|  |
| --- |
| Python程式設計與實務應用\_ZZZ001 |
| 應用程式開發說明書 |
| 112121931\_林智鴻 |

|  |
| --- |
| 112121931\_林智鴻  2024/10/4 |

# 主題名稱

由電腦進行1A2B猜數字程式（非AI，以邏輯進行答案的推理）。

目標，平均以4~7次的猜測得到答案。

小時候很喜歡和人玩這個遊戲，學會了寫程式之後，寫一個出題的程式來玩似乎不夠有趣，AI就能寫出來了。所以試著整理自己的思考邏輯，以Python來撰寫一個程式來猜測我所出的題目。

使用套件的成果雖然酷炫，但那實在都是別人寫的好程式，無法充分的運用課堂所學，所以純以自己進行程式撰寫，讓自己能更熟悉語法，期末作業再與小組成員多一些套件的整合與應用的實作。

# 所開發的應用程式功能說明

1. 模式一，可由使用者進行出題，4位不重複數字，由電腦進行猜測，使用者依猜測結果回覆A、B的數量，由電腦反覆猜測至4A為止。
2. 模式二，由電腦自動出題，與猜測、回答驗證結果等動作，反覆猜測至4A為止。
3. 猜測依每次的猜測回覆結果，進行推理與去除不合理的選項。
4. 每輪猜測，以十次為限，避免人工輸入錯誤時而造成的無解。

# 開發環境

|  |  |
| --- | --- |
| 工具與版本 | 說明 |
| Python 3.12.5 | 使用程式語言與版本 |
| Visual Studio Code 1.93.1 | 主要開發IDE工具，與版本簽入GitHub。 |
| GitHub | 版本控制軟體，並於每次Commit時自動透過Pylint進行檢核。  [GitHub 存放路徑](https://github.com/112121931/PY/tree/main/Source/midtermwork) (未開放) |
| Pylint 3.3.1 | 確保程式開發風格符合PEP8標準。  無使用其它套件。 |

# 程式碼撰寫技巧說明

（一）、由使用者輸入一個四位的數字, 若未輸入, 由程式自動產生不重複的4碼數字。

初始化，先將四位數字, 分別設定為獨立的LIST, 內容為0~9

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 第一位 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 第二位 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 第三位 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 第四位 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |

（二）、透過以下程序來猜答案,

1. 猜測一個數字
2. 由相對的位置中, 選擇目前LIST裡的內容, 組成猜測的數字, 數字不可重覆猜測的數字不可與之前重複

EX : 下表中，可猜測的組合為第一位可選數為1,4 以此類推，可能數字僅剩4123

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 第一位 | ~~0~~ | 1 | ~~2~~ | ~~3~~ | 4 | ~~5~~ | ~~6~~ | ~~7~~ | ~~8~~ | ~~9~~ |
| 第二位 | ~~0~~ | 1 | ~~2~~ | ~~3~~ | ~~4~~ | ~~5~~ | ~~6~~ | ~~7~~ | ~~8~~ | ~~9~~ |
| 第三位 | ~~0~~ | 1 | 2 | 3 | ~~4~~ | ~~5~~ | ~~6~~ | ~~7~~ | ~~8~~ | ~~9~~ |
| 第四位 | ~~0~~ | ~~1~~ | ~~2~~ | 3 | ~~4~~ | ~~5~~ | ~~6~~ | ~~7~~ | ~~8~~ | ~~9~~ |

1. 每一次的猜測, 都需符合前面的猜測結果

每一輪猜測結果，以以下結構保留，以新猜測的數值為答案，之前的猜測結果需能符合。

|  |  |
| --- | --- |
| 5678 | 0A0B |
| 1309 | 1A1B |
| 1234 | 4A0B |

1. 不能符合前幾項要求時, 重新產生新的數字
2. 如果結果A==0 And B != 0, 將對應位置的LIST中, 將目前猜測的數字移除

EX, 答案為1234, 本輪猜測為4378 : 0A2B

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 第一位 | 0 | 1 | 2 | 3 | ~~4~~ | 5 | 6 | 7 | 8 | 9 |
| 第二位 | 0 | 1 | 2 | ~~3~~ | 4 | 5 | 6 | 7 | 8 | 9 |
| 第三位 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | ~~7~~ | 8 | 9 |
| 第四位 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | ~~8~~ | 9 |

