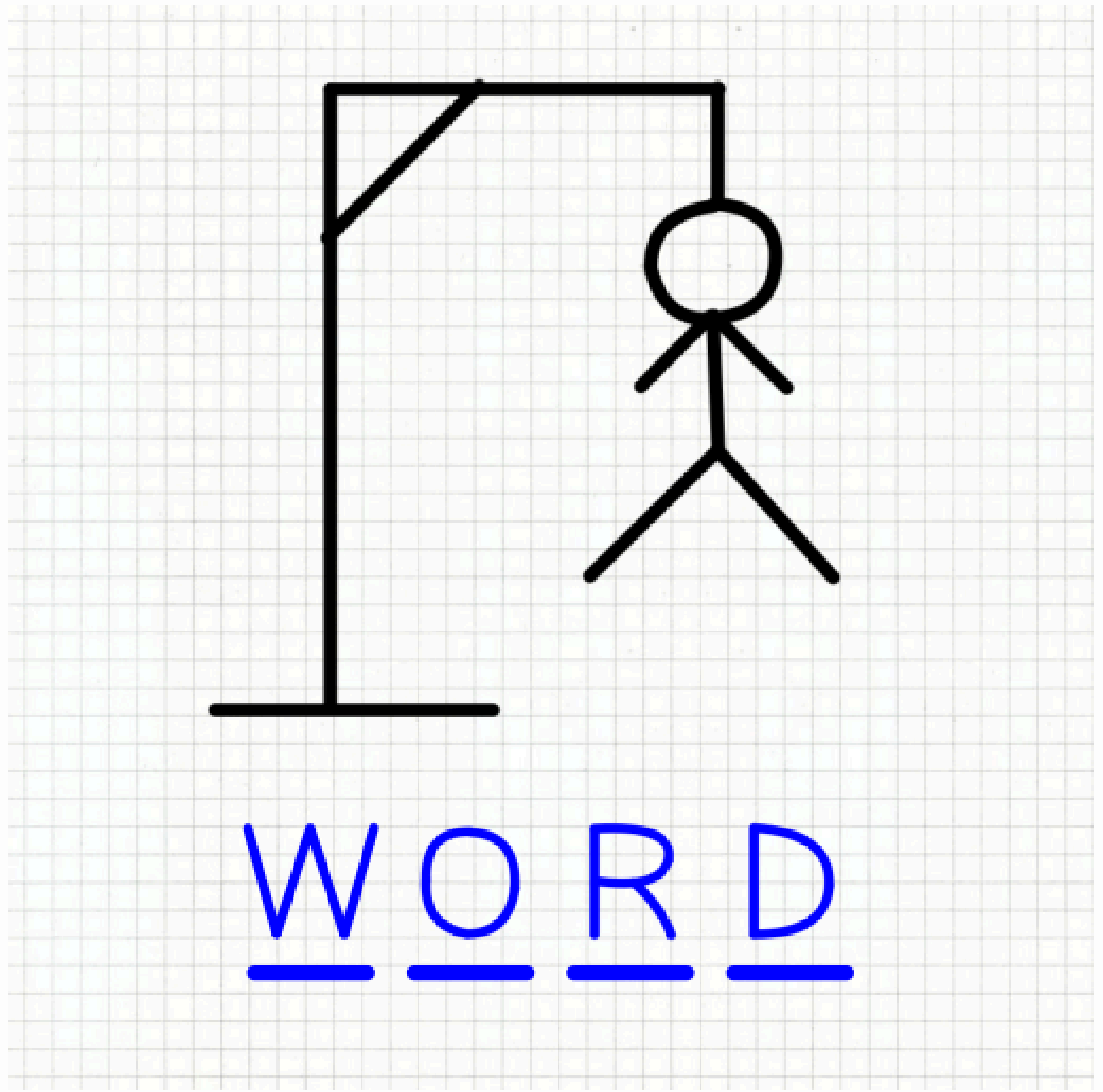
Hangman

Presented By:

- P.Pavani (23EG105P51)
- P.Sai Tejasri (23EG105P52)
- P.Vijay Naidu (23EG105P53)
- Pradeep(23EG105P54)
- Nikhil(23EG105P55)



Introduction

Hangman is a word-guessing game where the player tries to guess a hidd word by suggesting letters. For each incorrect guess, a part of the stick figure is drawn until it is complete, resulting in the "hanging" of the figure (but here there are only eight chances to guess the word).

How to play Hangman

Playing Hangman is simple and engaging. Players need to understand the rules of the game, follow the instructions, and use their vocabulary and deduction skills to guess the hidden word. The game is played in turns, with the player guessing a letter in each turn.

- The game starts by selecting a secret word randomly chosen by program.
- The player then guesses a letter that they think is in the word. Each correct guess reveals the letter's position in the hidden word.
- The process of guessing letters continues until the player either correctly guesses the entire word or the stick figure is fully drawn, indicating a loss.

Rules

Rule 1: Alphabet Only
The player can only guess individual letters from the alphabet, one at a time.

Rule 2: No Repeating Letters
Players cannot guess the same letter multiple times in a single game. This prevents players from simply guessing all letters in the alphabet.

Rule 3: Letters Only
Players cannot guess whole words or phrases. They are limited to guessing individual letters only.

Rule 4: Guessing Ends When Chances are over
The player can continue to guess letters until given chances are over after which the game is over and the player loses.

Overview of Python

Python is a high-level, interpreted language known for its simplicity and broad applicability. Created by Guido van Rossum in 1991, it's widely used for web development, data science, machine learning, automation, and more.

Key Features:

- Readable: Easy-to-read syntax, great for beginners.
- Versatile: Supports tasks from web development to AI.
- Interpreted: Executes code line-by-line, simplifying debugging.
- Dynamic Typing: No need for explicit data type declarations.
- Large Library: Built-in support for tasks like regex and networking.
- Cross-Platform: Runs on Windows, macOS, and Linux without changes.

Popular Libraries:

- Web Development: Django, Flask
- Data Science/ML: Pandas, TensorFlow, Scikit-learn
- Automation: Selenium
- GUI: Tkinter, PyQt

Why Choose Python?

- Easy to learn, vast library support, strong community, ideal for all scales of development.

Concepts Used in Implementing

Conditional Statements :

Conditional statements like if,if else,if elif else are used to check whether the given letter by user exists in the word or not.

Loops:

While loop is used to iterate until the given chances are comoketed and gives the final result of the game.

Functions:

Functions are used to make the code easy to understand and increases the reusability of the code.There are 6 functions defined in this code .

Words.txt:

This progam uses the external file words.txt in which the words for the game are defined already.

