Assignment 6: Reading and Programming Project due 10/26

Reading Assignment

This week’s reading assignment will be on arrays - Section 6.1- we will start the topic on Tuesday, so until then the reading assignment is just to review the loops.

Programming Project

**HangmanPart2:** Complete the Hangman game [due 11:00 p.m. on 10/26]
worth 12 points, 3 points extra credit

This week’s programming assignment is to complete the Hangman game. You may base your solution on your solution to part 1, which was assigned last week, or use my solution to that part (posted on our course web page).

Recall, that in the previous part we have developed two methods hangmanTemplate(), and countLetter().

The extension required in this assignment is to allow the user to guess the word by entering characters. The program will check and report whether a guessed character occurs in the word, until the user either makes 6 incorrect guesses, or wins by correctly guessing all letters of the word.

Here are a couple of sample interactions describing the required functionality. In the first one the word that the program has picked is `temperature`. The mechanism by which your program selects a word is presented in a later section of this project description.

Let’s play a game of Hangman.
I have picked a word that starts with letter T and ends with E.
Here’s the template in which each dash denotes a single letter:
T - - - - - - - - - E
Now it’s your turn to guess a letter.
Please enter your next guess: F
Incorrect. Letter F does not occur in this word.
Please enter your next guess: E
Correct. Letter E occurs in this word at position 1
Please enter your next guess: Z
Incorrect. Letter Z does not occur in this word.
Please enter your next guess: P
Correct. Letter P occurs in this word at position 3
Please enter your next guess: Q
Incorrect. Letter Q does not occur in this word.
Please enter your next guess: O
Incorrect. Letter O does not occur in this word.
Please enter your next guess: L
Incorrect. Letter L does not occur in this word.
Please enter your next guess: I
Incorrect. Letter I does not occur in this word.
You lost this game. The word is TEMPERATURE

In the interaction above the program declared victory after the user has made 6 incorrect guesses.

Here’s another interaction in which the user wins.

Let’s play a game of Hangman.
I have picked a word that starts with letter L and ends with P.
Here’s the template in which each dash denotes a single letter:
L - - P
Now it’s your turn to guess a letter.
Please enter your next guess: A
Incorrect. Letter A does not occur in this word.
Please enter your next guess: O
Correct. Letter O occurs in this word at
  position 1
  position 2
You won!

A sketch of an algorithm:

While the user has not won and has made less than 6 incorrect guesses:
{
  Get a character from user
  Check if the character occurs in this word and report (call countLetter() method and store the return value)

  if guessed right (i.e. countLetter() returned a value greater than 0)
   compute how many positions remain unfilled using the result
   returned by the countLetter()

   if that number is 0 then the user won!

  else, i.e. bad guess
   increment the counter of bad guesses
}
Declare the winner.

For extra credit (3 points) make your program identify and not count repeated guesses, for example if the user suggests letter Q more than once, the program should not count the second and any later attempt and should print:

This letter was used earlier.