***Hangman(Dynamic)***

The Hangman Game is a Python-based word guessing game. A random word is chosen, and the player must guess it letter by letter within limited attempts. Wrong guesses reduce chances and draw parts of the hangman, while correct guesses reveal letters. The player wins by completing the word before the hangman is fully drawn.

**Code explaination:**

1**. import :** is used to bring external Python modules or libraries into the program so that their built-in functions can be used.

2. **random.choice(list) :** selects one random element from a given list, which is useful for creating unpredictability in the game.

3. **len()** :returns the total number of items in a string, list, or iterable, helping to determine the word length.

4. **print()** :displays messages, variables, or outputs on the screen for the user to see.

5. **input()** :takes input directly from the user through the keyboard, always returning the value as a string.

6. **lower() :** converts an entire string into lowercase letters, ensuring consistency and avoiding input mismatches.

7. **" ".join(list) :** combines elements of a list into a single string with spaces in between, making the word display more readable.

8. **enumerate(iterable) :** allows iteration over a sequence while providing both the index and the value at the same time, which is useful for checking and updating letters in the word

Code logic:

1. **Word Selection**

* The program begins by choosing a random word from a predefined list using the random.choice() function.
* This ensures that every new game is different and unpredictable.

2. **Initialization**

* A list of underscores (\_) is created, equal to the length of the chosen word, representing hidden letters.
* A fixed number of attempts (6 in this case) is set, which represents how many wrong guesses the player can make before losing.

3. **Gameplay Loop**

* The game enters a loop that continues until either the player guesses the word completely or runs out of attempts.
* In each iteration, the current state of the word (with guessed and hidden letters) and the corresponding hangman stage are displayed.

4. **User Input & Checking**

* The player inputs a single letter guess.
* If the letter exists in the chosen word, the program updates the hidden word by replacing the corresponding underscores with the guessed letter.
* If the letter does not exist, the attempts counter decreases by one, and the hangman drawing progresses.

5. **Updating Progress**

* Correct guesses bring the player closer to completing the word.
* Wrong guesses bring the hangman closer to being fully drawn, visually showing the remaining chances.

6. **Game End Conditions**

* **Win Condition:** If there are no underscores left (meaning the player guessed all letters), the game declares victory.
* **Lose Condition:** If attempts reach zero, the game ends with the hangman fully drawn, and the correct word is revealed.

