CS160 Computer Science I
Lab 10

Objectives
Work with functions
Work with lists
Work with strings

Assignment
Write a program that plays hangman.

Required Functionality
- Create a file that contains 20 words. You can do with a text editor, it doesn't have to be done with a Python program. Name it `words.txt` and save it in the same folder as your program.
- Must use at least 2 functions (including main)
- At the start of the program ask the user to select difficulty (easy (9), medium (7), hard (5)) – this will be the number of missed a player can have before they lose.
- At the beginning of the *program* fill a list with the words from `words.txt`. **DO NOT** ask the user for the name of the file. Only do this once each time the program runs.
- At beginning of each *game* randomly choose a work from the list of words to decide what word the person will try to guess
- Prompt the user to enter a letter
- After each letter input, show the following:
    - Incorrectly guessed letters.
    - Correctly guessed letters (where they are in the word).
    - Number of guesses remaining.
    - Feedback indicating if the most recent guess is correct or incorrect.
    - If the user has run out of guesses – if so, show the correct word and tell them they lose this game.
    - If the user has successfully guessed the work – if so, congratulate them.
- Challenge: Draw the hanged man each step as they guess wrong letters using ASCII art
- After game is over; ask the user if they wish to play another game. Repeat playing the game until the user indicates they wish to quit.

Hint
To keep track of the letters of the word, use a list of single letters rather than trying to maintain a single string.
You will need several lists, likely one for correct guesses, incorrect guesses, and maybe the "word", or at least as much of it has been guess, to display.
Remember that printing out individual letters from a list using a for loop, and setting end= to an empty string creates the same appearance as printing out a single string.

Sample Output (typed by me, NOT pasted from actual output)

```
Word: ******
Your guess? a
a is not in the word, 8 incorrect guesses remaining

Word: ******
Incorrect Letters: a
Your guess? p
p is in the word, 8 incorrect guesses remaining

Word: p*****
Incorrect Letters: a
Your guess? u
u is not in the word, 7 incorrect guesses remaining

Word: p*****
Incorrect Letters: a u
Your guess? y
y is in the word, 7 incorrect guesses remaining

Word: py****
```

and so on…

It could end like this if they get it right:

```
Word: pyt*on
Incorrect letters: a u
Your guess? h

You got it - the word is hangman
```

Or something like this is they don't get the word

```
Word: pyt*on
Incorrect letters: a u g r q w l i
Your guess? e

Sorry, that is your last incorrect guess
The word is hangman
```

And whether they win or lose ask the user if they wish to play another game

```
Do you want to play another game (yes/no)?
```