Mini Project -1 (Number Guessing Game) (Language PYTHON)

1. Problem Analysis and statement:

Objective:

Our goal is to create an interactive and educational Python-based number guessing game that provides a fun experience for users. The game will include dynamic hinting and leaderboard features to keep players engaged.

Features:

Difficulty Levels:

- **Easy:** The target number will be within the range of 1 to 10.
- **Medium:** The target number will be within the range of 1 to 50.
- **Hard:** The target number will be within the range of 1 to 100.

Hints:

The game will provide different types of hints to assist players in guessing the target number:

- **Range Hint:** This hint will let players know if the target number is higher or lower than their guess.
- **Proximity Hint:** Players will receive feedback on how close their guess is to the target number, whether it's very close, close, or far off.
- **Even/Odd Hint:** This hint will indicate whether the target number is even or odd.
- **Divisibility Hint:** Every third guess, the game will provide a hint indicating whether the target number is divisible by certain selected numbers.

Leaderboard:

The game will keep track of players' names and their attempts. After each game, the leaderboard will display the scores in ascending order of attempts.

2.Algorithm:

- **Initialization:**

  - We will start by initializing an empty leaderboard list.

- **Print Leaderboard Function:**

  - The game will check if the leaderboard is empty. If there are no scores, it will inform the players.

  - The leaderboard entries will be sorted and displayed based on the number of attempts.

- **Get Difficulty Range Function:**

  - The game will display the difficulty options and ask the player to select one.

  - Based on the player's choice, the corresponding range will be returned.

- **Give Hint Function:**

  - This function will provide hints to the players based on their guesses.

  - Hints will include information about the relationship of the guess to the target number, proximity of the guess to the target number (very close, close, or far off), even/odd nature of the target number, and divisibility of the target number (every third guess).

- **Number Guessing Game Function:**

  - The game will start by displaying a welcoming message.

  - The player will be asked to select the difficulty range.

  - A random number within the specified range will be generated.

- The game will loop until the correct guess is made.
- The player's guess will be prompted and validated.
- Appropriate hints will be provided based on the guess.
- The number of attempts will be tracked.
- Once the correct guess is made, the player's score will be recorded.
- The updated leaderboard will be displayed.
- The player will be asked if they want to play again.

3.pseudo code:

```
Initialize an empty list called leaderboard

Function print_leaderboard():
    If leaderboard is empty:
        Print "No scores yet. Be the first to play!"
    Else:
        Sort leaderboard by 'attempts' in ascending order
        Print "Leaderboard:"
        For each score in sorted leaderboard:
            Print the position (index + 1), player's name, and number of attempts

Function get_difficulty_range():
    Print "Select Difficulty Level:"
    Print "1. Easy (1-10)"
    Print "2. Medium (1-50)"
    Print "3. Hard (1-100)"
    Prompt user for difficulty choice and store it in 'difficulty'
    If difficulty is 1:
        Return range (1, 10)
    Else if difficulty is 2:
        Return range (1, 50)
    Else if difficulty is 3:
        Return range (1, 100)
    Else:
        Print "Invalid choice. Defaulting to Easy level."
        Return range (1, 10)

Function give_hint(number_to_guess, guess, guess_count):
    Initialize an empty list called 'hints'

    # Range hint
    If guess is less than number_to_guess:
        Append "The number is greater than your guess." to 'hints'
    Else:
        Append "The number is less than your guess." to 'hints'

    # Proximity hint
    Calculate difference as absolute value of (number_to_guess - guess)
    If difference is 0:
        Return "Correct!"
    Else if difference is less than or equal to 3:
        Append "Very close!" to 'hints'
    Else if difference is less than or equal to 10:
        Append "Close!" to 'hints'
    Else:
        Append "Far off!" to 'hints'

    # Even/Odd hint
    If guess_count is even:
        If number_to_guess is even:
            Append "The number is even." to 'hints'
        Else:
            Append "The number is odd." to 'hints'

    # Divisibility hint
    If guess_count is divisible by 3:
        For each i in [3, 5, 7]:
            If number_to_guess is divisible by i:
                Append "The number is divisible by i." to 'hints'
                Break out of the loop

    Return concatenated 'hints' as a string with spaces between hints
```

```
Function number_guessing_game():
    Print "Welcome to the Number Guessing Game!"
     X, Y = get_difficulty_range()
    Randomly generate number_to_guess between X and Y
    Initialize guess_count to 0
    Print "Guess the number between X and Y"

    Loop forever:
        Increment guess_count by 1
        Prompt user to enter a guess and store it in 'guess'

        If guess is less than X or greater than Y:
            Print "Please enter a number within the range X and Y."
            Continue to the start of the loop

        Call give_hint function with arguments number_to_guess, guess, and guess_count
        Store the returned hint in 'hint'

        If "Correct!" is found in hint:
            Print hint
            Prompt user for their name and store it in 'name'
            Append {'name': name, 'attempts': guess_count} to leaderboard
            Print "You've guessed the number in guess_count attempts."
            Break out of the loop

        Else:
            Print hint

    Call print_leaderboard function to display leaderboard

    Prompt user "Do you want to play again? (yes/no):" and store the response in 'replay'
    If 'replay' is "yes":
        Restart the game by calling number_guessing_game function
    Else:
        Print "Thank you for playing!"
```

