Admin Blues.

- Prepared by: Sagun Shakya (https://github.com/sagsshakya)
- MSc. Data Science
- GITAM Institute of Science, Visakhapatnam.
- Email: sags.shakya@gmail.com (mailto:sags.shakya@gmail.com)

A person, new to his job in a start-up, was given a task by his supervisor to prepare a report where the user IDs (of all the employees) is a part of the report. He fetched the user IDs from database and then used a spreadsheet function to concatenate them to share it in email as text. However, while concatenating, he forgot to put a comma , in between the two user IDs As a rule, user IDs of the portal should be minimum 4 characters and can only contain alphabets and cannot start or end with a vowel.

Supervisor knows about this rule and the number of employees in the department. Therefore, he tries to guess a possible set of user IDs from the given string (of user IDs in the email). You need to find the total no. of different set of user Ids that are possible given the number of employees in the department and the string of user IDs.

- Constraints
  - 1<Number of Employees<10
  - 7<Length of user id string <51
- Input Format
  - First Line contains an integer, which provides the Number of Employees in the organization
- Second Line contains a string of concatenated user ids
- Output
  - Number of possibilities of user ID sets
  - If none are possible, print 0
- Example Input 1
  - 2
  - nonajklop
- Output
  - 1
- Explanation
  - Since supervisor knows there are only 2 user IDs and they cannot start or end with a vowel, there is only one possibility - nonaj and klop so there is only one probability of the user ID sets.

C++ Code:

#include

using namespace std;

```
#define SUMAN ios_base::sync_with_stdio(false);cin.tie(NULL);cout.tie(NULL);
#define SUMANBUIE___ SUMAN
bool isVouel(char c){
if(c=='a'||c=='A'||c=='E'||c=='e'||c=='I'||c=='i'||c=='O'||c=='o'||c=='U'||c=='u')return true;
return false;
}
bool isItFallow(string s){
if(isVouel(s[0])||isVouel(s[s.length()-1])||s.length()<4)return false;
return true;
}
void parmutations(string s,int n,int &ans,unordered_set&st){
if(isVouel(s[0]||isVouel(s[s.length()-1])))return;
if(n==1){
if(isItFallow(s)){
auto it = st.find(s);
if(it==st.end()){
ans+=1;
}
}
return;
}
int i = 3;
while(i<(s.length()-4)){
if(!isVouel(s[i])&&!isVouel(s[i+1])&&isItFallow(s.substr(0,i+1))){
if(st.empty()){
st.insert(s.substr(0,i+1));
parmutations(s.substr(i+1),n-1,ans,st);
st.erase(s.substr(0,i+1));
}
```

```
else{
auto it = st.find(s.substr(0,i+1));
if(it==st.end()){
st.insert(s.substr(0,i+1));
parmutations(s.substr(i+1),n-1,ans,st);
st.erase(s.substr(0,i+1));
}
}
}
i+=1;
}
}
void solve(){
int n;cin>>n;
string s;cin>>s;
int ans = 0;
unordered_setst;
parmutations(s,n,ans,st);
cout<<ans;
}
int main(){
SUMANBUIE___
int testcase;
//cin>>testcase;
testcase = 1;
while(testcase--){
solve();
}
return 0;
```

```
}
```

The End.