```
#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.
 */

string s,s1,s2;
bool checons(char c1,char c2)
{
   if(c1=='c'&&c2=='h')
      return 1;
   if(c1=='c'&&c2=='k')
      return 1;
   if(c1=='p'&&c2=='h')
      return 1;
   if(c1=='s'&&c2=='h')
      return 1;
   if(c1=='t'&&c2=='h')
      return 1;


   if(c1=='w'&&c2=='h')
      return 1;
   if(c1=='w'&&c2=='r')
```

```
        return 1;
    return 0;
}
vector<string>v;
int chesuff()
{
    if(v[0].size()!=1||v[1].size()!=1)
        return 0;
    char c1=v[0][0],c2=v[1][0],c3=v[2][0];
    if(c1=='c'&&c2=='o')
        return 2;
    if(c1=='d'&&c2=='e')
        return 2;
    if(c1=='d'&&c2=='i'&&c3=='s')
        return 3;

    if(c1=='p'&&c2=='r'&&c3=='e')
        return 3;
    if(c1=='r'&&c2=='e')

        return 2;
    if(c1=='u'&&c2=='n')
        return 2;
    return 0;
}
int checaff()
{
    if(v[v.size()-4].size()!=1||v[v.size()-3].size()!=1||v[v.size()-2].size()!=1) return 0;
    char c1,c2,c3,c4;
```

```
    c4=v[v.size()-1][0];
    c3=v[v.size()-2][0];
    c2=v[v.size()-3][0];
    c1=v[v.size()-4][0];



    if(c2=='a'&&c3=='g'&&c4=='e')
        return 3;
    if(c2=='i'&&c3=='n'&&c4=='g')
        return 3;



    if(c2=='f'&&c3=='u'&&c4=='l')
        return 3;
    if(c1=='m'&&c2=='e'&&c3=='n'&&c4=='t')
        return 4;
    if(c1=='l'&&c2=='e'&&c3=='s'&&c4=='s')
        return 4;
    return 0;
}
bool vv(string s)
{
    if(s.size()!=1)
        return 0;
    char ch=s[0];
```

```
    if(ch=='a' || ch=='e'||ch=='i'||ch=='o'||ch=='u')
        return 1;
    return 0;
}

int syllables(string word)
{
    string s=word;
    int i;
    for(i=0; i<s.size(); i++)
    {
        s1=s2;
        if(checons(s[i],s[i+1])) s1=s1+s[i]+s[i+1],i++;
        else s1=s1+s[i];
        v.push_back(s1);
    }
    /*
    fdgsgds

    sdgds
    */

    int st=0,dr=v.size()-1,sum=0,poz ,baricnt=0;
    poz=chesuff();

    if(poz)//dsfgdsdda
    {
        for(register int i=0; i<poz; i++)
```

```
        s2=s2+v[i];
    s2=s2+'|';
    st=poz;
}
/*
ll["LOAD"]=1;
ll["STORE"]=1;
ll["ADD"]=1;
ll["SUB"]=1;
ll["MULT"]=1;
ll["DIV"]=1;
ll["BE"]=1;
ll["BG"]=1;
ll["BL"]=1;
ll["BU"]=1;
ll["END"]=1;
ll["READ"]=1;
ll["PRINT"]=1;
ll["DC"]=1;
*/

poz=checaff();
if(poz)
    dr=dr-poz;
int voc=0;
/*

for(i=0;i<s.size();i++)
        if(isalpha(s[i]))
```

```
        s[i]=toupper(s[i]);
        if(s=="END")
           s2=s;<<2;
  if(s2.size())
    fout<<s2;
    if(s3.size())
  fout<<6-s2.size();
fout<<s3<<"\n";
*/

for(register int i=st; i<=dr; i++)
{
    if(vv(v[i])) voc=1, s2=s2+v[i];
    else
    {
        /*
        for(i=0;i<s.size();i++)
        if(isalpha(s[i]))
        s[i]=toupper(s[i]);
        if(s=="END")
           s2=s;
  if(s1.size()==0)
    fout<<2;
  else
    fout<<s1<<2;
  if(s2.size())
    fout<<s2;
    if(s3.size())
  fout<<6-s2.size();
```

```
fout<<s3<<"\n";
      */
      int tru1,tru2;

      if(i+1<=dr)
         tru1=vv(v[i+1]);
      else tru1=-1;
      if(i+2<=dr)
      tru2 = vv(v[i+2]);
      else tru2=-1;
      if((voc==1 && tru1==0) &&tru2==1)
      {
         voc=0;

         s2=s2+v[i]+'|';
      }
      else if(voc==1&&tru1==1)
      {
         voc=0;
         s2=s2+'|'+v[i];
      }
      else
         s2=s2+v[i];
   }
}
poz=checaff();
if(poz)
   s2 = s2+'|';
for(register int i=0; i<s2.size(); i++)
```

```
    {

        if(s2[i]=='|') sum=sum+i;
        /*
        ll["LOAD"]=1;
    ll["STORE"]=1;
    ll["ADD"]=1;
    ll["SUB"]=1;
    ll["MULT"]=1;
    ll["DIV"]=1;
    ll["BE"]=1;
    ll["BG"]=1;
    ll["BL"]=1;
    ll["BU"]=1;
    ll["END"]=1;
    ll["READ"]=1;
    ll["PRINT"]=1;
    ll["DC"]=1;
        */
    }

    return sum;
}



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

```
    string word;
    getline(cin, word);

    int result = syllables(word);

    fout << result << "\n";

    fout.close();

    return 0;
}
```