

Program Name: barney.cpp

Input File: barney.dat

Barney is a dinosaur from our imagination. But he is also a television show in which a big purple dinosaur sings children's songs. Many of Barney's songs are really just old songs with new lyrics. The purpose of this program is to use a set of new lyrics and find an old song with a tune that can be used for the new lyrics. Your program will do so by scanning a set of old songs and counting the number of syllables per line. Your program will try to find the old song whose syllables per line most closely match the corresponding syllables per line from a new lyric. To count the syllables on a line, your program should implement the following rules.

1. All words will have at least 1 vowel.
2. If a word has a set of one or more vowels {a, e, i, o, u, y} separated by one or more consonants (set of all letters minus the set of vowels) from another set of vowels, then each set of vowels in the word would constitute a syllable. The following table shows some examples that implement these rules.

| Sample Word | # of Syllables | Explanation |
|-------------|----------------|---|
| chat | 1 | There is only 1 vowel set {a} and there are no other vowel sets that are separated from it by 1 or more consonants. |
| apple | 2 | There are 2 vowel sets, {a} and {e}, separated by a set of consonants {p, p, l}. |
| alphabetic | 4 | There are 4 vowel sets separated from one another |
| one | 2 | There are 2 vowel sets, {o} and {e}, separated by a consonant {n}. Now you can see the weakness in this method, but it is OK because we are only going to identify candidate songs. |
| bureau | 2 | There are 2 vowel sets, {u} and {eau}, separated by a consonant {r}. |
| bayou | 1 | Here is the weakness of including y as a vowel. |
| city | 2 | And here is an argument for including y as a vowel. |

Each song that you examine (old song or new) will contain exactly 4 lines. For each of the four lines, you should compute the sum of the differences squared (SoDS) in the number of syllables for corresponding lines.

| Old Songs | | | | | | New Song | |
|--------------------------------|---|------|--------------------------------|---|------|----------------|---|
| "This Old Man" | # | SoDS | "Twinkle Twinkle Little Star" | # | SoDS | "I Love You" | # |
| this old man | 3 | 1 | twinkle twinkle little star | 7 | 9 | i love you | 4 |
| he played one | 4 | 0 | how i wonder what you are | 8 | 16 | you love me | 4 |
| he played knick knock | 4 | 4 | up above the world so high | 8 | 4 | we are a happy | 6 |
| on my thumb | 3 | 0 | like a diamond in the sky | 8 | 25 | family | 3 |
| Sum of the Differences Squared | | 5 | Sum of the Differences Squared | | 54 | | |

So the total of the sum of differences squared for "This Old Man" is 5 versus 54 for "Twinkle Twinkle Little Star". Therefore, we identify "This Old Man" as a better candidate for the new lyrics than "Twinkle Twinkle Little Star". In this program, you will be given a library of up to 20 old songs followed by a series of new songs. For each of the new songs, your program will identify the best candidate old song using the sum of the differences squared method explained above.

Input

Input to your program consists of two sections. The first section describes the library of old songs. Line 1 of the input file will contain a single integer ($1 \leq L \leq 20$) that describes the number of old songs in the library. Each 4 of the next $L * 4$ lines of the input file will contain the lyrics of one old song. Each old song will be made up of 4 lines of 1 to 80 lower case alphabetic characters only. Words in the lyrics will be separated by a single space.

A series of "new songs" will start at line # $4L + 2$ each formatted with the same rules as those in the library. Your program should read each of the "new songs" and determine which "old song" is the best candidate based on the

sum of differences squared methodology described above. If there is a tie for the best candidate, your program should select the song involved in the tie that appears first in the library.

Output

For each “new song” in the input file, your program should print exactly two lines of output. The first line of output should be the lyrics of the first line of the “old song” that is the best candidate. The best candidate line should be followed by exactly 1 blank line.

Example: Input File

```
3
twinkle twinkle little star
how i wonder what you are
up above the world so high
like a diamond in the sky
the itsy bitsy spider
went up the water spout
down came the rain and
washed the spider out
this old man
he played one
he played knick knack
on my thumb
i love you
you love me
we are a happy
family
a bee cee dee e ef gee
aich i jay kay ell em en oh pea
cue arr ess tea you vee
double u ex why and zee
```

Output to screen

```
this old man

twinkle twinkle little star
```