

## Homework B: Bulls and Cows game

### 目的：

本作業除了寫作 'Bulls and Cows' 遊戲的程式外，也同時學習如何有效地與 AI 輔助編程工具 (如 Copilot) 協作。

### 提示：

1. 分解問題：將複雜的遊戲邏輯拆解為獨立的函數。
2. 精準提問：透過清晰的函數簽名和註釋，引導 AI 工具生成正確的程式碼。
3. 驗證與調試：不盲目接受 AI 建議，而是驗證其正確性並進行必要的修改。
4. Python 基礎：熟悉條件判斷、迴圈、函數設計和字串/列表操作。

### 遊戲規則：

1. 程式隨機生成一個四位數且數字不重複的秘密數字。
2. 玩家猜測，程式給予 A (Bulls) 和 B (Cows) 提示。  
A: 數字和位置都正確的個數。  
B: 數字正確但位置不正確的個數。  
例一，當秘密數字為 8123，而猜謎者猜 1052 時，出題者必須提示 0A2B。  
例二，當秘密數字為 5637，而猜謎者猜 4931 時，出題者必須提示 1A0B。

### 實作步驟

請創建一個名為 PY\_HW\_B.py 或 PY\_HW\_B.ipynb 的 Python 程式。  
在每個步驟中，請先寫下函數簽名和註釋，然後觀察 AI 工具的建議，並決定是否接受、修改或自行編寫。

#### 步驟一：生成秘密數字 (Generate Secret Number)

這個函數負責遊戲的初始化。請在程式中輸入以下內容：

```
import random
```

```
def generate_secret_number():
```

```
    """
```

```
    隨機生成一個四位數且數字不重複的秘密數字字串。
```

```
    Copilot 提示：
```

1. 思考如何從 '0' 到 '9' 中選出四個不重複的數字。
2. 嘗試使用 random.sample() 或先打亂列表再取前四個。
3. 確保函數返回的是一個四個字元的字串。

```
    """
```

```
    # 在這裡等待 AI 工具的建議，或自行編寫程式碼
```

```
pass
```

## 步驟二：檢查輸入有效性 (Validate Guess)

在遊戲開始前，我們需要確保玩家的輸入是合法的。請在程式中新增以下內容

```
def is_valid_guess(guess):
```

```
    """
```

檢查玩家輸入是否為有效的四位數猜測。有效的猜測必須滿足：

1. 是一個四個字元的字串。
2. 每個字元都是數字。
3. 四個數字必須不重複。

Copilot 提示：

1. 使用 `len()` 檢查長度。
2. 使用 `str.isdigit()` 檢查是否為數字。
3. 使用 `set()` 檢查是否有重複數字 (`len(set(guess)) == 4`)。

```
    """
```

# 在這裡等待 AI 工具的建議，或自行編寫程式碼

```
pass
```

## 步驟三：計算提示 (Calculate Hint)

這是遊戲的核心邏輯，也是最複雜的部分。請在程式中新增以下內容

```
def calculate_hint(secret, guess):
```

```
    """
```

計算 Bulls (A) 和 Cows (B) 的數量。

Bulls: 數字和位置都正確。

Cows: 數字正確但位置不正確。

參數:

secret (str): 秘密數字，四位數。

guess (str): 玩家猜測，四位數。

返回:

tuple: (bulls, cows)

Copilot 提示：

請自行輸入給 AI 工具的提示文句

```
    """
```

# 在這裡等待 Copilot 的建議，或自行編寫程式碼

```
pass
```

#### 步驟四：主遊戲迴圈 (Main Game Loop)

將所有函數整合起來，實現完整的遊戲流程。請在您的程式中新增以下函數

```
def main():
```

```
    """
```

遊戲主函數，負責流程控制、輸入輸出和結果顯示。

Copilot 提示：

請自行輸入給 AI 工具的提示文句

```
    """
```

# 在這裡等待 Copilot 的建議，或自行編寫程式碼

```
    pass
```

```
if __name__ == "__main__":
```

```
    main()
```

## Deliverables

### 1. 評分標準

- A. 完成程式碼 (50%): 遊戲程式功能完整，遊戲過程確實呈現。
- B. 提交協作報告(50%): 提交一份簡短報告，回答以下問題。
  - 最成功的協作：您認為哪個函數的編寫過程中，Copilot 的幫助最大？您是如何提問或寫註釋來引導它的？
  - 遇到的問題：您是否遇到 Copilot 提供的程式碼有錯誤或不符合要求的情況？您是如何調試和修正的？
  - 反思：你認為 AI 輔助編程工具在學習 Python 基礎時，是幫助還是阻礙？為什麼？

- 2. Zip 程式碼 and 協作報告, then upload it to E3 with file name  
YourStudentID\_YourName\_PY\_HW\_B.zip