```
#include <bits/stdc++.h>

using namespace std;

/*
 * Complete the 'syllables' function below.
 *
 * The function is expected to return an INTEGER.
 * The function accepts STRING word as parameter.
 */

vector<string> prefix {"co", "de", "dis", "pre", "re", "un"};
vector<string> suffix {"age", "ful", "ing", "less", "ment"};
vector<string> combo {"ch", "ck", "ph", "sh", "th", "wh", "wr"};
vector<char> vowel {'a', 'e', 'i', 'o', 'u'};

pair<bool, bool> checkOne(string word, int i) {
   // check if i -> consonant...
   bool g, ic, l, r;
   g = 0;
   for(char j: vowel) {
      if(word[i] == j) {
         g = 1;
         break;
      }
   }

   if(g) {
      return make_pair(false, false);
```

```
    }

    // check if it is a combo...
    ic = 0;
    for(string j: combo) {
        if(word.substr(i, 2) == j) {
            ic = 1;
            break;
        }
    }

    // check if left and right are vowels...
    l = 0, r = 0;
    for(char j: vowel) {
        l |= word[i-1] == j;
        r |= word[i + (ic ? 2 : 1)] == j;
    }

    if(!l || !r) {
        return make_pair(false, ic);
    }

    return make_pair(true, ic);
}

bool checkTwo(string word, int i) {
    bool g, ic;

    // check if i -> consonant...
```

```
g = 0;
for(char j: vowel) {
   if(word[i] == j) {
      g = 1;
      break;
   }
}

if(g) {
   return false;
}

// check if it is a combo...
ic = 0;
for(string j: combo) {
   if(word.substr(i, 2) == j) {
      ic = 1;
      break;
   }
}

if(ic) {
   return false;
}

// check vowel before and after...
g = 0;
for(int j = i-1; j >= 0; --j) {
   for(char k: vowel) {
```

```
        if(word[j] == k) {
            g = 1;
            break;
        }
    }
    if(g) {
        break;
    }
}

if(!g) {
    return false;
}

g = 0;
for(int j = i+1; j < word.length(); ++j) {
    for(char k: vowel) {
        if(word[j] == k) {
            g = 1;
            break;
        }
    }
    if(g) {
        break;
    }
}

if(!g) {
    return false;
```

```
    }

    return true;
}

int syllables(string word) {
    int s = 0, c = 0;
    for(int i = 0; i < prefix.size(); ++i) {
        if(word.substr(0, prefix[i].length()) == prefix[i]) {
            word = word.substr(prefix[i].length());
            s = prefix[i].length();
            c = prefix[i].length()+1;
            break;
        }
    }

    for(int i = 0; i < suffix.size(); ++i) {
        if(word.substr(word.length()-suffix[i].length()) == suffix[i]) {
            s += c + word.length() - suffix[i].length();
            word = word.substr(0, word.length()-suffix[i].length());
            ++c;
            break;
        }
    }

    for(int i = 1; i < word.length()-1; ++i) {
        pair<bool, bool> r1 = checkOne(word, i);

        if(r1.first) {
```

```
            word = word.substr(i);
            s += c + i;
            c += i + 1;

            i = 0;
            if(r1.second) {
               ++i;
            }
         }else{
            bool r2 = checkTwo(word, i);

            if(r2) {
               word = word.substr(i+1);
               s += c + i + 1;
               c += i + 2;

               i = 0;
            }
         }
      }

      return s;
}

int main()
{
   ofstream fout(getenv("OUTPUT_PATH"));

   string word;
```

```
    getline(cin, word);

    int result = syllables(word);

    fout << result << "\n";

    fout.close();

    return 0;
}
```