S.No: 1	Exp. Name: ***Project Module***	Date:2024-05-12

Aim:

Project Module.

Source Code:

CTP28132.py

```
import random
import sys

def initialize_game():
    print("Welcome to the Guessing a Number Game!")
    difficulty = input("Choose difficulty Level (easy, medium, hard): ").lower
()

    if difficulty == 'easy':
        print("Guessing between 1 to 10")
        return 1, 10, 5
    elif difficulty == 'medium':
        print("Guessing between 1 to 100")
        return 1, 100, 7
    elif difficulty == 'hard':
        print("Guessing between 1 to 1000")
        return 1, 1000, 10
    else:
        print("Invalid difficulty choice. Exiting game.")
        sys.exit()

def generate_random_number(low, high):
    return random.randint(low, high)

def get_player_guess():
    while True:
        try:
            guess = int(input("Enter your guess: "))
            return guess
        except ValueError:
            print("Invalid input. Please enter a valid number.")

def provide_feedback(guess, number):
    if guess < number:
        print("Your Guessing Is Too low!")
    elif guess > number:
        print("Your Guessing Is Too high!")
    else:
        print("Congratulations! You guessed it!")

def track_attempts(attempts, max_attempts):
    if attempts >= max_attempts:
        return True
    return False

def provide_hint(number):
```

```
    if number % 2 == 0:
        print("Hint: The number is even.")
    else:
        print("Hint: The number is odd.")

    # Additional hint: providing a narrower range
    range_hint = 5  # Adjust this value for a smaller or larger range hint
    low_hint = max(1, number - range_hint)
    high_hint = min(100, number + range_hint)
    print(f"Hint: The number is between {low_hint} and {high_hint}.")

def play_game():
    low, high, max_attempts = initialize_game()
    number = generate_random_number(low, high)
    attempts = 0

    while attempts < max_attempts:
        guess = get_player_guess()
        attempts += 1

        if guess == number:
            provide_feedback(guess, number)
            break
        else:
            provide_feedback(guess, number)

            # Provide a hint after half the attempts
            if attempts == max_attempts // 2:
                provide_hint(number)

        if track_attempts(attempts, max_attempts):
            print(f"Sorry, you've used all your attempts. The number was {numb
er}.")
            break

    play_again = input("Do you want to play again? (yes/no): ").lower()
    if play_again == 'yes':
        play_game()
    else:
        print("Thanks for playing!")

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

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Hello World
Hello World