1. 如果結果A != 0 And B==0, 保留對應位置的數值, 於另外三個LIST中, 將猜測的數字移除,

EX, 答案為1234, 本輪猜測為7834 : 2A0B

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 第一位 | 0 | 1 | 2 | ~~3~~ | ~~4~~ | 5 | 6 | 7 | ~~8~~ | 9 |
| 第二位 | 0 | 1 | 2 | ~~3~~ | ~~4~~ | 5 | 6 | ~~7~~ | 8 | 9 |
| 第三位 | 0 | 1 | 2 | 3 | ~~4~~ | 5 | 6 | ~~7~~ | ~~8~~ | 9 |
| 第四位 | 0 | 1 | 2 | ~~3~~ | 4 | 5 | 6 | ~~7~~ | ~~8~~ | 9 |

1. 若結果為A==0 And B==0, 將目前猜測的數值, 於四個LIST中移除,

EX, 答案為1234, 本輪猜測為7834 : 0A0B

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 第一位 | 0 | 1 | 2 | ~~3~~ | ~~4~~ | 5 | 6 | ~~7~~ | ~~8~~ | 9 |
| 第二位 | 0 | 1 | 2 | ~~3~~ | ~~4~~ | 5 | 6 | ~~7~~ | ~~8~~ | 9 |
| 第三位 | 0 | 1 | 2 | ~~3~~ | ~~4~~ | 5 | 6 | ~~7~~ | ~~8~~ | 9 |
| 第四位 | 0 | 1 | 2 | ~~3~~ | ~~4~~ | 5 | 6 | ~~7~~ | ~~8~~ | 9 |

1. 若A+B的結果為4, 則將不是目前的四個數字, 由目前的陣列中移除,

EX, 答案為1234, 本輪猜測為1243 : 2A2B

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 第一位 | ~~0~~ | 1 | 2 | 3 | 4 | ~~5~~ | ~~6~~ | ~~7~~ | ~~8~~ | ~~9~~ |
| 第二位 | ~~0~~ | 1 | 2 | 3 | 4 | ~~5~~ | ~~6~~ | ~~7~~ | ~~8~~ | ~~9~~ |
| 第三位 | ~~0~~ | 1 | 2 | 3 | 4 | ~~5~~ | ~~6~~ | ~~7~~ | ~~8~~ | ~~9~~ |
| 第四位 | ~~0~~ | 1 | 2 | 3 | 4 | ~~5~~ | ~~6~~ | ~~7~~ | ~~8~~ | ~~9~~ |