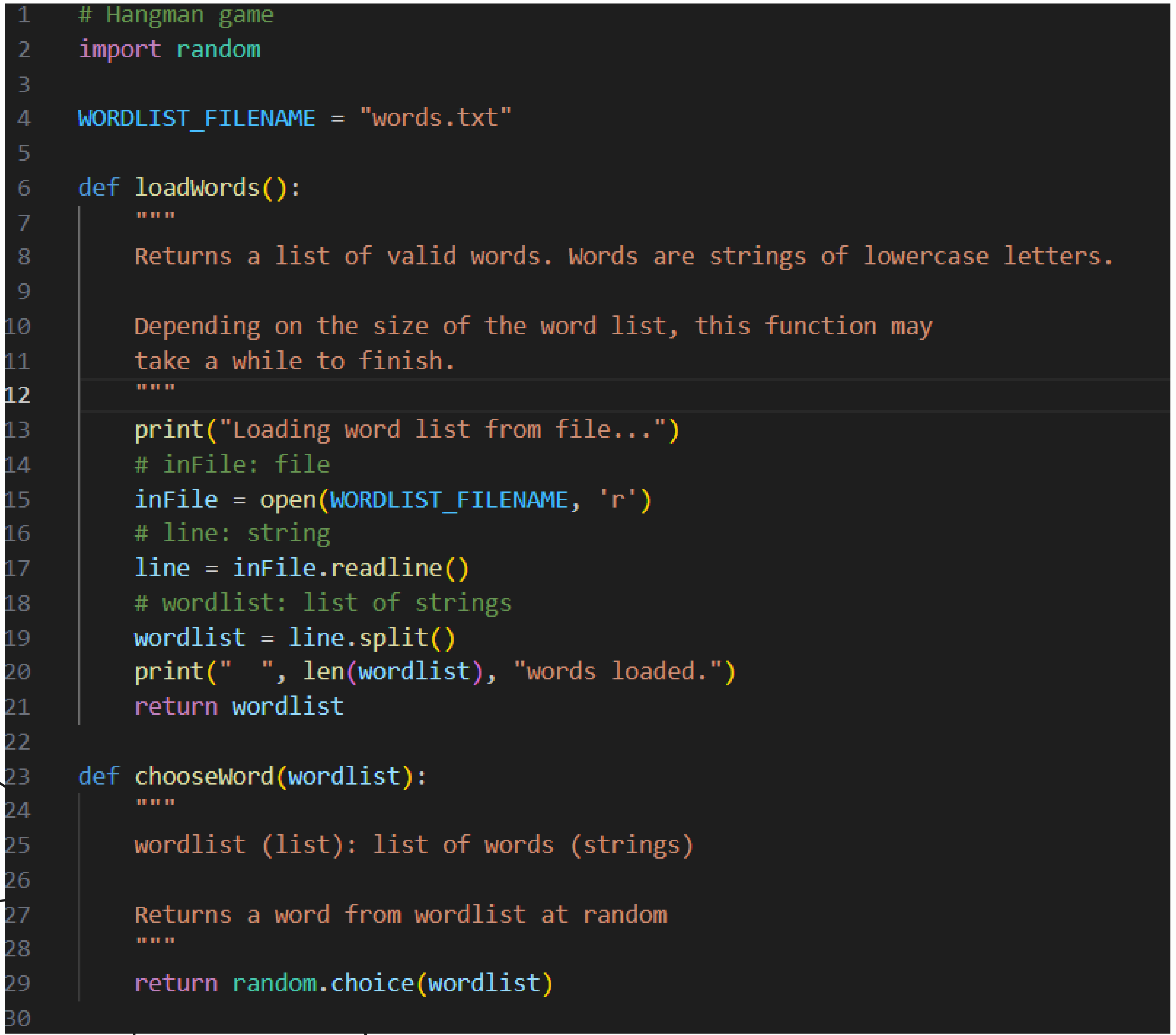
Random module:

Random is a module in which it is used to choose a word ramdomly from the text file .

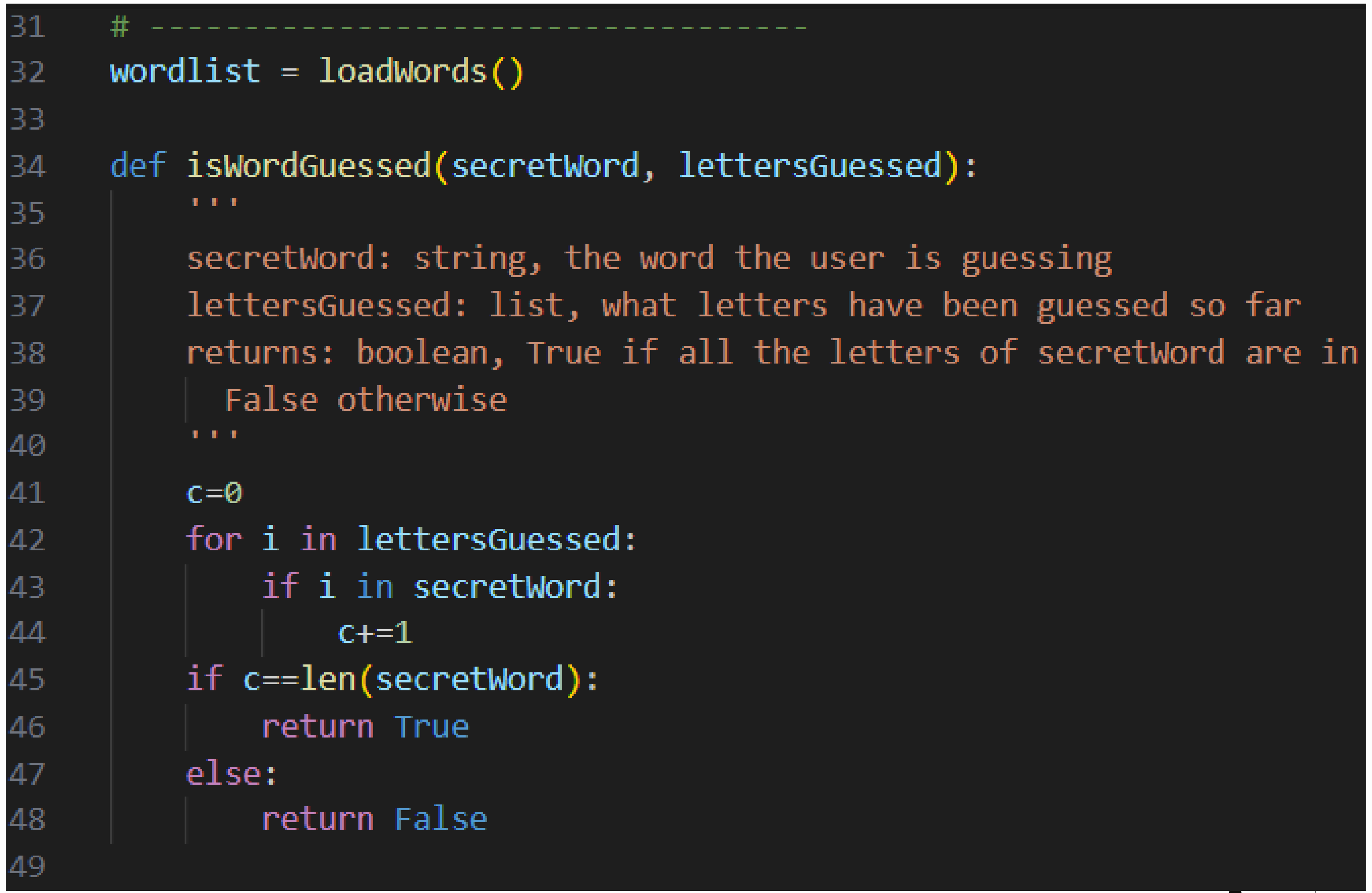
String module:

