/\*

Name: Mr. Jain Samkitkumar Hasmukhlal

Roll no: 20 BEIT

AIM: Implementation of Conflation Algorithm.

\*/

import java.lang.\*;

import java.io.\*;

import java.util.\*;

class Confletion

{

String high\_frq\_words[] = {"i","am","cannot","into","our","thus" ,"about","is","to" ,"above", "you",

"too","after","a","is" ,"are","was","were","have","has","had","of","there","now","whom","next", "but",

"next","than","still","he","she","it","they","them","their","no","as","be","the","his","her","if"," in" ,"this"};

String str\_token="",str\_token\_cmp="";

int i=0,flg=0,cnt=0,j=0,spe\_cnt=0;;

BufferedReader br=null,br2=null;

FileWriter fw=null;

//----------------------------------Step One : removing high frequency words-------------------------------

void confletionStepOne(String filename) throws Exception

{

br=new BufferedReader(new FileReader(filename) );

fw = new FileWriter("Step1.txt");

while((str\_token = br.readLine()) != null )

{

StringTokenizer st =new StringTokenizer(str\_token," ,.");

while(st.hasMoreTokens())

{

String str\_tmp = st.nextToken();

i=0;

while(i<high\_frq\_words.length)

{

if(str\_tmp.equals(high\_frq\_words[i++]))

{

flg=1;

break;

}

else

flg=0;

}

if(flg==0)

{

// System.out.println(str\_tmp);

fw.write(str\_tmp + " ");

}

}

}

fw.close();

}

//--------------------------------------------------Step One Finished-----------------------------------------------

//----------------------------------------------Step Two : Suffix stripping-----------------------------------------

void confletionStepTwo(String filename) throws Exception

{

br=new BufferedReader(new FileReader(filename) );

fw = new FileWriter("Step2.txt");

while((str\_token = br.readLine()) != null )

{

StringTokenizer st =new StringTokenizer(str\_token," ");

while(st.hasMoreTokens())

{

String str\_tmp = st.nextToken();

if (str\_tmp.endsWith("ate"))

str\_tmp= str\_tmp.replace("ate","");

if (str\_tmp.endsWith("iveness"))

str\_tmp=str\_tmp.replace("iveness","ive");

if (str\_tmp.endsWith("fulness"))

str\_tmp= str\_tmp.replace("fulness","ful");

if (str\_tmp.endsWith("ed"))

str\_tmp= str\_tmp.replace("ed","");

if (str\_tmp.endsWith("ousness"))

str\_tmp= str\_tmp.replace("ousness","ous");

if (str\_tmp.endsWith("sses"))

str\_tmp= str\_tmp.replace("sses","ss");

if (str\_tmp.endsWith("ness"))

str\_tmp= str\_tmp.replace("ness","");

if (str\_tmp.endsWith("ies"))

str\_tmp= str\_tmp.replace("ies","y");

if (str\_tmp.endsWith("s"))

str\_tmp= str\_tmp.replace("s","");

if (str\_tmp.endsWith("eed"))

str\_tmp= str\_tmp.replace("eed","ee");

if (str\_tmp.endsWith("ational"))

str\_tmp= str\_tmp.replace("ational","ate");

if (str\_tmp.endsWith("tional"))

str\_tmp= str\_tmp.replace("tional","tion");

if (str\_tmp.endsWith("ization"))

str\_tmp= str\_tmp.replace("ization","ize");

if (str\_tmp.endsWith("ation"))

str\_tmp= str\_tmp.replace("ation","ate");

if (str\_tmp.endsWith("ator"))

str\_tmp= str\_tmp.replace("ator","ate");

if (str\_tmp.endsWith("iviti"))

str\_tmp= str\_tmp.replace("iviti","ive");

if (str\_tmp.endsWith("biliti"))

str\_tmp= str\_tmp.replace("biliti","ble");

if (str\_tmp.endsWith("icate"))

str\_tmp= str\_tmp.replace("icate","ic");

if (str\_tmp.endsWith("alize"))

str\_tmp= str\_tmp.replace("alize","al");

if (str\_tmp.endsWith("iciti"))

str\_tmp= str\_tmp.replace("iciti","ic");

if (str\_tmp.endsWith("ical"))

