**PROJECT TITLE**: AI COWS AND BULLS MASTER

**DESCRIPTION:**

The Cows and Bulls Game is a classic code-breaking game designed for two players: Player A and an AI bot. Player A starts by selecting a 3-digit secret number through a dialog box, while the AI bot generates its own 3-digit number randomly. The objective of the game is for each player to guess the opponent's secret number.

The players take turns guessing the 3-digit number. After each guess, the guessing player receives feedback in terms of "cows" and "bulls." "Cows" indicate the number of correct digits that are in the wrong positions, while "bulls" represent the number of correct digits in the correct positions. For example, if the secret number is "123" and the guess is "321," the feedback would be 3 cows and 0 bulls.

Player A begins by entering a guess for the AI bot's number. If the guess is incorrect, the AI bot then makes a guess, which is randomly selected from possible valid numbers. The AI bot uses the feedback of cows and bulls to narrow down its future guesses. This alternation continues until one of the players correctly guesses the opponent's secret number, achieving three bulls. The game is engaging as it challenges players' logical thinking and deductive reasoning skills to decipher the opponent's number using the feedback provided.

**KEY FEATURES:**

1. **get\_cows\_and\_bulls(secret\_number, guess)** function calculates feedback by comparing the guess to the secret number, counting cows (correct digits in the wrong position) and bulls (correct digits in the correct position).
2. **get\_secret\_number(player)** function prompts Player A to input a 3-digit secret number using a Tkinter dialog box, ensuring the input is valid.
3. **generate\_ai\_secret\_number()** function generates a random 3-digit number for the AI bot using Python's random module.
4. **play\_game()** function manages the game loop. It starts with Player A inputting their secret number and the AI bot generating its number. The function alternates turns between Player A and the AI bot, providing feedback on guesses until one correctly guesses the opponent's number, achieving three bulls.

**FUTURE IMPROVEMENTS:**

* 1. **Enhanced AI:** Implement a smarter AI bot with advanced guessing strategies, improving challenge and engagement by making more intelligent guesses based on previous feedback.
  2. **User Interface Upgrade:** Develop a graphical user interface (GUI) for a more interactive and visually appealing experience, including features like displaying past guesses and feedback.
  3. **Multiplayer Mode:** Add a multiplayer mode for local or online play, allowing two human players to compete, increasing the game's appeal and social interaction.
  4. **Mobile Version:** Create a mobile app version to make the game accessible on the go, with touch-friendly controls and features optimized for smartphones and tablets.

**PROGRAM:**

import random

import tkinter as tk

from tkinter import simpledialog

def get\_cows\_and\_bulls(secret\_number, guess):

cows = 0

bulls = 0

secret\_number = str(secret\_number)

guess = str(guess)

secret\_digits\_count = {}

guess\_digits\_count = {}

# Count occurrences of each digit in secret\_number and guess

for digit in secret\_number:

if digit in secret\_digits\_count:

secret\_digits\_count[digit] += 1

else:

secret\_digits\_count[digit] = 1

for digit in guess:

if digit in guess\_digits\_count:

guess\_digits\_count[digit] += 1

else:

guess\_digits\_count[digit] = 1

# Calculate bulls and cows

for i in range(len(secret\_number)):

if guess[i] == secret\_number[i]:

bulls += 1

for digit in guess\_digits\_count:

if digit in secret\_digits\_count:

cows += min(secret\_digits\_count[digit], guess\_digits\_count[digit])

cows -= bulls # Adjust cows to exclude digits counted as bulls

return cows, bulls

def get\_secret\_number(player):

root = tk.Tk()

root.withdraw()

while True:

number = simpledialog.askstring(player, f"{player}, enter your 3-digit secret number:", show='\*')

if number and number.isdigit() and len(number) == 3:

root.destroy()

return number

else:

tk.messagebox.showerror("Invalid input", "Please enter a 3-digit number.")

def generate\_ai\_secret\_number():

return random.randint(100, 999)

def play\_game():

print("Welcome to the Cows and Bulls Game!")

# Player A chooses a number

player\_a\_number = get\_secret\_number("Player A")

# AI\_Bot generates a number

ai\_bot\_number = generate\_ai\_secret\_number()

possible\_numbers\_a = [str(i).zfill(3) for i in range(1000)]

current\_player = 'A'

print(f"\nPlayer A will start guessing first.\n")

while True:

if current\_player == 'A':

guess = input("Player A, enter your guess: ")

if guess.isdigit() and len(guess) == 3:

cows, bulls = get\_cows\_and\_bulls(ai\_bot\_number, guess)

print(f"Cows: {cows}, Bulls: {bulls}")

if bulls == 3:

print("Player A has guessed the number correctly!")

break

current\_player = 'AI\_Bot'

else:

print("Invalid guess. Please enter a 3-digit number.")

elif current\_player == 'AI\_Bot':

guess = random.choice(possible\_numbers\_a)

print(f"AI\_Bot guesses: {guess}")

cows, bulls = get\_cows\_and\_bulls(player\_a\_number, guess)

print(f"Cows: {cows}, Bulls: {bulls}")

if bulls == 3:

print("AI\_Bot has guessed the number correctly!")

break

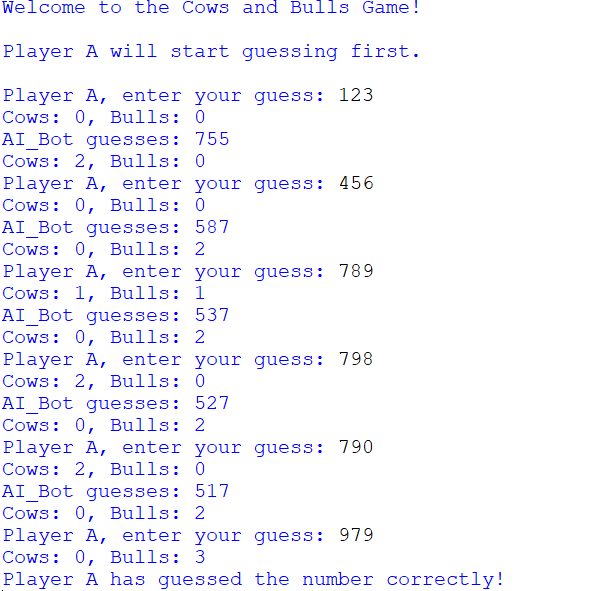
possible\_numbers\_a = [num for num in possible\_numbers\_a if get\_cows\_and\_bulls(num, guess) == (cows, bulls)]

current\_player = 'A'

if \_\_name\_\_ == "\_\_main\_\_":

play\_game()

**OUTPUT:**



=== Code Execution Successful ===