```
 1  /* Problem 1--Splitting The Atom
 2     This was easily handled recursively.  For each A see how many TOMs
 3     be found in the remaining strings.  Since the search string is
 4     always ATOM, this could also be done with nested loops without
 5     it becoming too onerous. */
 6
 7  import java.io.*;
 8  import java.util.*;
 9
10  public class prob1 {
11
12   private static Scanner in;
13   private static PrintWriter out;
14   private static int cs;
15
16   public static void main (String[] args) throws Exception {
17
18    cs = 1;
19    in = new Scanner (new File ("prob1.in"));
20    out = new PrintWriter ("prob1.out");
21    while (true) {
22     String sentence = in.nextLine();
23     if (sentence.equals ("")) break;
24     int ct = Process (sentence,"ATOM");
25     out.printf ("Case %d: There are %d way(s) to split an atom.\r\n\r\n",
26                 cs++,ct);
27    }
28    in.close ();
29    out.close ();
30   }
31
32   /* This is the recursive method that counts subsequences */
33   public static int Process (String s, String w) throws Exception {
34
35    int ct = 0;
36    if (w.equals("")) return 1; //we've found all letters, increment count
37    for (int i=0; i < s.length(); i++)
38     if (s.charAt(i)==w.charAt(0)) //if a string is matched
39      ct += Process (s.substring(i+1),w.substring(1));
40       //Process with rest of both strings
41    return ct;
42   }
43  }
44
```