**8.8 Assignment 8, due Thursday 2018-04-19, at 11:00pm**

For this assignment, you are to implement a solver for a simple solitaire game using cards marked with lowercase letters.

8.8.1 The game

Given a string, imagine that it is a line of cards, arranged from left to right. The rules of the game are:

1. At each step, we may pick two cards that are either adjacent, or have two other cards in between them.
2. Two cards match if they are both vowels (letters in "aeoiu") or if their ASCII values differ by at most 5.
3. If we pick two cards that match, we can move the right matching card onto the left matching card, effectively replacing it. The gap left by the right card is closed up by moving subsequent cards one space to the left.
4. We win if we can reduce the original word to one card.

8.8.2 Examples

Suppose the initial word is shrink.

Because i and n are 5 apart in their ASCII encodings and adjacent, we can move the n on top of the i to get shrnk.

Because h and k are 3 apart in their ASCII encodings and have two cards between them, we can move the k on top of the h to get skrn.

Because r and n are 4 apart in their ASCII encodings and adjacent, we can move the n on top of the r to get skn.

Because k and n are 3 apart in their ASCII encodings and adjacent, we can move the n on top of the k to get sn.

Because s and n are 5 apart in their ASCII encodings and adjacent, we can move the n on top of the s to get n.

Since we ended up with a single letter, we won. (Note that there might be other ways to win in this case.)

We can summarize the play of the game by printing each word, starting with shrink and ending with n, like this:

shrink

shrnk

skrn

skn

sn

n

Because shrink has only one vowel, we couldn't use the vowel rule. An example of a sequence that uses the vowel rule is this reduction of road:

road

oad

ad

d

In the second step, we can move a onto o even though their ASCII values differ by 14, because they are both vowels.

If we start with ax, we can't win, because a and x differ by too much, and only one of them is a vowel.

8.8.3 Your task

Write a program shrink that takes a word as argv[1] and either prints out a reduction to a single letter following the rules defined above, or prints nothing if there is no such reduction. In either case your program should exit with a zero error code.

Some typical executions of shrink might look like this:

$ ./shrink yaleuniversity

yaleuniversity

yaluniversity

yaluiversity

yaluiversiy

yiluversiy

yluversiy

yluvirsy

yluvisy

yiuvsy

yuvsy

uvsy

yvs

vs

s

$ ./shrink elephant

$ ./shrink a

a

8.8.4 Efficiency

You should try to pick an approach to solving the problem that is reasonably efficient. If you solve the program using some sort of search, it may be helpful to include a mechanism that prevents searching past the same intermediate result more than once.

8.8.5 Bad inputs

It might be polite to check for bad inputs (for example, calling ./shrink with no arguments or with a word that contains a character that is not a lowercase ASCII letter), but you are not required to do so, and the test script will supply only good inputs.