4.Analysis:

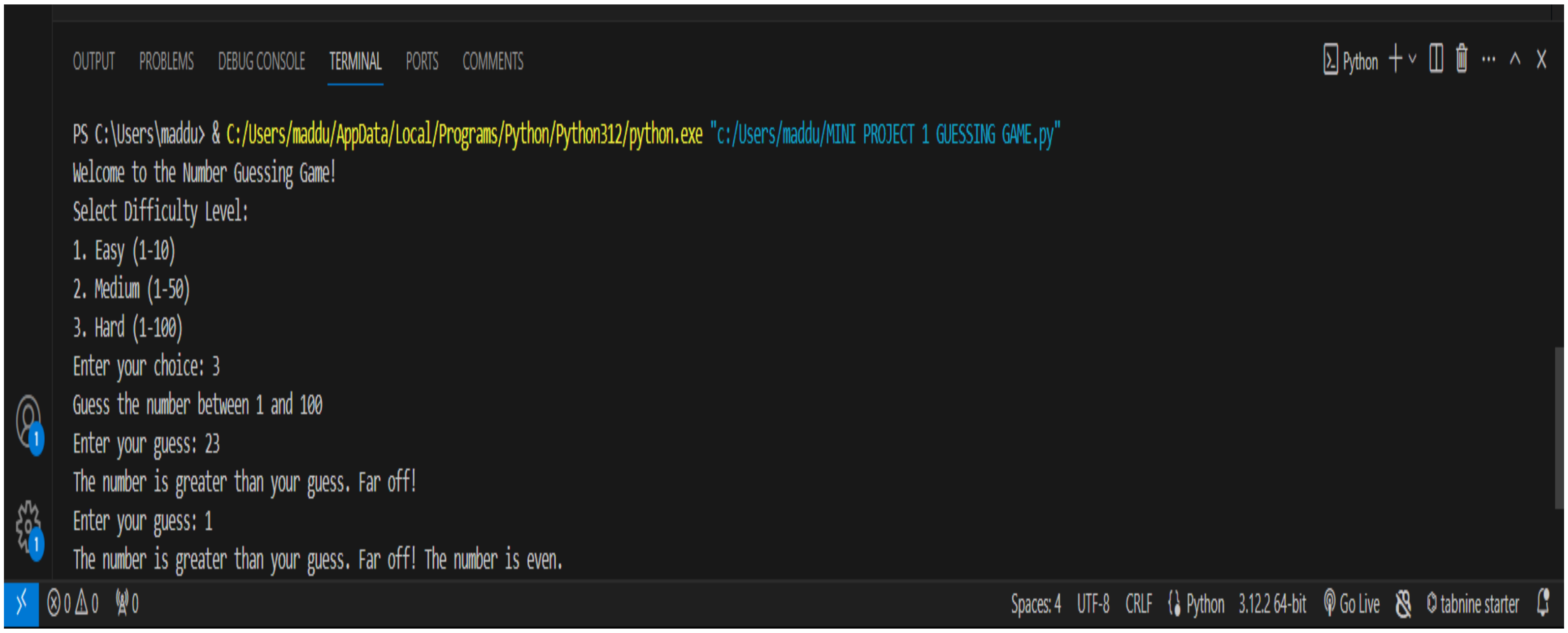
- **User Engagement:** Our game aims to engage players by providing an interactive and educational experience that enhances logical thinking and number awareness.
- **Hints Enhance Gameplay:** The dynamic hinting system will guide players and provide feedback, helping them deduce the target number more effectively.
- **Competitive Element:** The leaderboard fosters competition among players and motivates them to achieve the lowest number of attempts.
- **Scalability:** Our game is designed to be adaptable for future updates, such as adding more difficulty levels or multiplayer features.

