# Pranav Abbaraju

# Hangman Game

import random

#\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

words = [] #empty list

def fileLoader(): #loads the txt file with the hangman words and adds them to the list

with open('hangmanWords.txt', mode='r') as y:

for x in range(0, 58108):

words.append(y.readline().strip())

fileLoader()

#\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

def instructions(): #method that prints the instructions for playing Hangman

print

print

print "HANGMAN"

print

print "The goal of Hangman is for you to correctly guess the word making sure that the man is not hanged. \nThe dashes below represent the length of the word, and you must guess the letters it contains. \nFor each failed attempt, one more part will be added to the hanger. \nIf all the parts of the man are added, you will lose."

print

print

#\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

def hangman(hang): #method that controls how hangman will be printed

if hang == 0:

return " \_\_\_\_\_\_\_\_\_\_\n |\t\t \ |\n\t\t \|\n\t\t |\n\t\t |\n\t\t |\n\_\_\_\_\_\_\_\_\_\_\_|"

elif hang == 1:

return " \_\_\_\_\_\_\_\_\_\_\n |\t\t \ |\n O\t\t \|\n\t\t |\n\t\t |\n\t\t |\n\_\_\_\_\_\_\_\_\_\_\_|"

elif hang == 2:

return " \_\_\_\_\_\_\_\_\_\_\n |\t\t \ |\n O\t\t \|\n |\t\t |\n\t\t |\n\t\t |\n\_\_\_\_\_\_\_\_\_\_\_|"

elif hang == 3:

return " \_\_\_\_\_\_\_\_\_\_\n |\t\t \ |\n O\t\t \|\n |\t\t |\n \\\t |\n\t\t |\n\_\_\_\_\_\_\_\_\_\_\_|"

elif hang == 4:

return " \_\_\_\_\_\_\_\_\_\_\n |\t\t \ |\n O\t\t \|\n |\t\t |\n / \\\t |\n\t\t |\n\_\_\_\_\_\_\_\_\_\_\_|"

elif hang == 5:

return " \_\_\_\_\_\_\_\_\_\_\n |\t\t \ |\n O\t\t \|\n |\\\t |\n / \\\t |\n\t\t |\n\_\_\_\_\_\_\_\_\_\_\_|"

elif hang == 6:

return " \_\_\_\_\_\_\_\_\_\_\n |\t\t \ |\n O\t\t \|\n /|\\\t |\n / \\\t |\n\t\t |\n\_\_\_\_\_\_\_\_\_\_\_|"

#\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

game = 1 #game variable that determines if the game will run

play = raw\_input("Enter 'play' to begin playing Hangman: ") #asks the player to type and enter 'play' to begin

playCheck = 0 #controls the computer asking the user again if 'play' is not entered correctly

while playCheck == 0:

if play == "play" or play == "Play" or play == "PLAY" or play == "p" or play == "P":

instructions() #if 'play' is entered correctly, then the instructions method is called

playCheck = 1 #makes sure the while loop doesn't run again and the next while loop runs

else:

play = raw\_input("Please enter 'play': ") #if 'play' is not entered correctly, player is asked to enter it correctly

#\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

#\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

#\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

while game == 1 and playCheck == 1:# main loop of the game: it will keep running as long as game variable is 1

#\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

hang = 0 #determines the actual hangman

pos = 0 #determines the position of a letter in the new output

randomNum = random.randint(1, 58108) #

word = words[randomNum] #random number between 1 and the number of lines in the txt file (58108), and that word that number represents is chosen from the list

#this is for testing the hangman code

print word

length = len(word) #length of the randomly chosen word

blanks = length\*" \_\_\_ " #these are the blank dashes for the letters

spaces = blanks.count(" ") #this is the number of spaces between the dashes

output = length\*" " #this is the output of letters that go on the dashes

originalOutput = output # this is a duplicate of output variable for later use

incorrectLetters = "" #empty string of the failed guesses

#\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

while output.count(" ")%spaces != 0 and hang != 6: # runs as long as all the letters have not been guessed correctly, and the player has not reached the maximum number of failed attempts

print hangman(hang) #more body parts added on the more times the player fails to guess the correct letter

print

print output #output is blank the first time; letters are added as player guesses them correctly

print blanks #dashes below the letters

print

letter = ""

letter = raw\_input("Enter a letter from a to z: ") #user input to guess a letter

letterCheck = 0 #variable that will control the computer asking the user again if he/she did not enter a valid letter

alphabet = "abcdefghijklmnopqrstuvwxyz"

while letterCheck == 0:

letter = letter.lower()#converts letter to lowercase

if len(letter) == 1 and alphabet.find(letter) != -1: #if it is one letter

if output.find(letter) != -1 or incorrectLetters.find(letter) != -1: #if the player has already guessed the letter

letter = raw\_input("You have already guessed this letter, please try again: ")

letterCheck = 0 #makes sure the new user input goes through this loop again to check if it is a valid letter

else: # if the player has not guessed this letter yet

print

letterCheck = 1 #makes sure that this loop doesn't run again right after, because the player has entered a valid letter

if word.find(letter) == -1: #if the letter is not part of the word

hang += 1 # hang variable increases by one (one more failed attempt)

incorrectLetters += letter # the incorrect letter is added to the list

else: # at this point, we know that the player has entered one letter that he/she has not already guessed, and that the letter is in the word

for x in range(len(word)): #this loop runs from 0 to the length of the word

if word[x]==letter: # if a certain index of the word is the letter that the player entered

pos = (x\*5)+2 # the position of the letter in the new output that goes above the dashes

output = output[0:pos] + letter + output[pos+1:len(output)] # the part of the output up to the position of the letter is sliced and added, the letter is added, and then the rest of the output is sliced and added (this is reassigned to the output variable)

else: # this is if the first if statement's condition failed (meaning it is not a valid letter)

letter = raw\_input("Please enter a valid letter from a to z: ")

letterCheck = 0 #makes sure the new user input goes through this loop again

#\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

#when the code has reached this point, it means the main game loop has been exited either because the player lost or won

if output.count(" ")%spaces == 0: #if the player has won (checks if the number of spaces in the finished output is double the spaces between the blanks)

print hangman(hang)

print

print output #output of the complete word is printed

print blanks #dashes below the output are printed

print

print "YOU WON!"

print

if hang == 6: # if the player has reached the maximum number of failed attempts (the man has been hanged)

print

print hangman(hang)

print

print "GAME OVER"

print

print "The correct word was " + word + "." #the correct word is printed

print

#\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

game = 2 #variable that controls the computer asking the player if he/she wants to play again; also controls if the game will run again (at the check for the while loop this whole game is a part of)

gameAgain = raw\_input("Do you want to play again? (Enter 'yes' or 'no'): ")

while game == 2: #keeps running until the player has entered yes or no

if gameAgain == "yes" or gameAgain == "Yes" or gameAgain == "YES" or gameAgain == "Y" or gameAgain == "y":

game = 1

elif gameAgain == "no" or gameAgain == "No" or gameAgain == "NO" or gameAgain == "N" or gameAgain == "n":

game = 0

else:

gameAgain = raw\_input("Please enter 'yes' or 'no': ")

print

print

print

print

#\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_