String module is used to do string manupulation or any other operations by usinf the predefined functions in the string module.

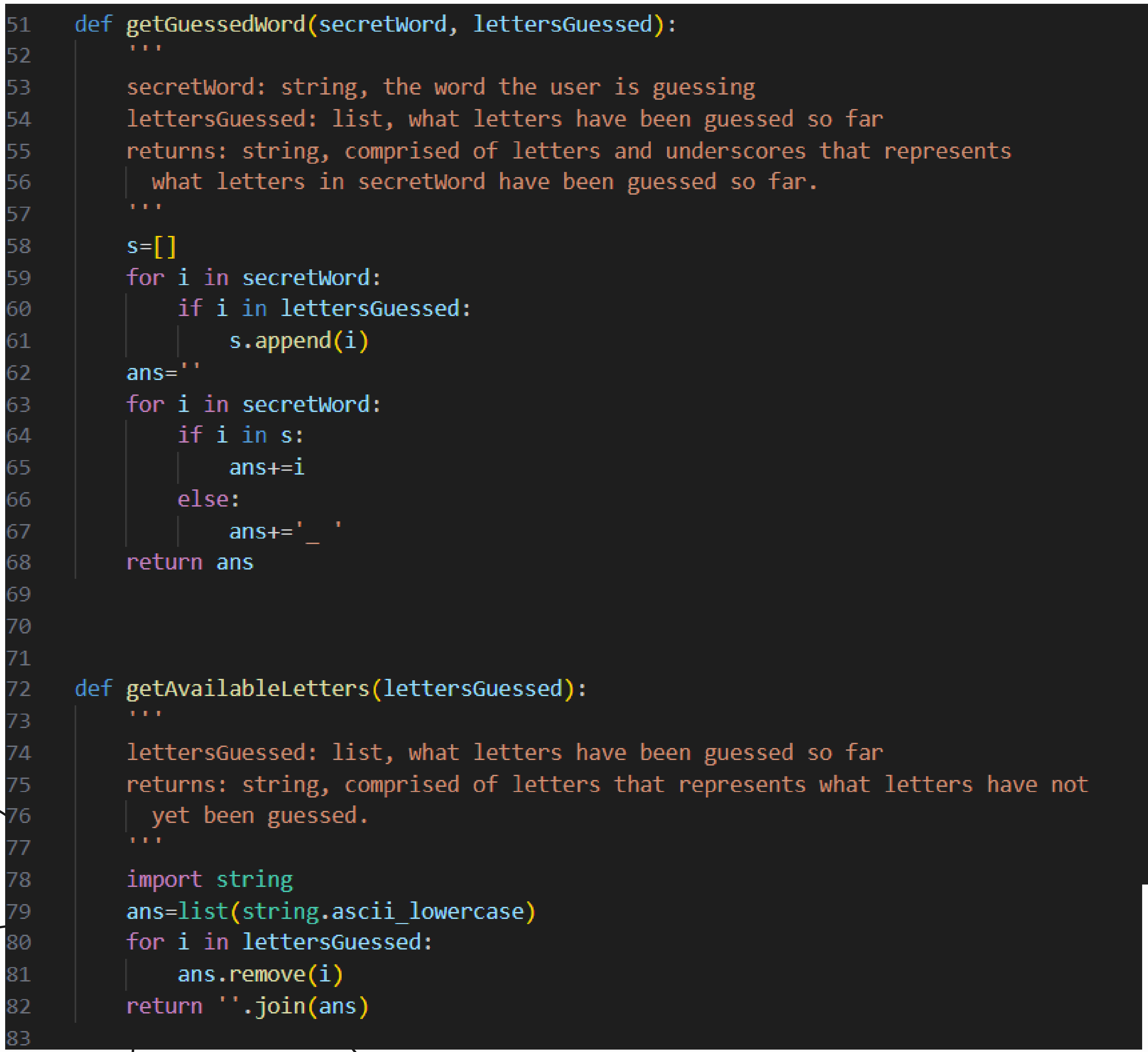
Code



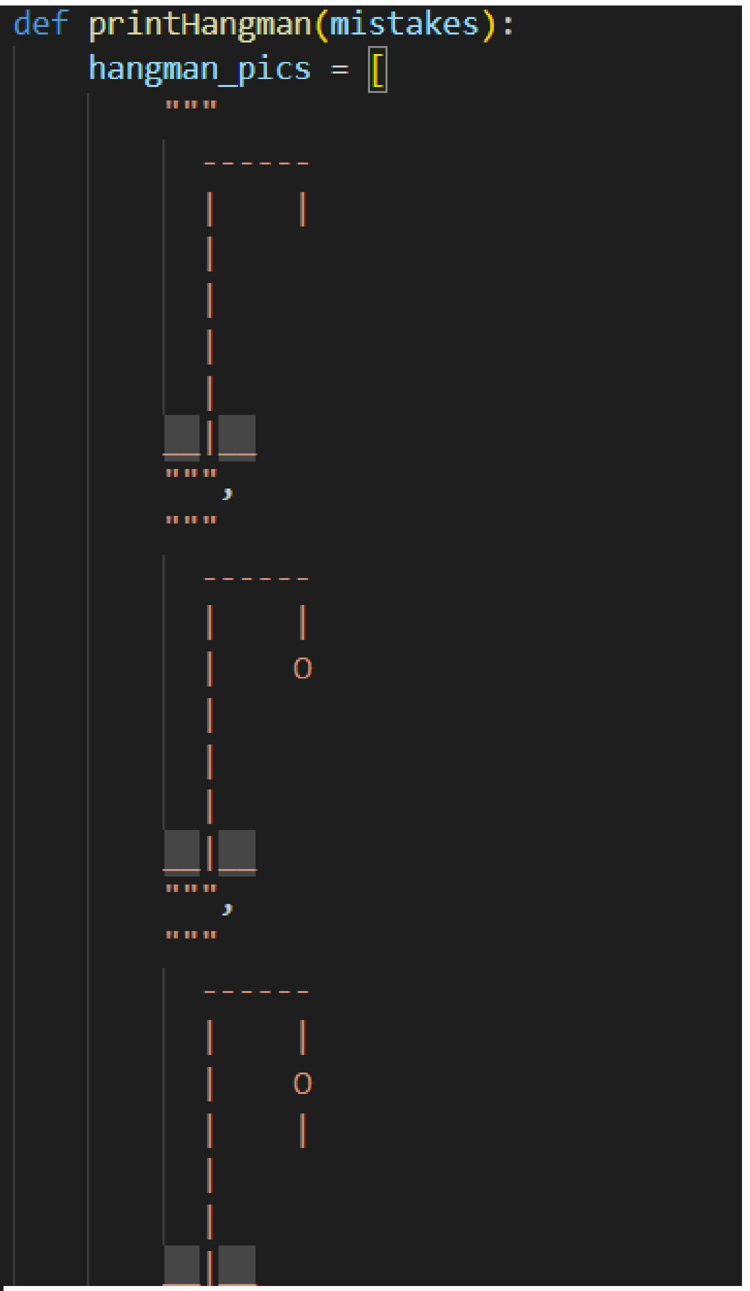
```python
# Hangman game
import random

WORDLIST_FILENAME = "words.txt"

def loadWords():
    """
    Returns a list of valid words. Words are strings of lowercase letters.

    Depending on the size of the word list, this function may
    take a while to finish.
    """
    print("Loading word list from file...")
    # inFile: file
    inFile = open(WORDLIST_FILENAME, 'r')
    # line: string
    line = inFile.readline()
    # wordlist: list of strings
    wordlist = line.split()
    print("  ", len(wordlist), "words loaded.")