5.Source Code:

```
import random

leaderboard = []

def print_leaderboard():
    if not leaderboard:
        print("No scores yet. Be the first to play!")
        return
    print("Leaderboard:")
    for idx, score in enumerate(sorted(leaderboard, key=lambda x: x['attempts'])):
        print(f"{idx + 1}. {score['name']} - {score['attempts']} attempts")

def get_difficulty_range():
    print("Select Difficulty Level:")
    print("1. Easy (1-10)")
    print("2. Medium (1-50)")
    print("3. Hard (1-100)")
    difficulty = int(input("Enter your choice: "))
    if difficulty == 1:
        return 1, 10
    elif difficulty == 2:
        return 1, 50
    elif difficulty == 3:
        return 1, 100
    else:
        print("Invalid choice. Defaulting to Easy level.")
        return 1, 10

def give_hint(number_to_guess, guess, guess_count):
    hints = []

    # Range hint
    if guess < number_to_guess:
        hints.append("The number is greater than your guess.")
    else:
        hints.append("The number is less than your guess.")

    # Proximity hint
    difference = abs(number_to_guess - guess)
    if difference == 0:
        return "Correct!"
    elif difference <= 3:
        hints.append("Very close!")
    elif difference <= 10:
        hints.append("Close!")
    else:
        hints.append("Far off!")

    # Even/Odd hint
    if guess_count % 2 == 0:   # Provide even/odd hint on every alternate
guess
        if number_to_guess % 2 == 0:
            hints.append("The number is even.")
        else:
            hints.append("The number is odd.")

    # Divisibility hint
    if guess_count % 3 == 0:  # Provide divisibility hint every third guess
        for i in [3, 5, 7]:  # Choose a few numbers to check divisibility
```

```
            if number_to_guess % i == 0:
                hints.append(f"The number is divisible by {i}.")
                break

    return " ".join(hints)

def number_guessing_game():
    print("Welcome to the Number Guessing Game!")
    X, Y = get_difficulty_range()

    number_to_guess = random.randint(X, Y)
    guess_count = 0
    print(f"Guess the number between {X} and {Y}")

    while True:
        guess_count += 1
        guess = int(input("Enter your guess: "))

        if guess < X or guess > Y:
            print(f"Please enter a number within the range {X} and {Y}.")
            continue

        hint = give_hint(number_to_guess, guess, guess_count)

        if "Correct!" in hint:
            print(hint)
            name = input("Congratulations! You've guessed the number. What's your name? ")
            leaderboard.append({'name': name, 'attempts': guess_count})
            print(f"You've guessed the number in {guess_count} attempts.")
            break
        else:
            print(hint)

    print_leaderboard()

    replay = input("Do you want to play again? (yes/no): ").strip().lower()
    if replay == "yes":
        number_guessing_game()
    else:
        print("Thank you for playing!")

if __name__ == "__main__":
    number_guessing_game()
```

6.Output:



```
PS C:\Users\maddu> & C:/Users/maddu/AppData/Local/Programs/Python/Python312/python.exe "c:/Users/maddu/MINI PROJECT 1 GUESSING GAME.py"
Welcome to the Number Guessing Game!
Select Difficulty Level:
1. Easy (1-10)
2. Medium (1-50)
3. Hard (1-100)
Enter your choice: 3
Guess the number between 1 and 100
Enter your guess: 23
The number is greater than your guess. Far off!
Enter your guess: 1
The number is greater than your guess. Far off! The number is even.
```