Since the text file you have provided was very large which my laptop could not take, it gets hanged immediately, therefore I have used certain portion of it.

*/\*\*  
 \* Created by asmita on 6/3/2016.  
 \*/***import** java.io.BufferedReader;  
**import** java.io.File;  
**import** java.io.FileReader;  
**import** java.util.\*;  
**import** java.util.Set;  
**import** java.util.List;  
**import** java.util.HashMap;  
**import** java.util.Map;  
  
**public class** Probability {  
 **public static void** main(String[] args) **throws** Exception  
 {  
 FileReader file = **new** FileReader(**"C:/Users/asmita/Desktop/Text1.txt"**);  
  
 BufferedReader reader= **new** BufferedReader(file);  
 String text=**""**;  
 String line=reader.readLine();  
 **while**(line!=**null**){  
 text+=line;  
 line=reader.readLine();  
 }  
 *//System.out.println(text);* String text1=text.toUpperCase();  
 *//System.out.println(text1);* String text2[]=text1.split(**"[(' ),.!;-]"**);  
 **int** totalWords=text2.**length**;  
 System.***out***.println(totalWords);  
 HashMap<String, Integer> hasmap= **new** HashMap<String, Integer>();  
  
  
 **for** (String word:text2)  
 {  
 **if**(word.length()<=2){  
 **continue**;  
 }  
 Integer existingCount=hasmap.get(word);  
 hasmap.put(word,existingCount==**null**?1:(existingCount+1));  
  
 }  
  
  
 Map<Integer, String> map = *sortByValues*(hasmap);  
 **int** m=0;  
 Set set2 = map.entrySet();  
 Iterator iterator2 = set2.iterator();  
 System.***out***.println(**"word\t"**+ **"frequency\t"**+**"Rank"**);  
 **while**(m<20){  
 Map.Entry me2 = (Map.Entry)iterator2.next();  
  
 System.***out***.print(me2.getKey() + **" :"**);  
 System.***out***.print(me2.getValue()+**":"**);  
 System.***out***.println(++m);  
  
 }  
  
 **float** countThe=(hasmap.get(**"THE"**));  
 **float** countHis=(hasmap.get(**"HIS"**));  
 **float** countBrave=(**float**)(hasmap.get(**"BRAVE"**));  
 **float** countHave=hasmap.get(**"HAVE"**);  
 *// Calculating the relative frequency (probability estimate) of the words:  
 //float ppp=(float)ProbabilityForOne((hasmap.get("THE")),totalWords);  
 //System.out.println(ppp);* System.***out***.println(**"The relative frequency of the word THE is:"** + countThe/totalWords);  
 System.***out***.println(**"The relative frequency of the word HIS is:"** + countHis/totalWords);  
 System.***out***.println(**"The relative frequency of the word BRAVE is:"** + countBrave/totalWords);  
 System.***out***.println(**"The relative frequency of the word HAVE is:"** + countHave/totalWords);  
  
 *//Calculating conditional probability* System.***out***.println(**"-----------Conditional probability--------------"**);  
  
 **int** countPair1=*countInPairs*(text2,**"THIS"**, **"MAN"**);*// probability for p(man/this)* **int** countThis=hasmap.get(**"THIS"**);  
 System.***out***.println(**"The probability of occurance of MAN after THIS is:"** + (**float**)countPair1/countThis);  
  
 **int** countPair2=*countInPairs*(text2,**"HATH"**, **"SHE"**); *//probability for p(SHE/HATH)* **int** countHath=hasmap.get(**"HATH"**);  
 System.***out***.println(**"The probability of occurance of SHE after HATH is:"** + (**float**)countPair2/countHath);  
  
 **int** countPair3=*countInPairs*(text2,**"YET"**, **"YOU"**); *//probability for p(YOU/YET)* **int** countYet=hasmap.get(**"YET"**);  
 System.***out***.println(**"The probability of occurance of YOU after YET:"** + (**float**)countPair3/countYet);  
  
 **int** countPair4=*countInPairs*(text2,**"THEE"**, **"FOR"**); *//probability for p(FOR/THEE)* **int** countTHEE=hasmap.get(**"THEE"**);  
 System.***out***.println(**"The probability of occurance of FOR after THEE is:"** + (**float**)countPair4/countTHEE);  
  
   
  
 *// float probab=(\*(float)(count1/countWill)\*(float)(count2/countLook));  
 //System.out.println("The probability of probability(will, look, upon) is:"+ probab);* }  
 **private static** HashMap sortByValues(HashMap map) {  
 List list = **new** LinkedList(map.entrySet());  
 *// Defined Custom Comparator here* Collections.*sort*(list, **new** Comparator() {  
 **public int** compare(Object o1, Object o2) {  
 **return** ((Comparable) ((Map.Entry) (o2)).getValue())  
 .compareTo(((Map.Entry) (o1)).getValue());  
 }  
 });  
  
 *// Here I am copying the sorted list in HashMap  
 // using LinkedHashMap to preserve the insertion order* HashMap sortedHashMap = **new** LinkedHashMap();  
 **for** (Iterator it = list.iterator(); it.hasNext();) {  
 Map.Entry entry = (Map.Entry) it.next();  
 sortedHashMap.put(entry.getKey(), entry.getValue());  
 }  
 **return** sortedHashMap;  
 }  
  
 **private static int** countInPairs(String []str1, String word1, String word2){  
 *//String t;* **int** count=0;  
 **for** (**int** i = 0; i <str1.**length**-1 ; i++) {  
 *//System.out.println(str1[i]);* **if**(str1[i].equals(word1) && str1[i+1].equals(word2)){  
 count++;  
  
 }  
 *//System.out.println("ashmita");* }  
  
 **return** count;  
 }  
  
 **private static float** ProbabilityForOne( **int** countWord1,**int** totalCount){  
 **float** p1=(**float**)(countWord1/totalCount);  
 *//System.out.println(p1);* **return** p1;  
 }}

OUTPUT

