



Overview of the program :

This program is a game named Hangman implemented in the programming language C. It is both challenging and fun.

Objective of the Game:
Hangman is a classic word-guessing game that challenges players to test their vocabulary and deduction skills. The objective is simple: guess the hidden word, one letter at a time, within a limited number of attempts. Each incorrect guess brings the hangman closer to completion, so think carefully and choose wisely.

Conclusion of the Code:
Our Hangman game demonstrates the application of fundamental programming concepts such as loops, logical operators ,functions, and arrays in C. By combining logic with creativity, we have created an interactive and engaging game. This project showcases our ability to design a functional program, handle user input, and implement error handling effectively

- Hangman Game Algorithm:**
1. Set Up the Game:
 - Pick a word for the player to guess, e.g., "PROGRAM".
 - Figure out the length of the word.
 - Create an array (guessed[]) to track which letters have been guessed (initially set to all “not guessed”).
 - Start with zero wrong attempts.
 2. Show a Welcome Message:
 - Display a message introducing the game and how many attempts the player is allowed.
 3. Play the Game:
 - Repeat the following steps until the player guesses the word or runs out of attempts:
 - a. Show the Current Progress:
 - Display the word with the correctly guessed letters revealed, and the rest as underscores (e.g., _ R _ G R A _).
 - b. Ask for a Letter:
 - Prompt the player to guess a letter.
 - Convert their guess to uppercase to match the word format.
 - c. Check the Guess:
 - Go through the word to see if the guessed letter is there:
 - If yes: Reveal the letter(s) in the word and mark it as guessed.
 - If no: Add 1 to the number of wrong attempts and tell the player how many attempts they have left.
 - d. Check if the Player Won:
 - If the entire word is guessed correctly, congratulate the player and end the game.
 4. End the Game:
 - If the player runs out of attempts, reveal the correct word and show a game-over message.
 5. Exit:
 - End the program.

- Helper Functions:**
- displayWord():
Shows the word with guessed letters filled in and underscores for missing letters.
 - isWordGuessed():
Checks if the entire word has been guessed. If yes, returns true to indicate the player has won.

OUTPUT OF CODE

Output

```
Welcome to Hangman!
You have 8 attempts to guess the word.

Word: _ _ _ _ _

Guess a letter: u

Incorrect! You have 7 attempts left.
Word: _ _ _ _ _

Guess a letter: |
```

```
Guess a letter: o

Good guess!
Word: _ _ 0 _ _ _

Guess a letter: |
```

```
Guess a letter: r

Good guess!
Word: P R 0 G R _ M

Guess a letter:
```

```
Guess a letter: a

Good guess!

Congratulations! You guessed the word: PROGRAM

=== Code Execution Successful ===
```

```
Guess a letter: i

Incorrect! You have 2 attempts left.
Word: _ _ 0 _ _ _

Guess a letter: n

Incorrect! You have 1 attempts left.
Word: _ _ 0 _ _ _

Guess a letter: x

Incorrect! You have 0 attempts left.

You've run out of attempts! The word was: PROGRAM
Better luck next time!

=== Code Execution Successful ===|
```

NAME: MARIYA RAVAT
ER.NO: 12402130501033
BRANCH: CSD
TERM:ODD
SEMESTER: 1

NAME: JIYA VAGHANI
ER.NO: 12402130501024
BRANCH: CSD
TERM:ODD
SEMESTER: 1