（三）、回到猜測數字的流程。

（四）、猜測10還猜不到, 就放棄。

若由使用者回答時，有可能使用者的輸入錯誤，造成無解，限制次數，可必免無解的迴圈。

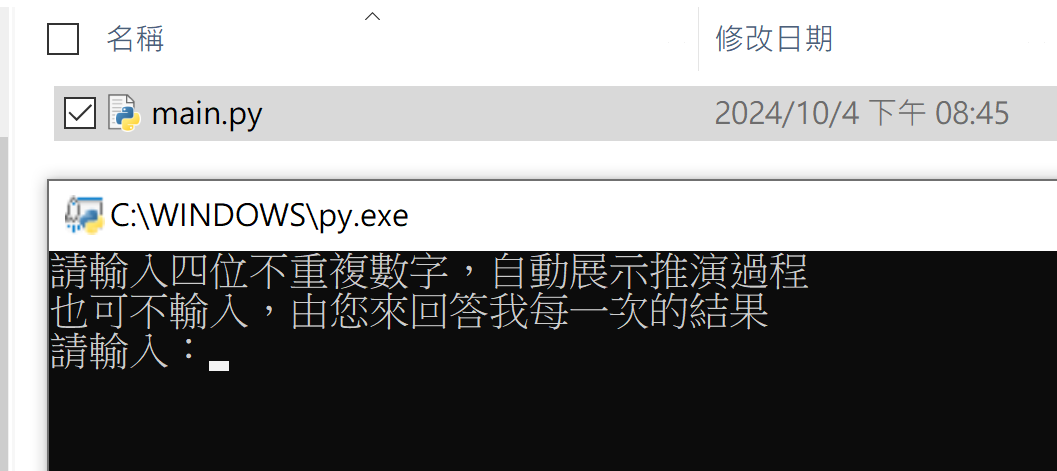
# 執行或安裝步驟說明

安裝步驟

