Prelab 1 (for Friday, September 2nd)

Alice and Bob decide to play a game that involves flipping a coin until a player's 3-toss sequence occurs. For example, if Alice's sequence is Heads-Heads-Tails (HHT), and Bob's is THH, then Alice would win if HHT occurs before THH. Their friend, Eve, says that she believes Bob is "a lucky guy" because he found a four-leaf clover near Hinkson creek and thus will tend to win much more often than Alice, who broke a mirror while moving into her dorm. Alice says Eve is silly for believing such superstitions and insists that each person will win approximately half the time. For this prelab you are to implement a simulation of this game and use it to assess whether Eve or Alice is correct.

Here is the prototype for the function you will use to perform your simulations:

```
/* This function simulates the coin-toss game for two players.
   The first two arguments are character strings of length 3,
   where each character is H or T. If one of the strings does not
   satisfy the requirements, the function returns a nonzero number
   to signify an error, otherwise it returns 0 to signify there
   were no errors. A sequence of coin tosses is then simulated
   until the sequence of one of the players is encountered.
   The number of the winning player will be stored at the
   location specified by winner and the number of tosses
   performed will be stored in numTosses. */

int coinTossGame(char player1[3], char player2[3],
                 int *winner, int *numTosses)
```

For your implementation, you're free to return different error codes to signify to the user (you, in this case) what error occurred. For example, error return code 1 might signify there was something wrong with player 1's sequence, and 2 might signify there's a problem with player 2's sequence. You could even provide more detailed error codes, e.g., that indicate which letter in a player's sequence was problematic. It's up to you to be as informative as you feel is appropriate.

EXAMPLE: If the above function produces the following sequence of random tosses, then it will terminate when it encounters the sequence THH, which signifies that Bob has won:

THTTHTHTHH *(Bob wins)*

Once you've implemented and tested this function, you can call it using the sequences selected by Alice and Bob. More specifically, you can call it a thousand times (or why not a million?) and determine their respective winning percentages. If they each win around 50% of the time, then that will tend to support Alice and refute Eve's prediction. Otherwise, you may end up concluding that Bob is in fact "a lucky guy".

The only aspect of your function that might prove a little tricky to implement is how to randomize a sequence of coin tosses. One way to simulate a single coin toss is:

```
toss = rand() % 2;
```

This will return 1 if the random integer is odd and 0 if it's even. You can interpret 1 as heads and 0 as tails, or you can assume the reverse – *it shouldn't matter*.

- JKU