ENGR 102 - Lab 7b

Program 1: (Pig Latin)

This program is meant to help give you practice with string manipulation.

"Pig Latin" is a way of converting words in standard English to similar words that sound different. The rules for converting from standard words to Pig Latin are as follows:

- If a word starts with a consonant, move the consonants to the end of the word, and add "ay" to the end.
  - e.g. "computer" becomes "omputercay"
- If a word starts with a vowel, add "yay" on to the end of the word.
  - e.g. "engineering" becomes "engineeringyay"
- Note: treat "y" as a vowel for this assignment.

Write a program that repeatedly reads in a word from a user and converts it to Pig Latin. The program should continue reading words until the user enters "stop".

Challenge: Try to write a program where, instead of just one word, the user enters an entire sentence, and all words in the sentence are converted to Pig Latin.

Program 2: (Vector Math)

This program is meant to give you practice with lists and looping on them, as well as practice with vector computations.

Write a program that lets a user enter two vectors, A and B, of arbitrary dimension. You should allow the user to first enter the dimension of the vector, then get the elements of the two vectors from the user. Then, you should output (in a clearly labeled way), the results of these computations:

- The magnitude of vector A and the magnitude of vector B
- A + B
- A – B
- And the dot product (inner product) of A and B

Note: You should use lists when solving this problem. Later we will see some other ways to work with vectors more directly.

Program 3: (Password Protection)

This program is mean tot give you practice using dictionaries.

Write a program that first reads from a user a set of usernames and passwords, and then enters a program that simulates a user typing in passwords.

In the first part of the program, you are to read in a single integer that states the number of username/password pairs that you will read in. Following this, there will be a set of that many user names, one per line. Then there will be a set of that many passwords, one per line. You only need to prompt one time at the beginning of the program.

For example, input might be:

```
3
John
Jim
Joe
SecreT_passWorD
12345
G$a-4(ztY
```

Once that is read in, you should then repeatedly ask the person to type in a username and then password. If they have a valid username and password combination, then print a message that they are allowed into the system. If they have an incorrect username/password, then tell them (and allow them) to try again (repeatedly).

Note: in practice, although this is the basic way passwords are handled, they should not be stored in unencrypted format even in the program. Instead, passwords are encrypted when they are typed in, they are stored in an encrypted format, and the comparisons are always between encrypted passwords. There are also more extreme security measures.