents of the Directory \*\*\*\*\*  
     Directory:     hdfs://localhost:9000/user/acadgild/hadoop/adirectory  
File:     hdfs://localhost:9000/user/acadgild/hadoop/adirectory/emp\_details.txt  
File:     hdfs://localhost:9000/user/acadgild/hadoop/adirectory/wordcount.txt  
File:     hdfs://localhost:9000/user/acadgild/hadoop/max-temp.txt  
File:     hdfs://localhost:9000/user/acadgild/hadoop/session2.txt  
File:     hdfs://localhost:9000/user/acadgild/hadoop/wordcount.txt  
File:     hdfs://localhost:9000/user/acadgild/hadoop/wordcount1.txt

[acadgild@localhost training]$ hadoop jar aaa1.jar /user/acadgild/company

File:     hdfs://localhost:9000/user/acadgild/company/\_SUCCESS  
File:     hdfs://localhost:9000/user/acadgild/company/part-m-00000

**Task 3:**

**Modify the previous program to take multiple HDFS paths (separated by space) and list all the files andsub-directories in those HDFS paths recursively.**

package list;  
  
  
import java.io.IOException;  
  
import java.net.URISyntaxException;  
  
  
import org.apache.hadoop.conf.Configuration;  
import org.apache.hadoop.fs.FileStatus;  
import org.apache.hadoop.fs.FileSystem;  
import org.apache.hadoop.fs.Path;  
  
public class filemr {  
  public static void main(String[] args) throws IOException, URISyntaxException  
  {  
      if (args.length >0)  
  
          for (String val :args)  
          {  
          Path path = new Path(val);  
          fileStatusFunc(path);  
      }  
      
    }  
    
  
public static void fileStatusFunc(Path path) throws IOException {  
    Configuration conf = new Configuration();  
      
    FileSystem filesystem = FileSystem.get(path.toUri(), conf);  
          
      
    FileStatus[] fileStatus = filesystem.listStatus(path);  
         
     
      
    for(FileStatus eachFileStatus : fileStatus){  
          
        if(eachFileStatus.isDirectory() == true)  
        {  
        System.out.println("\*\*\*\*\* Contents of the Directory \*\*\*\*\*");  
        System.out.println("\t Directory: \t" + eachFileStatus.getPath());  
        //recursive call  
        fileStatusFunc(eachFileStatus.getPath());  
        }  
        if(eachFileStatus.isFile() == true){  
        System.out.println("File: \t" + eachFileStatus.getPath());  
        }  
        if(eachFileStatus.isSymlink() == true){  
        System.out.println("Link: \t" + eachFileStatus.getPath());  
        }  
    }  
    }  
  
      
}

OUTPUT:

[acadgild@localhost training]$ hadoop jar filemr.jar /user/acadgild/company /user/acadgild/hadoop  
  
File:     hdfs://localhost:9000/user/acadgild/company/\_SUCCESS  
File:     hdfs://localhost:9000/user/acadgild/company/part-m-00000  
\*\*\*\*\* Contents of the Directory \*\*\*\*\*  
     Directory:     hdfs://localhost:9000/user/acadgild/hadoop/adirectory  
File:     hdfs://localhost:9000/user/acadgild/hadoop/adirectory/emp\_details.txt  
File:     hdfs://localhost:9000/user/acadgild/hadoop/adirectory/wordcount.txt  
File:     hdfs://localhost:9000/user/acadgild/hadoop/max-temp.txt  
File:     hdfs://localhost:9000/user/acadgild/hadoop/session2.txt  
File:     hdfs://localhost:9000/user/acadgild/hadoop/wordcount.txt  
File:     hdfs://localhost:9000/user/acadgild/hadoop/wordcount1.txt