    return wordlist

def chooseWord(wordlist):
    """
    wordlist (list): list of words (strings)

    Returns a word from wordlist at random
    """
    return random.choice(wordlist)

```



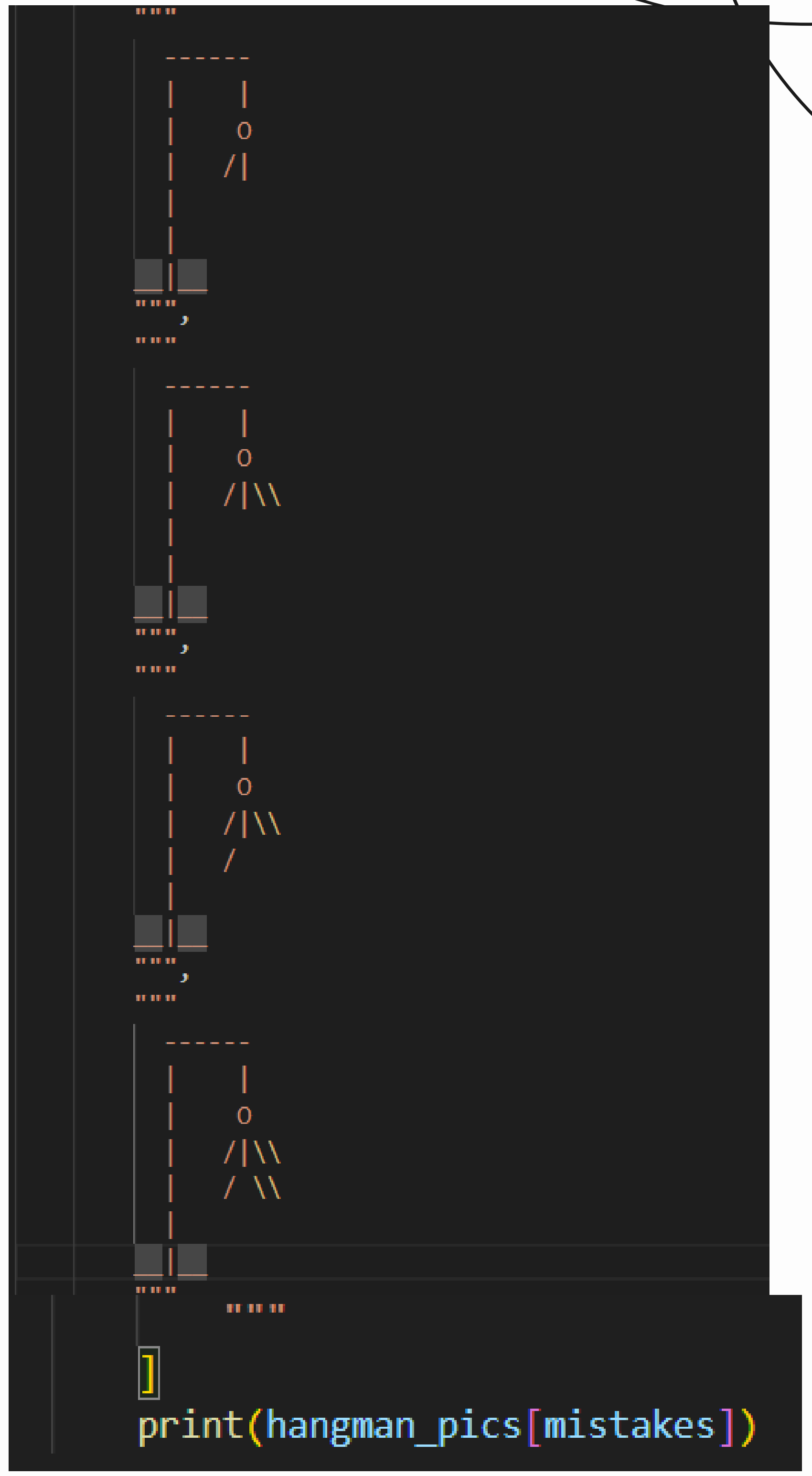
```python
# -----------------------------------
wordlist = loadWords()

def isWordGuessed(secretWord, lettersGuessed):
    '''
    secretWord: string, the word the user is guessing
    lettersGuessed: list, what letters have been guessed so far
    returns: boolean, True if all the letters of secretWord are in
      False otherwise
    '''
    c=0
    for i in lettersGuessed:
        if i in secretWord:
            c+=1
    if c==len(secretWord):
        return True
    else:
        return False

