import random

# Returns a randomly selected word

def choose\_word():

words = ["python", "coding", "portfolio", "courses", \

"programming", "code", "data", "visual", "studio"]

# choice() returns a randomly selected word from words list

return random.choice(words)

# Returns a string representing the status of the word to guess with respect

# to the guess\_letters so far. i.e. if the word to guess is "python" and

# the guessed\_letters so far are ["y", "t"] then \_yt\_\_ would the status of

# the word to guess.

def word\_status(word, guessed\_letters):

# Loop through the word one letter at at time, concatenating the letter

# to the display string if it IS a guess\_letter and \_ if it is not

display = ""

for letter in word:

if letter in guessed\_letters:

display += letter

else:

display += "\_"

return display

# Runs the word guessing game

def word\_guessing\_game():

# Initial game status: pick a secret word, no guess letters yet, and give

# the player 7 attempts to guess a correct letter

secret\_word = choose\_word()

guessed\_letters = []

attempts = 7

# Output the game title and initial status of the word to guess which will

# be all \_ characters BUT this will tell the player how many letters are in

# the word total.

print("Word Guessing Game")

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

print("Secret Word:", word\_status(secret\_word, guessed\_letters))

# Continue the game so long as the player has attempts remaining

while attempts > 0:

# Prompt the user to enter their next guess/letter, use lower() to

# make it lowercase regardless of what case the player enters. This

# is just so when checking if the letter is in the word we don't run

# recognize they guessed correctly due to the player using uppercase

# comparing against words that are lowercase.

guess = input("Guess A Letter: ").lower()

# If the length of the string entered is not 1 and all the characters

# in the string are not letters, output an error message, and use

# continue to skip execution of the rest of the loop body.

if len(guess) != 1 or not guess.isalpha():

print("You must enter a single letter.")

continue

# Also output an error message and skip the remainder of the loop body

# if the guessed letter is a previously guessed letter.

if guess in guessed\_letters:

print("You already guessed that letter.")

continue

# Otherwise if it is a new single letter that was guessed add it to the

# list of guessed letters.

guessed\_letters.append(guess)

# If the guessed letter is NOT in the word to be guessed the player

# loses an attempt and we output a message informing them of this

# and the number of remaining attempts they have left.

if guess not in secret\_word:

attempts -= 1

print(f"No letter '{guess}' occurs in the word.")

print(f"You have {attempts} attempts remaining.")

# Otherwise if the guessed letter IS in the word we can output the

# number of time it occurs.

else:

occurrences = secret\_word.count(guess)

print(f"Letter '{guess}' occurs {occurrences} times.")

# Output the new current status of the word

current\_status = word\_status(secret\_word, guessed\_letters)

print("Secret Word:", current\_status)

# If there are no remaining \_ characters in the status of the word this

# means the player has guessed all the letters and has won the game, we

# output a congratulations message and use break to stop the loop (and

# end the game).

if "\_" not in current\_status:

print("Congratulations! You guessed the word.")

break

# If the player runs out of attempts and there is an \_ character in the

# status of the word this means the user failed to guess all the letters

# before running out of attempts. In this case we just inform the user

# of this and output what the word to be guessed was.

if "\_" in current\_status:

print(f"You ran out of attempts! The word was: {secret\_word}")

# Call the function to play the game

word\_guessing\_game()