**Report**

This program was fun to make, but kind of threw me in a loop for a while. I read program 9 before I even started this, and I just went blank for a while because all I could think about was program 9. After a few hours of staring off into space, fingers started hitting keys and a program was born. There were some issues in the process however. Problems such as figuring out how to add WIN and LOSS to their own separate lines in the high score file. Then I had to figure out how to take those wins and losses and make them appear in numeric form in the high score selection from the menu. The other stuff was quite easy when I used my brain compartments and Google powers. I hope you enjoyed/will enjoy (depending on when you’re reading this) my Hangman game. I know I did! \*thumbs up emoji\*

**Algorithm**

import random and os library

define welcomeScreen

print welcome message

print different options and allow user to pick which he/she wants to play

if statement to run game module if 1 is entered by user

elif to run displayScore module if 2 is entered by user

else statement to quit if 1 or 2 isn't entered

define randomWordPicker

import the random words, iFile, I created in a txt file

read from the file

create variable, word to generate a random choice from iFile

make the selected word lowercase

return the word

define game

create variable maxScore to 0

make word variable equal randomWordPicker module

Empty variables correctGuesses and incorrectGuesses

make playAgain equal to 'Y'

gameplayed equal to 0

Allow 10 guesses to the user

print a good luck message for user

while loop to allow the user to play 4 games if playAgain continues to be 'Y'

create empty value for displayWord

for loop for char in word

if statement to check if characters are in correctGuesses

if true, then displayWord equals displayWord plus char plus empty space ('')

if false, displayWord equals displayWord plus '-'

print displayWord

if you run out of guesses

add game to gamesPlayed

saveScore is False

print a losing statement

if statement to see if you've played the max number of games

print max games statement

break loop

create playAgain to ask user if he/she wants to play again and make that input uppercase

generate a new random word from randomWordPicker

allot 10 new guesses

reset the correct and incorrect guesses

continue

if you win the game

add game to gamesPlayed

print winning statement

saveScore is True

if statement to see if you've played the max number of games

print max games statement

break loop

Ask user if he/she wants to play again and make their input uppercase

generate a new random word from randomWordPicker

allot 10 new guesses

reset the correct and incorrect guesses

continue

print newline

guess variable to ask user to guess a letter or word for the game

if statement to find if a guess has already been used

print statement saying you've already user this letter/word

subtract a guess

else

if the guess is in the random word

add the guessed letter to the correctGuesses var

else

add the guessed letter to incorrectGuesses var

print a row to show your correct guesses and incorrect guesses

return to the welcomeScreen

define saveScore(win)

data equals empty value

if statement to find if highscore.txt file is available

if it is, read the file as data

if you won a game, write WIN to data

else

write LOSS to data

open highscore.txt as a writeable file

write data into file

close file

define displayScore

create empty values for wins and losses, both equal to 0

if statement to find if highscore.txt file is available

if so, open highscore.txt and rename as f

content variable equals realines of f (highscore.txt)

for the lines in content

if 'WIN' is in the line

wins equals wins plus 1

elif 'LOSS' is in the line

losses equals losses plus 1

print newline

print statement saying however many games you've won

print statement saying however many games you've lost

else highscore.txt isn't found

print no scores found

run welcomeScreen

define main

run welcomeScreen

run main