```



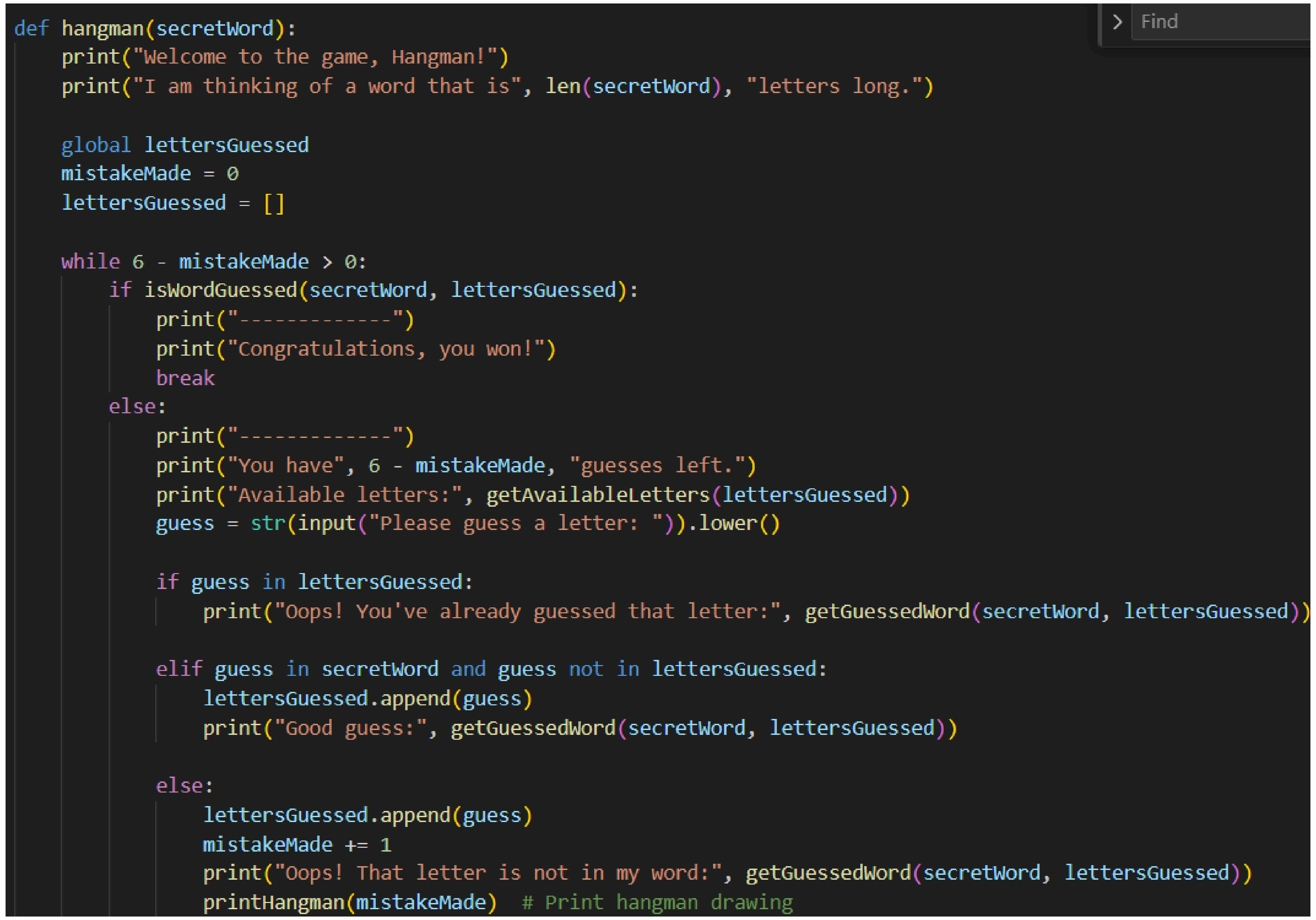
```python
def getGuessedWord(secretWord, lettersGuessed):
    '''
    secretWord: string, the word the user is guessing
    lettersGuessed: list, what letters have been guessed so far
    returns: string, comprised of letters and underscores that represents
      what letters in secretWord have been guessed so far.
    '''
    s=[]
    for i in secretWord:
        if i in lettersGuessed:
            s.append(i)
    ans=''
    for i in secretWord:
        if i in s:
            ans+=i
        else:
            ans+='_ '
    return ans



def getAvailableLetters(lettersGuessed):
    '''
    lettersGuessed: list, what letters have been guessed so far
    returns: string, comprised of letters that represents what letters have not
      yet been guessed.
    '''
    import string
    ans=list(string.ascii_lowercase)
    for i in lettersGuessed:
        ans.remove(i)
    return ''.join(ans)
```



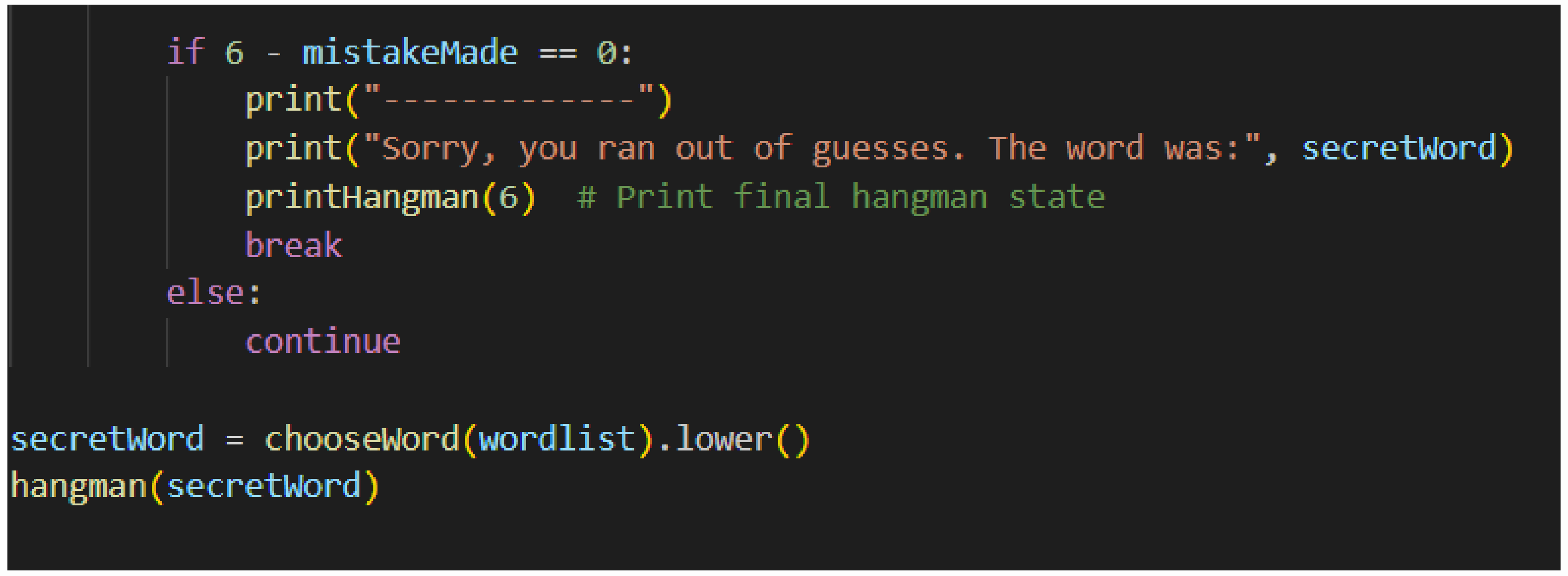
```python
def printHangman(mistakes):
    hangman_pics = [
        """
          ------
          |    |
          |
          |
          |
          |
        ___|___
        """,
        """
          ------
          |    |
          |    O
          |
          |
          |
        ___|___
        """,
        """
          ------
          |    |
          |    O
          |    |
          |
          |
        ___|___
```



```python
        """
          ------
          |    |
          |    O
          |   /|
          |
          |
        ___|___
        """,
        """
          ------
          |    |
          |    O
          |   /|\\
          |
          |
        ___|___
        """,
        """
          ------
          |    |
          |    O
          |   /|\\
          |   /
          |
        ___|___
        """,
        """
          ------
          |    |
          |    O
          |   /|\\
          |   / \\
          |
        ___|___
        """    """

        ]
        print(hangman_pics[mistakes])
```

