

```
In [ ]: from random import randint
```

```
def secret_generator():
    """Generates a 4 digits secret code consisting of unique numbers

    Return:
        -- A String representing the 4-digits secret code with unique digits.
    """
    digits_set = set()
    while len(digits_set) < 4:
        digits_set.add(randint(0, 9))

    secret_code = ""

    for i in digits_set:
        secret_code += str(i)

    return secret_code

def count_bulls(secret_code, user_guess):
    """Checks the user's guess against the secret code and calculates the number

    Bulls: Digits in the user's guess that match both the value and position in

    Arguments:
        secret_code -- String with 4-digits generate by the secret_generator() funct
        user_guess -- String representin the user's guess.

    Return values: int bulls representing the numbers of Bulls.
    """
    bulls = 0
    for i in range(0, len(user_guess)):
        for j in range(0, len(secret_code)):
            if user_guess[i] == secret_code[j] and i==j:
                bulls += 1
    return bulls

def count_cows(secret_code, user_guess):
    """Checks the user's guess against the secret code and calculates the number

    Cows: Digits in the user's guess that exist in the string secret_code but an

    Arguments:
        secret_code -- String with 4-digits generate by the secret_generator() funct
        user_guess -- String representin the user's guess.

    Return values: int cows representing the numbers of Cows.
    """
    cows = 0
    for i in range(0, len(user_guess)):
        for j in range(0, len(secret_code)):
            if user_guess[i] == secret_code[j] and i != j:
                cows += 1

    return cows
```

```

def check_bulls_and_cows(bulls, cows, secret_code):
    """Return the grade"""
    is_secret_code_revealed = False

    if bulls == 4:
        str = f"Grade: {bulls} bulls.\nCongrats! The secret code is {secret_code}"
        is_secret_code_revealed = True
        return (str, is_secret_code_revealed)
    elif bulls == 1 and cows == 0:
        str = f"Grade: {bulls} bull."
        return (str, is_secret_code_revealed)
    elif bulls == 0 and cows == 1:
        str = f"Grade: {cows} cow."
        return (str, is_secret_code_revealed)
    elif bulls == 0 and cows == 0:
        str = "Grade: None."
        return (str, is_secret_code_revealed)
    elif bulls > cows:
        str = f"Grade: {bulls} bulls."
        return (str, is_secret_code_revealed)
    elif bulls == cows:
        str = f"grade: {bulls} bulls and {cows} cow."
        return (str, is_secret_code_revealed)
    else:
        str = f"Grade: {cows} cows."
        return (str, is_secret_code_revealed)

def start_game():
    attempts = 1
    secret_code = secret_generator()
    print("The secret code is prepared: ****.")

    is_secret_code_revealed = False
    while is_secret_code_revealed == False:
        user_guess = input(f"Turn {attempts}. Answer:\n")
        bulls = count_bulls(secret_code, user_guess)
        cows = count_cows(secret_code, user_guess)
        print(check_bulls_and_cows(bulls, cows, secret_code)[0])
        is_secret_code_revealed = check_bulls_and_cows(bulls, cows, secret_code)
        attempts += 1

def main():
    start_game()

if __name__ == "__main__":
    main()

```

In []: