day34: hangman
Due: Wednesday 11/9/23

Today we are going to create a hangman game. If you recall, hangman is a game where you guess letters, trying to discover a secret word. The idea is to guess all the hidden letters before you get "hanged", and each wrong guess gets you a little bit closer to the end. A sample run is at the end of this document.

There are many ways to make a hangman game, and I'm going to suggest one way to you that I have found works well. You can do it a different way if you want.

Store the secret word in a variable, then create a list containing "_" underline characters, the same number of entries as your word has letters. When the user guesses a letter, replace the blank in the right spot with the letter the user guessed correctly, printing out the results each time. Keep track of how many bad guesses the user makes and end the game after 9 failed guesses.

Then, you'd need to find exactly where the letter occurs. A great way to do this would be to use a for loop to check every letter in the word. Remember, a letter may occur more than once in a word and all instances of the letter should be shown after a guess. The following code would do this; you'll obviously have to adapt it to your program for it to work:

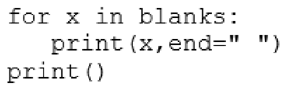
```
for x in range(len(secretWord)):
    if secretWord[x]==guess:
        blanks[x]=guess
```

Here is a checklist:

- Create a repl.
- Put "import random" up at the top of your program.
- Create a list variable with a bunch of words in it. For example,
  wordList = ["cat","hamster","zebra"] Make your list have at least 5 words.
- Do a while True:
- Choose a word randomly from the list to be the secret word like this: secretWord = random.choice(wordList)
- Make the secret word and all guesses uppercase. For example: secretWord = secretWord.upper()
- Create a list with the right number of blanks in it: blanks = ['_'] * len(secretWord)
- Make an empty list called guesses: guesses = []
- Create a variable called badGuesses and set it to 0
- Do another while True: loop

(continued on next page)

- First thing in the loop, print the list of blanks. We want the blanks printed on a single line with spaces between them. You can do that like this:



```
for x in blanks:
   print(x,end=" ")
print()
```

- After printing the blanks, on the next line print "Already guessed: " and then print the list of guessed letters.

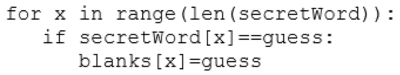
- Ask the user to make a guess. This would be guess = input("Please make a guess.").upper()
Notice that I made the guess uppercase. We're doing all the words and letters uppercase to keep things easier to code.

- Check if the user has already guessed that letter: if guess in guesses:
If they have already guessed that letter, tell them they already guessed it, and then use the continue command to start this turn over again.

- If they had not guessed that letter before, add the guess to the guesses list: guesses.append(guess)

- Check if the new guess is in the secretWord: if guess in secretWord:

  If the guess IS in the secretWord, replace all blanks where the letter should go with the letter. This is probably the trickiest part of today's program. Here is code that could do this; you will have to adapt this to your variable names:



```
for x in range(len(secretWord)):
   if secretWord[x]==guess:
      blanks[x]=guess
```

  If the secret word was Analy and the user guessed A, this would change _ _ _ _ _ to A _ A _ _

  If guess NOT in secretWord, add one to badGuesses. Check if badGuesses is equal to 9. If it is, tell the user they have lost, print the secret word, and do a break to get out of the inner loop. After the inner loop, add code to ask if the user wants to run again. If they say no, say goodbye and break and the game is over.

- Check to see if the user has guessed the entire word.

  You can do this by doing this: if "_" not in blanks:
If there are no more _ characters in the list called blanks, then the user guessed the entire word. Congratulate them, and break out of the inner loop.

- Just after the inner loop, as already mentioned above, ask the user if they would like to go again. If they choose not to, say goodbye and break. Otherwise, your code should loop back up, choose a new secret word, reset the bad guess count and all other important variables and start over.

- Make sure your program works. Test it, test saying that you want to run again. Make sure it works with a new word. Test saying that you do not want to run again. Make sure it quits.

- Include a copy of all of your testing efforts in triple quotes after your code so I know you tested your program.

  (continued on the next page)

For the record, I know you can go out on the internet and find a hangman program written in Python without doing any work. Doing this would be cheating. Please do your own work.

Optional extra credit:

After everything else is working, make your program print out ascii art of a hangman or something else fun. Some fun examples are at the end of this document.

If you get everything working and want to print a hangman art on each turn, you can use the badGuesses variable to see which piece of art to print (with the user hanging more and more with each new badGuess.) This is optional. Please get your entire program to work before you spend time trying to print artwork. The art part of this is optional.

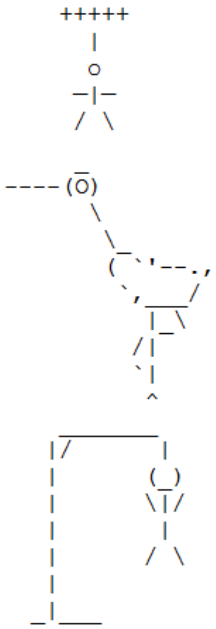
Sample run on next page.

Sample run:

```
        Welcome to hangman.

        _ _ _ _ _ _ _
        Already guessed:
        9 guesses remaining.

        Please guess a letter: h

        _ H _ _ _ _ _
        Already guessed:  H
        9 guesses remaining.

        Please guess a letter: a

        _ H _ _ _ _ _
        Already guessed:  H A
        8 guesses remaining.

        Please guess a letter: b

        _ H _ _ _ _ _
        Already guessed:  H A B
        7 guesses remaining.

        Please guess a letter: c

        C H _ C _ _ _
        Already guessed:  H A B C
        7 guesses remaining.

        Please guess a letter: n

        C H _ C _ _ N
        Already guessed:  H A B C N
        7 guesses remaining.

        Please guess a letter: e

        C H _ C _ E N
        Already guessed:  H A B C N E
        7 guesses remaining.

        Please guess a letter: i

        C H I C _ E N
        Already guessed:  H A B C N E I
        7 guesses remaining.

        Please guess a letter: k
        You won! Congratulations.
        The secret word was CHICKEN
        Play again? (y/n) n
        Goodbye
```

Sample art ideas on next page

Sample art ideas for extra credit when/if you finish the core program:



```
    +++++
      |
      o
     -|-
     / \

     _
----(O)
      \
       \_
       ( `'--.,
        `,___/
          |_\
         /|
         `|
          ^
    ________
   |/      |
   |      (_)
   |      \|/
   |       |
   |      / \
   |
  _|___
```