CSCI 301, Lab # 5
'Winter, 2017

Due: Your program, named `lab05.rkt`, must be submitted to Canvas before midnight, Tuesday, February 21.

Pig Latin (from Wikipedia): Pig Latin is a language game in which words in English are altered. The objective is to conceal the words from others not familiar with the rules. For words that begin with consonants, all letters before the initial vowel are placed at the end of the word sequence. Then, "ay" is added, as in the following examples:

- "pig" = "igpay"
- "latin" = "atinlay"
- "banana" = "ananabay"
- "trash" = "ashtray"
- "duck" = "uckday"
- "glove" = "oveglay"
- "too" = "ootay"
- "thanks" = "anksthay"
- "cheers" = "eerschay"

For words that begin with vowels, one just adds "way" to the end. Examples are:

- "eat" = "eatway"
- "omelet" = "omeletway"
- "are" = "areway"
- "egg" = "eggway"

For purposes of this assignment, we will consider only letters in the set $\{a, e, i, o, u\}$ to be vowels. We will also assume that every word has at least one vowel in it.

Program:

Write a program to translate a string of text into pig latin. To simplify this first attempt, assume all punctuation has been removed and all letters are lowercase, for example:

```
> (piglatin  "hello there you gorgeous thing are you busy tonight")
"ellohay erethay ouyay orgeousgay ingthay areway ouway usybay onighttay"
```

A few utility functions from the standard racket library (no extra modules need be loaded) that you can use:
`string->list` `list->string` `string-join` `string-split`
Look up their documentation in the Racket help system.

Using these will allow you to use list processing instead of string processing (which is much messier).

Do not do string processing!

First turn the string into a list of words, then turn each word into a list of characters, convert these lists into piglatin lists of characters, then turn the characters back to a word string, and finally join the word strings into a single string.

Feel free to use list processing procedures, such as:
`cons` `car` `cdr` `append` `map` `member`
Look up their documentation in the Racket help system.