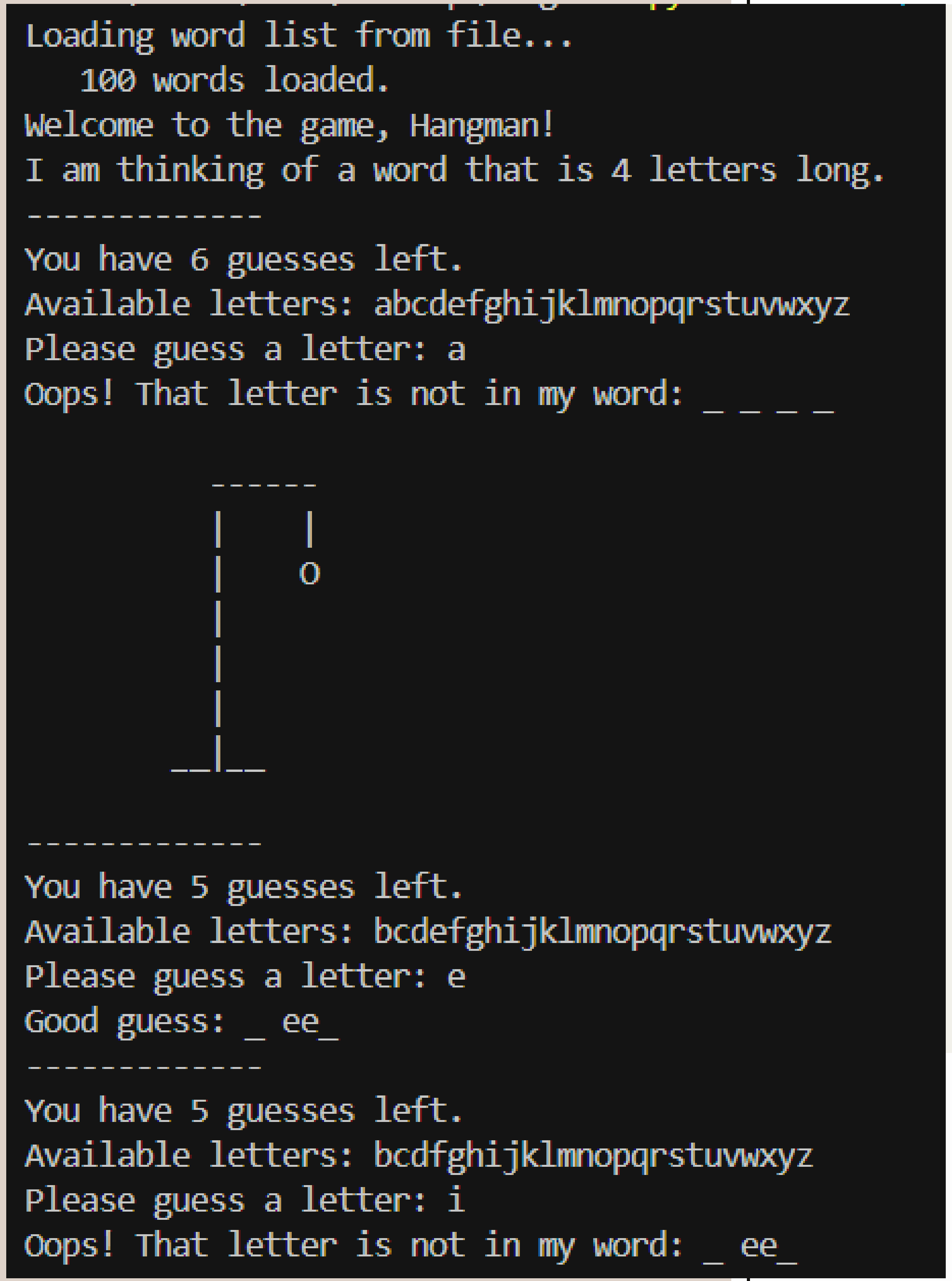


```python
def hangman(secretWord):
    print("Welcome to the game, Hangman!")
    print("I am thinking of a word that is", len(secretWord), "letters long.")

    global lettersGuessed
    mistakeMade = 0
    lettersGuessed = []

    while 6 - mistakeMade > 0:
        if isWordGuessed(secretWord, lettersGuessed):
            print("-------------")
            print("Congratulations, you won!")
            break
        else:
            print("-------------")
            print("You have", 6 - mistakeMade, "guesses left.")
            print("Available letters:", getAvailableLetters(lettersGuessed))
            guess = str(input("Please guess a letter: ")).lower()

            if guess in lettersGuessed:
                print("Oops! You've already guessed that letter:", getGuessedWord(secretWord, lettersGuessed))

            elif guess in secretWord and guess not in lettersGuessed:
                lettersGuessed.append(guess)
                print("Good guess:", getGuessedWord(secretWord, lettersGuessed))

            else:
                lettersGuessed.append(guess)
                mistakeMade += 1
                print("Oops! That letter is not in my word:", getGuessedWord(secretWord, lettersGuessed))
                printHangman(mistakeMade)  # Print hangman drawing
```

