****

**Task-1**

**Eman Iqbal**

**SU92-BSAIM-F24-111**

**Section: BsAi-3B**

**Hangman Game**

**Overview:**

This is a Python-based Hangman game.  
The player tries to guess a secret word, one letter at a time. Each wrong guess adds a part to the hangman figure. The goal is to reveal the entire word before the hangman is fully drawn.

The game runs continuously, letting the player guess letters until either all letters are revealed or all attempts are used.

**How It Works.**

**Choosing the Secret Word:**

The game randomly selects a word from a predefined list. Words include fun names and terms, such as “Orange”, “cherry”, and “kiwi”. The chosen word remains hidden from the player.

**Displaying Blanks:**

Each letter in the secret word is initially replaced with an underscore \_.  
The player sees only these blanks, indicating how many letters they need to guess.

**Player Input:**

The game asks the player to enter a single letter as a guess.  
Repeated guesses are detected and the player is prompted to try a new letter.

**Checking the Guess:**

* **Correct Guess:** If the guessed letter is in the secret word, all occurrences are revealed in the displayed blanks.
* **Wrong Guess:** If the letter is not in the word, the player loses one turn. Each lost turn adds a body part to the hangman drawing.

**Drawing the Hangman:**

The hangman is drawn progressively for each incorrect guess:

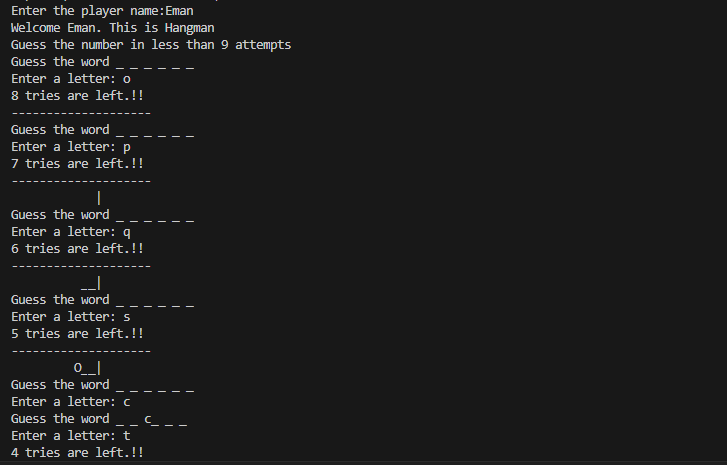
1. Head
2. Body
3. Arms
4. Legs

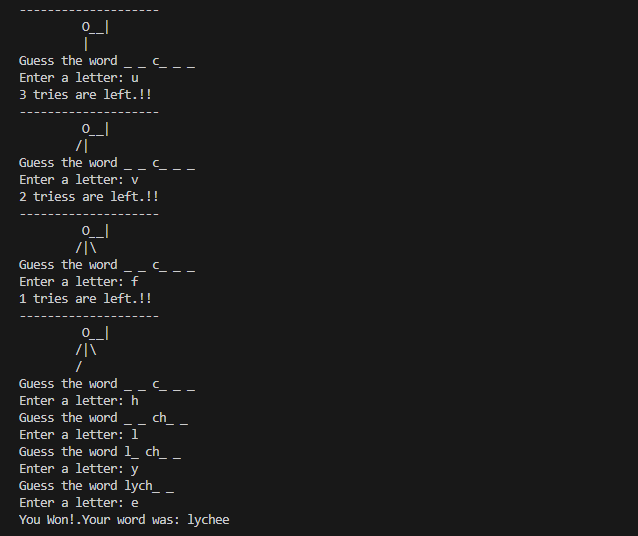
When all turns are used, the hangman is complete, and the game ends.

**Winning and Losing:**

* **Win:** If the player guesses all letters before running out of turns, the game congratulates the player.
* **Lose:** If all turns are exhausted, the hangman is fully drawn, the game ends, and the secret word is revealed.

**Example Run:**

****

****

**Secret word: lychee**

* **Player guesses c → Display: \_ \_ c \_ \_ \_**
* **Player guesses h → Display: \_ \_ c h\_ \_**
* **Player guesses l → Display: l \_ c h\_ \_**
* **Player guesses y → Display: l y c h \_ \_**
* **Player guesses e → Display: l y c h e e→ Player won**

If the player guesses wrong letters instead, the hangman figure is drawn progressively with each wrong attempt until all turns are used and the game ends with a loss.

**Features:**

* Reveals correct letters dynamically
* Detects repeated guesses
* Shows progressive hangman drawing for wrong guesses
* Game ends with a win or loss message
* Randomly selects words from a predefined list