# Hangman

Sometimes, the apps we build are just for fun and to suck up some time. A classic game to play on a rainy day is hangman. You will create a console application that will allow one user to enter in a puzzle and another user (or users) to solve it.

# Competencies Covered:

While this lab may be used to help pass off other competencies, these are the ones upon which it focuses:

* If-statements
* Else statements
* Iteration (partial coverage)
* Basic array usage
* Simple Variables
* Sequential Operations

# Requirements

1. At the start of your application the user will be prompted for the puzzle to be presented to the player. Limit the puzzle to at least 1 word and no more than 8. Loop the prompt until the puzzle is valid.
2. Once the puzzle is entered display the game. The game display consists of the gallows (for the “Man” we’re trying not to hang), the puzzle itself, and the current collection of letters guessed. The puzzle is displayed with underscores for every letter that needs to be guessed. Spaces, apostrophes, quotation marks and other punctuation should be displayed as normal. Only letters should be concealed.
3. The game is played in rounds. Each round the player may guess a letter, solve the puzzle, or give up. These options should be presented in menu form each time the game display is printed.
4. When the player chooses to guess a letter, the player will be prompted to enter a single letter. If the letter appears in the puzzle, every instance of that letter in the puzzle is unveiled. Guesses are to be considered case insensitive. In other words guessing a capital “A” in the puzzle “An apple a day” would lead to the output   
   “A\_ a \_ \_ \_ \_ a \_ a \_” (extra spacing added for readability).
5. If the player guesses a letter that is not in the puzzle, the Man gains a body part. The Man’s body is built in the following order: head, torso, arm, arm, leg, leg. This allows for 6 incorrect answers before the Man is hanged and the game is lost.
6. Every time the player makes a guess, that letter is added to the “already guessed” collection. Making a duplicate guess does not count against the player. For instance, if “Q” is not in the puzzle, guessing “Q” once would result in the hangman gaining a body part. However, if the player then guesses “Q” 5 more times, the game isn’t over. In fact, the Man doesn’t change.
7. If the player chooses at the start of the round to solve the puzzle, prompt the player to enter the solution. Treat the solution guess as case insensitive. If the guess is correct, the player wins! If the guess is wrong, the Man gets hanged and the player loses.
8. If the player chooses to surrender, the Man gets hanged and the player loses.
9. Regardless of whether the player wins or loses, the final output for the game should be the current game display with the puzzle completely unveiled. Also, you will announce whether the player won or lost.
10. At the end of each game, ask the player if he/she would like to play again. If so, prompt for the new puzzle and then start a new game (complete with Man and letters guessed completely reset).
11. If the player chooses not to play another game, thank the user for playing and then end the application.

# Rubric

Please note that the following items are graded in a binary fashion; either you met *all* the requirements of the item, or you didn’t. No partial points will be given for a single line item.

**Automatic Zero:** All coding is in the main method, the game cannot be played, or the application cannot be tested for any reason.

(10 points) The user can enter only valid puzzles

(20 points) The game display contains all required elements and functions as expected

(10 points) Guessing a letter works correctly

(5 points) Solving the puzzle works correctly

(5 points) Surrendering works correctly

(10 points) The Man is managed correctly

(10 points) The Letter Guessed collection is managed correctly

(10 points) Unveiling the puzzle is managed correctly

(10 points) Win/lose conditions work properly

(10 points) After a game ends, the player should be able to play again or choose to end the application