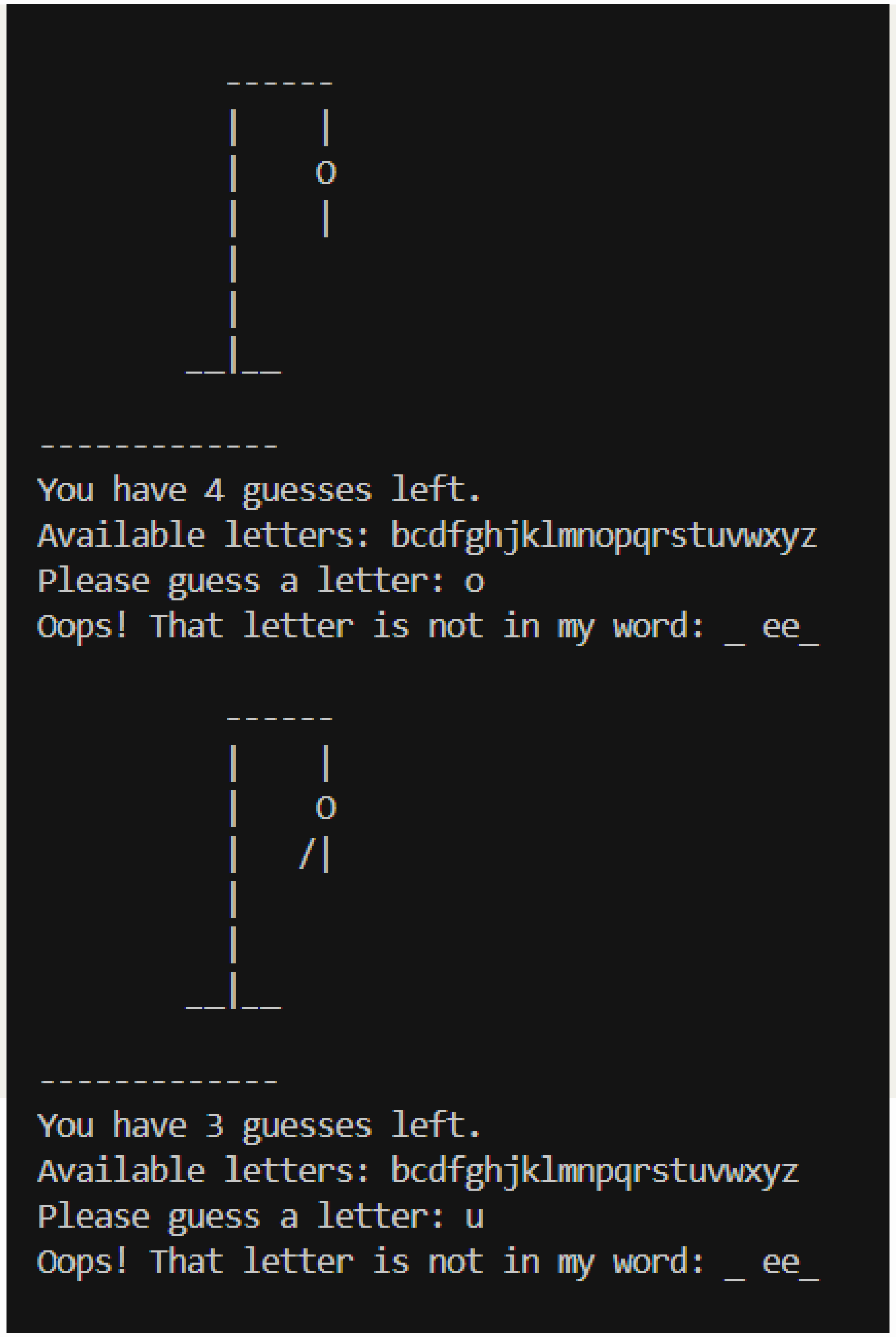


```python
            if 6 - mistakeMade == 0:
                print("-------------")
                print("Sorry, you ran out of guesses. The word was:", secretWord)
                printHangman(6)  # Print final hangman state
                break
            else:
                continue

secretWord = chooseWord(wordlist).lower()
hangman(secretWord)
```

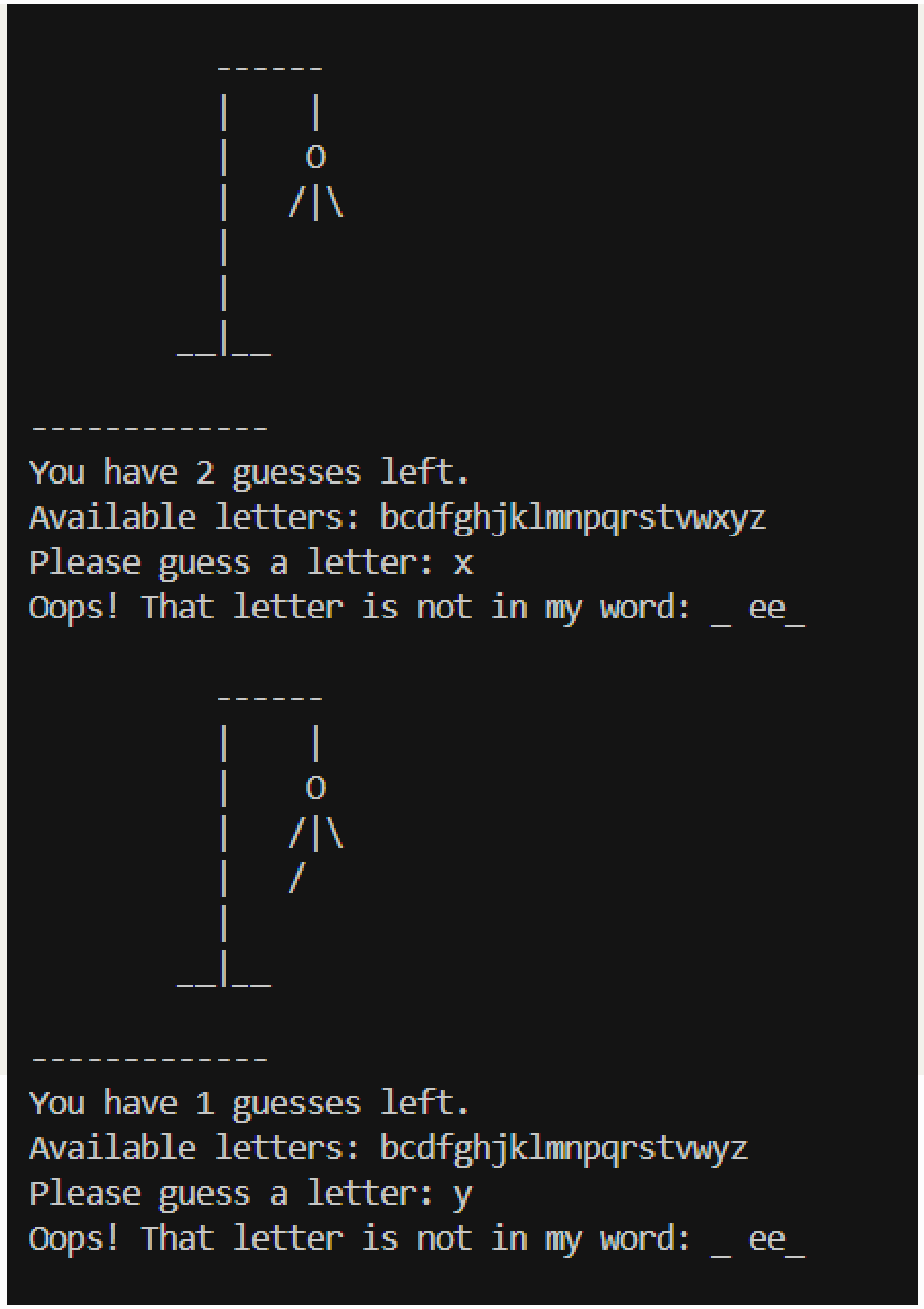
OUTCOMES



```
Loading word list from file...
  100 words loaded.
Welcome to the game, Hangman!
I am thinking of a word that is 4 letters long.
-------------
You have 6 guesses left.
Available letters: abcdefghijklmnopqrstuvwxyz
Please guess a letter: a
Oops! That letter is not in my word: _ _ _ _

          ------
          |    |
          |    O
          |
          |
          |
        __|__

-------------
You have 5 guesses left.
Available letters: bcdefghijklmnopqrstuvwxyz
Please guess a letter: e
Good guess: _ ee_
-------------
You have 5 guesses left.
Available letters: bcdfghijklmnopqrstuvwxyz
Please guess a letter: i
Oops! That letter is not in my word: _ ee_
```

