```
import java.io.*;

import java.math.*;

import java.security.*;

import java.text.*;

import java.util.*;

import java.util.concurrent.*;

import java.util.function.*;

import java.util.regex.*;

import java.util.stream.*;

import static java.util.stream.Collectors.joining;

import static java.util.stream.Collectors.toList;


class Result {


    /*

     * Complete the 'syllables' function below.

     *

     * The function is expected to return an INTEGER.

     * The function accepts STRING word as parameter.

     */


    public static int syllables(String word) {

        StringBuilder temp = new StringBuilder(word);

         ArrayList<Character> vowels = new ArrayList<Character>(Arrays.asList('a','e','i','o','u'));

        ArrayList<String> prefixes = new ArrayList<String>(Arrays.asList("co", "de", "dis","pre", "re","un",
"tri"));

        ArrayList<String> suffixes = new ArrayList<String>(Arrays.asList("age", "ful", "ing","less", "ment",
"port"));

        ArrayList<String> comboCon = new ArrayList<String>(Arrays.asList("ch", "ck", "ph","sh", "th", "wh",
"wr", "ll", "rd"));
```

```
//find prefixes
String prefix = "";
String firstTwo = temp.substring(0,2);
String firstThree = temp.substring(0,3);

if (prefixes.contains(firstTwo)){
   temp.insert(temp.indexOf(firstTwo)+ 2,'|');
   prefix = firstTwo;
}
else if (prefixes.contains(firstThree)){
   temp.insert(temp.indexOf(firstTwo)+ 3,'|');
   prefix = firstThree;
}

//find suffixes
String suffix = "";
String lastThree = temp.substring(temp.length()-3);
String lastFour = temp.substring(temp.length()-4);

if (suffixes.contains(lastThree)){
   temp.insert(temp.indexOf(lastThree),'|');
   suffix = lastThree;
}
else if (suffixes.contains(lastFour)){
   temp.insert(temp.indexOf(lastFour),'|');
   suffix = lastFour;
}
```

```
        //take out the prefix/suffixes for the rest of the syllable splitting
        if (!prefix.equals(""))
            temp = new StringBuilder(temp.substring(prefix.length()+1));
        if (!suffix.equals(""))
            temp = new StringBuilder(temp.subSequence(0, temp.length()-suffix.length()-1));


        //split single consonants
        //first look for combos surrounded by vowels
        for (int i =1; i<temp.length()-2; i++){
            String combo = temp.substring(i, i+2);
            //if combo is surrounded by vowels
            if (comboCon.contains(combo) && vowels.contains(temp.charAt(i-1)) &&
vowels.contains(temp.charAt(i+2))){
                temp.insert(temp.indexOf(combo),'|');
            }
        }


        //then look for non combos surrounded by vowels
        for (int i = 1; i<= temp.length()-2; i++){
            char letter = temp.charAt(i);
            if (!vowels.contains(letter) && letter!= '|' && vowels.contains(temp.charAt(i-1)) &&
vowels.contains(temp.charAt(i+1)))
                temp.insert(i, '|');


        }


        //then look for double consonants
        for (int i = 0; i<=temp.length()-2; i++){
            char letter1 = temp.charAt(i);
            char letter2 = temp.charAt(i+1);

            String combo = Character.toString(letter1)+Character.toString(letter2);

            //neither is '|', not a consonant combo

            if (letter1!= '|' && letter2 !='|' && !comboCon.contains(combo)){

                //s isn't added for plurality + neither is a vowel

                //how are we supposed to take into account that 'bl' is a combo consonant at the beginning???

                if (!(i== temp.length()-2 && letter2 == 's') && !vowels.contains(letter1) &&
!vowels.contains(letter2) && !(i == 0 && combo.equals("bl")))

                    temp.insert(i+1,'|');

            }

        }


        //add back on the prefix/suffixes

        StringBuilder answr = new StringBuilder(temp);

        if (!prefix.equals(""))

            answr.insert(0, prefix+"|");

        if (!suffix.equals(""))

            answr.append("|"+suffix);


        //answr = new StringBuilder("ta|bles|poon|ful");

        int sum = 0;

        for (int i =0; i<answr.length(); i++){

            if (answr.charAt(i) == '|')

                sum+=i;

        }


        return sum;


    }
```

```
}


public class Solution {
    public static void main(String[] args) throws IOException {
        BufferedReader bufferedReader = new BufferedReader(new InputStreamReader(System.in));
        BufferedWriter bufferedWriter = new BufferedWriter(new
FileWriter(System.getenv("OUTPUT_PATH")));


        String word = bufferedReader.readLine();


        int result = Result.syllables(word);


        bufferedWriter.write(String.valueOf(result));
        bufferedWriter.newLine();


        bufferedReader.close();
        bufferedWriter.close();
    }
}
```