Assignment Objectives:  
- Test understanding of stacks and queues.  
- Implement an array-based stack class, with each node containing strings.  
- Implement a linked list-based queue class, with each node containing a stack.

In this assignment, you will develop an array implementation of a stack, and a linked list implementation of a queue, the elements of which will, in fact, be the aforementioned array-based stacks. In addition, you will need a driver program to test the functionality of your classes.

First, you'll define and implement a new array-based Stack class. Each element in the Stack will be a structure composed of a question (a string), four possible answers (a static array of strings), and a single character (valued 'a' through 'd') indicating which of the four answers is correct. The Stack will also have a string data member specifying the category of the questions in the Stack, as well as messages to be output if this category of questions is successfully or unsuccessfully completed.

In addition, you will define and implement a new QueueOfStacks class, based upon a circular linked list. Every element in a QueueOfStacks variable will be an instance of the new Stack class.

Finally, you will need to write a driver program, which will test the functionality of your classes by reading several input files loaded with questions and then conducting an interactive exam with the user.

Three input files (Language1.txt through Language3.txt) have been set up to help you test your program. Each file (available on the course website) has been formatted as follows:

- First, a one-word category describing the type of questions in the file.
- Next, a one-line output message to be used if the user successfully completes the portion of the exam using this category, followed by a one-line message to be used if the user is unsuccessful in this category.
- Finally, ten question-and-answer sections, each consisting of a one-line question, four one-line answers, and a single character representing which of the answers is correct.

While you may assume that each category, output message, question, answer, and correct-answer character are on separate lines, you may not assume how many blank lines separate these features.

The QueueOfStacks class should be implemented as a circular linked list version of a queue of Stack items with only a tail pointer. Whenever the user is about to be asked a question, the front Stack in the QueueOfStacks is dequeued, and its top element is popped and output. The questions are rather lengthy, so some should be split at an appropriate blank space and output on two lines. Each answer should be output on a separate line, preceded by some indentation and a letter.

The user is then asked to select one of the answers. If the user responds with the correct answer, that Stack is not enqueued again. If the user responds with an incorrect answer and there are additional questions in the Stack, then the Stack should be enqueued. If the user responds incorrectly to the last question in a Stack, then the exam is considered unsuccessful. To successfully complete the exam, the user must answer one question from each Stack correctly before any Stack is completely depleted (i.e., if the QueueOfStacks is empty after the last correct response, then the user has succeeded).

A correct response to a question should be met with one of ten randomly generated congratulatory messages:

- "United Nations, here you come!"
- "You're certainly open to interpretation!"
- "You're ready for a world tour!"
- "You multilingual devil!"
- "Pretty fancy tonguework!"
- "You display a real knack for languages!"
- "You've conquered that language!"
- "You put the "slay" in "translation"!"
- "That language sure wasn't foreign to you!"
- "Good answer! Say goodbye to that section of POLYGLOT!"
An incorrect response to a question should be met with one of ten randomly generated insults:

- "Sorry, but that was a MIS-communication..."
- "Maybe that one was a little TOO foreign..."
- "You might want to postpone that trip abroad..."
- "Not much of a world traveler, are you...?"
- "Ouch! That wasn't very inter-pretty..."
- "Perhaps the problem is your English..."
- "Unfortunately, no "Bingo" on that lingo..."
- "Too bad it's not the MONOGLOT exam..."
- "As an interpreter, you'd probably cause an international incident..."
- "You're not supposed to use foul language..."

A response of 'Q' (or 'q') to any question indicates that the user is quitting the exam, and should be met with the message:

- "Au revoir, mon ami"

Appropriate messages should also be output for missing every question in one category of the exam:

- "<loss message for category> You have missed every <losing category> question! Better luck next time!"

for correctly responding to a question from every category of the exam:

- "<win message> You have passed the exam!"

and for entering meaningless input

- "INAPPROPRIATE RESPONSE... (Your response must be between A and D)"

A sample interactive session is illustrated at right. Note that the questions are initially randomized within each stack, ensuring that taking the exam will not be too repetitive. Your output and interactive sessions are expected to reflect this example. An executable version of this program is also available on the course website. Sample exam question text files are also available at that location.

Zip-compress your entire program folder and place it on your dropbox by the deadline mentioned above. Several files of test data are available on the course website.

---

You must write your own code on this assignment, with adequate explanatory documentation. Obtaining code assistance from any outside source is considered academic misconduct. The only person permitted to see your code prior to the assignment deadline is the instructor.