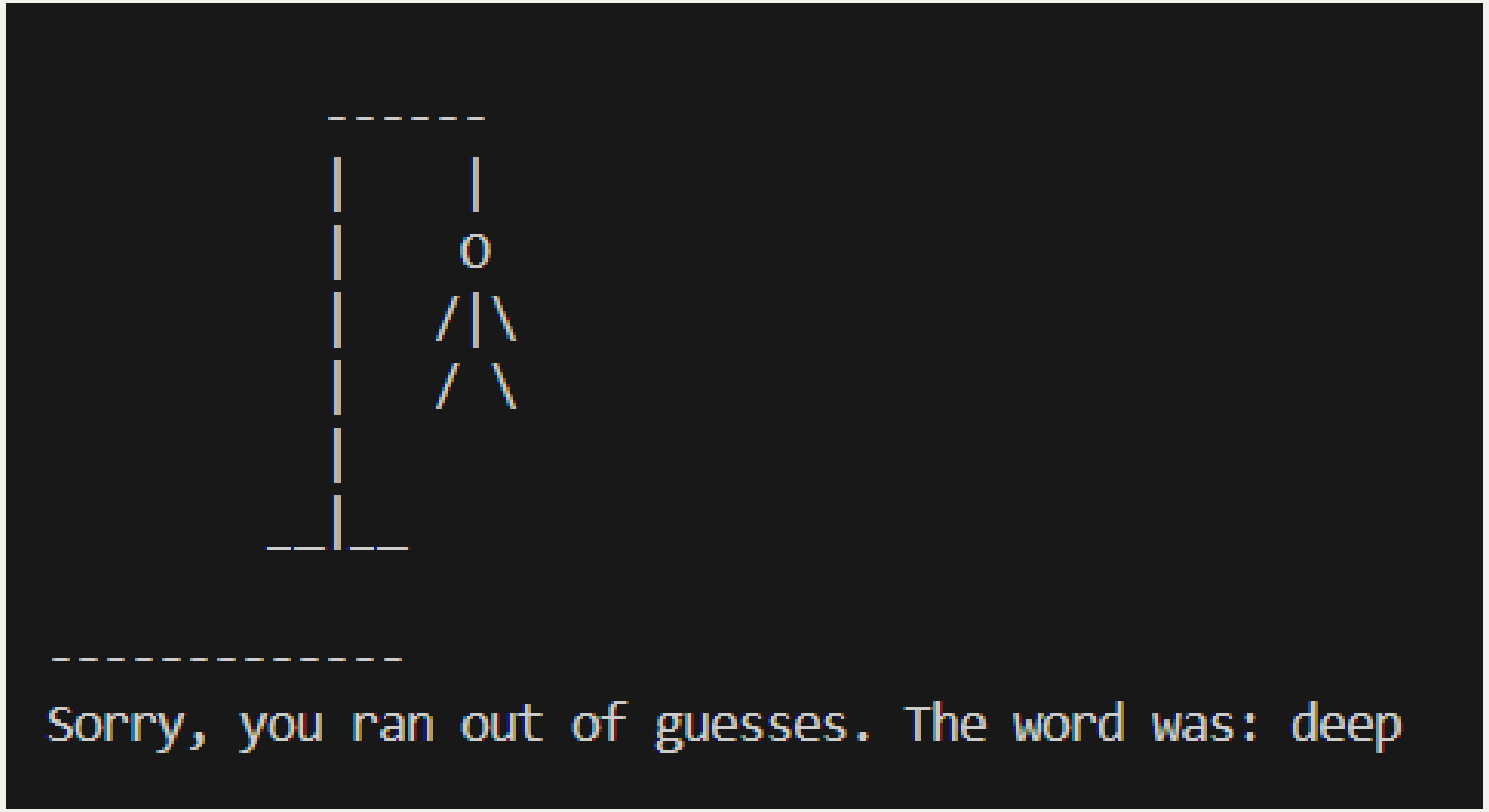


```
          ------
          |    |
          |    O
          |    |
          |
          |
        __|__

-------------
You have 4 guesses left.
Available letters: bcdfghjklmnopqrstuvwxyz
Please guess a letter: o
Oops! That letter is not in my word: _ ee_

          ------
          |    |
          |    O
          |   /|
          |
          |
        __|__

-------------
You have 3 guesses left.
Available letters: bcdfghjklmnpqrstuvwxyz
Please guess a letter: u
Oops! That letter is not in my word: _ ee_
```



```
          ------
          |    |
          |    O
          |   /|\
          |
          |
        __|__

-------------
You have 2 guesses left.
Available letters: bcdfghjklmnpqrstvwxyz
Please guess a letter: x
Oops! That letter is not in my word: _ ee_

          ------
          |    |
          |    O
          |   /|\
          |   /
          |
        __|__

-------------
You have 1 guesses left.
Available letters: bcdfghjklmnpqrstvwyz
Please guess a letter: y
Oops! That letter is not in my word: _ ee_
```

OUTCOMES



```
        ------
        |    |
        |    O
        |   /|\
        |   / \
        |
      __|__

-------------
Sorry, you ran out of guesses. The word was: deep
```

THANK YOU