str\_tmp= str\_tmp.replace("ical","ic");

if (str\_tmp.endsWith("ative"))

str\_tmp= str\_tmp.replace("ative","");

if (str\_tmp.endsWith("ful"))

str\_tmp= str\_tmp.replace("ful","");

if (str\_tmp.endsWith("ism"))

str\_tmp= str\_tmp.replace("ism","");

if (str\_tmp.endsWith("ous"))

str\_tmp= str\_tmp.replace("ous","");

if (str\_tmp.endsWith("ive"))

str\_tmp= str\_tmp.replace("ive","");

if (str\_tmp.endsWith("ize"))

str\_tmp= str\_tmp.replace("ize","");

if (str\_tmp.endsWith("ship"))

str\_tmp= str\_tmp.replace("ship","");

if (str\_tmp.endsWith("al"))

str\_tmp= str\_tmp.replace("al","");

if (str\_tmp.endsWith("ance"))

str\_tmp= str\_tmp.replace("ance","");

if (str\_tmp.endsWith("ence"))

str\_tmp= str\_tmp.replace("ence","");

if (str\_tmp.endsWith("able"))

str\_tmp= str\_tmp.replace("able","");

if (str\_tmp.endsWith("ant"))

str\_tmp= str\_tmp.replace("ant","");

if (str\_tmp.endsWith("ment"))

str\_tmp= str\_tmp.replace("ment","");

// System.out.println(str\_tmp);

fw.write(str\_tmp + " ");

}

}

fw.close();

}

//---------------------------------------------------Step Two Finished----------------------------------------------

//----------------------------------------------Step Three : ----------------------------------------

void confletionStepThree(String filename) throws Exception

{

String str\_one="",str\_two="";

br=new BufferedReader(new FileReader(filename) );

ArrayList al=new ArrayList();

fw = new FileWriter("Step3.txt");

while((str\_token = br.readLine()) != null )

{

StringTokenizer st1 =new StringTokenizer(str\_token," ");

while(st1.hasMoreTokens())

{

al.add(st1.nextToken());

}

}

Object o[]=al.toArray();

Object o1[]=new Object[o.length];

//-------Removing duplicate element

for(i=0;i<o.length;i++)

{

for(j=0;j<cnt;j++)

if(o[i].equals(o1[j]))

break;

if(j==cnt)

{

o1[cnt] = o[i];

cnt++;

}

}

//-----------------------------------------------

System.out.println("----------------------------------------");

System.out.println("Frequency \t Word");

System.out.println("----------------------------------------");

fw.write("----------------------------------------\n");

fw.write("\nFrequency \t Word \n");

fw.write("\n---------------------------------------- \n");

for(i=0;i<cnt;i++)

{

spe\_cnt=0;

for(j=0;j<o.length;j++)

if(o1[i].equals(o[j]))

spe\_cnt++;

System.out.println(spe\_cnt+"\t\t"+o1[i]);

fw.write("\n"+spe\_cnt +"\t\t"+o1[i]+"\n");

}

fw.close();

}

//---------------------------------------------------Step Three Finished----------------------------------------------

}

public class ConfTest

{

public static void main(String args[]) throws Exception

{

Confletion cf=new Confletion();

cf.confletionStepOne("input.txt");

cf.confletionStepTwo("step1.txt");

cf.confletionStepThree("step2.txt");

}}

/\* input.txt \*/

this this

is not a regular file regular

it contains some regular data

about this file itself

serial

factorial

equal

factual

Output

D:>javac ConfTest.java

Note: ConfTest.java uses unchecked or unsafe operations.

Note: Recompile with -Xlint:unchecked for details.

D:>java ConfTest

----------------------------------------

Frequency Word

----------------------------------------

1 not

3 regular

2 file

1 contain

1 some

1 data

1 itself

1 seri

1 factori

1 equ

1 factu

STEP1.txt

not regular file regular contains some regular data file itself serial factorial equal factual

STEP2.txt

not regular file regular contain some regular data file itself seri factori equ factu

STEP3.txt

----------------------------------------

Frequency Word

----------------------------------------

1 not

3 regular

2 file

1 contain

1 some

1 data

1 itself

1 seri

1 factori

1 equ

1 factu