確認安裝python 3.12.5 或以上版本

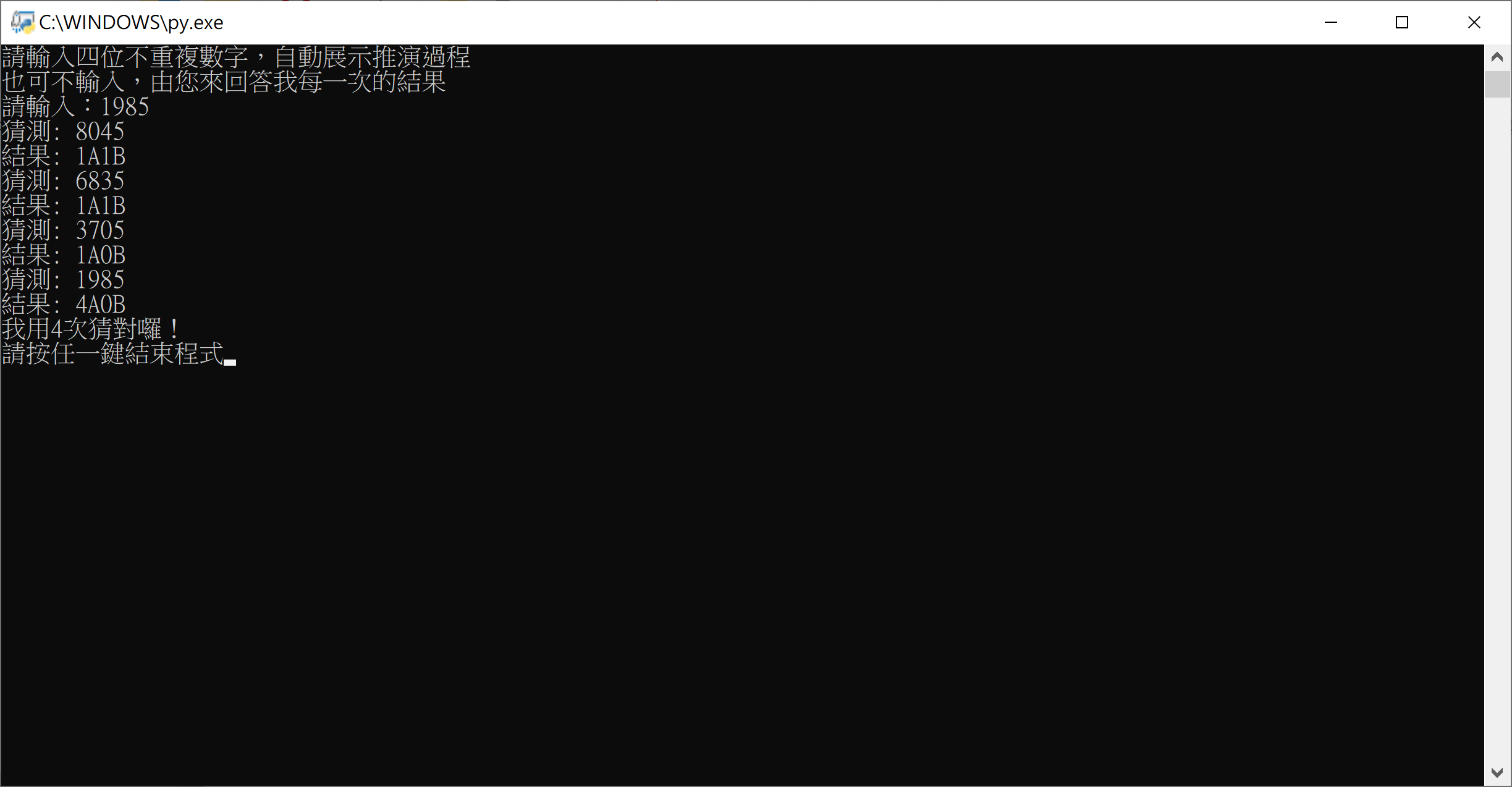
執行方式

點擊main.py可直接執行

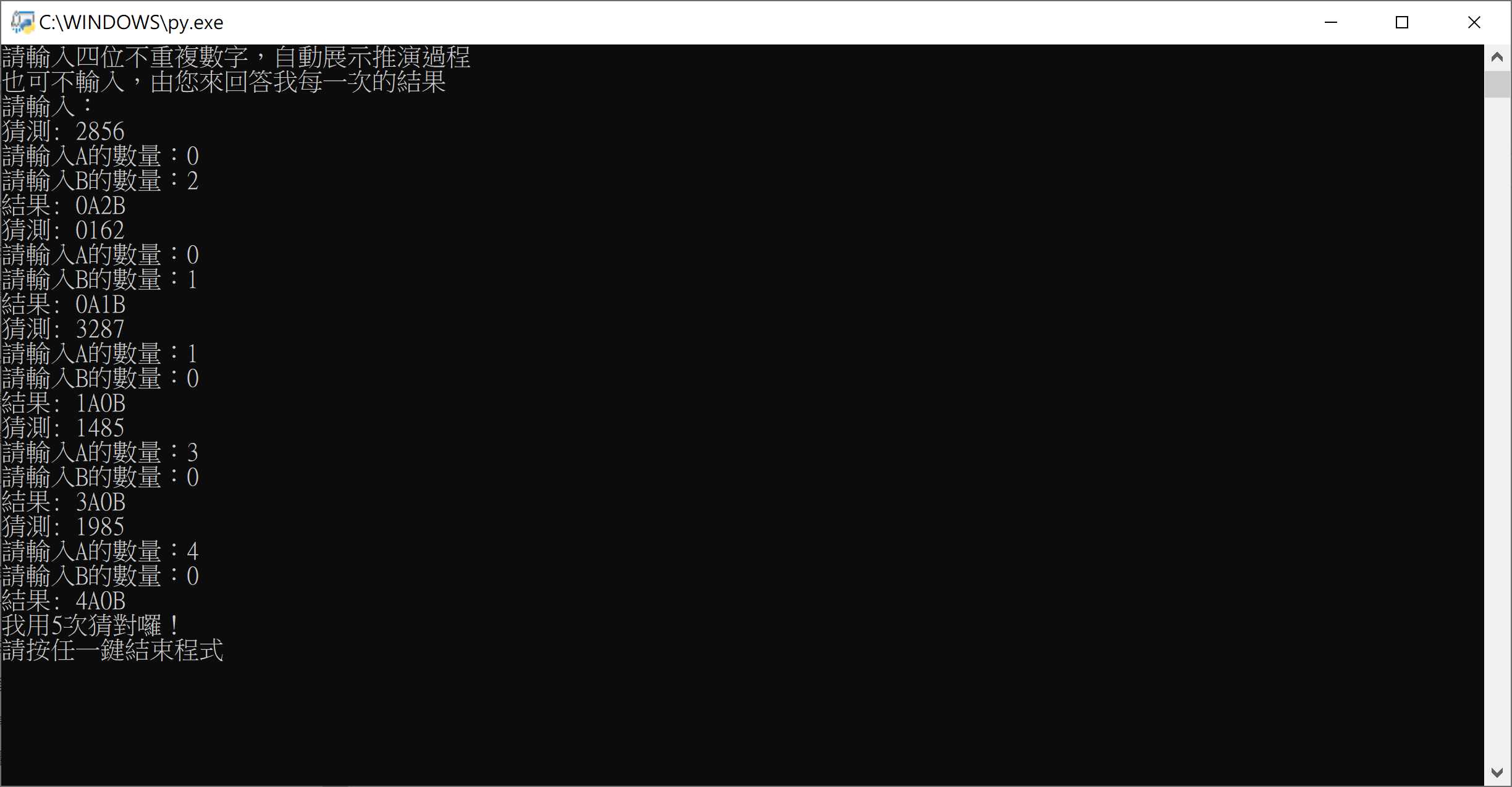


# 執行結果示範與說明。

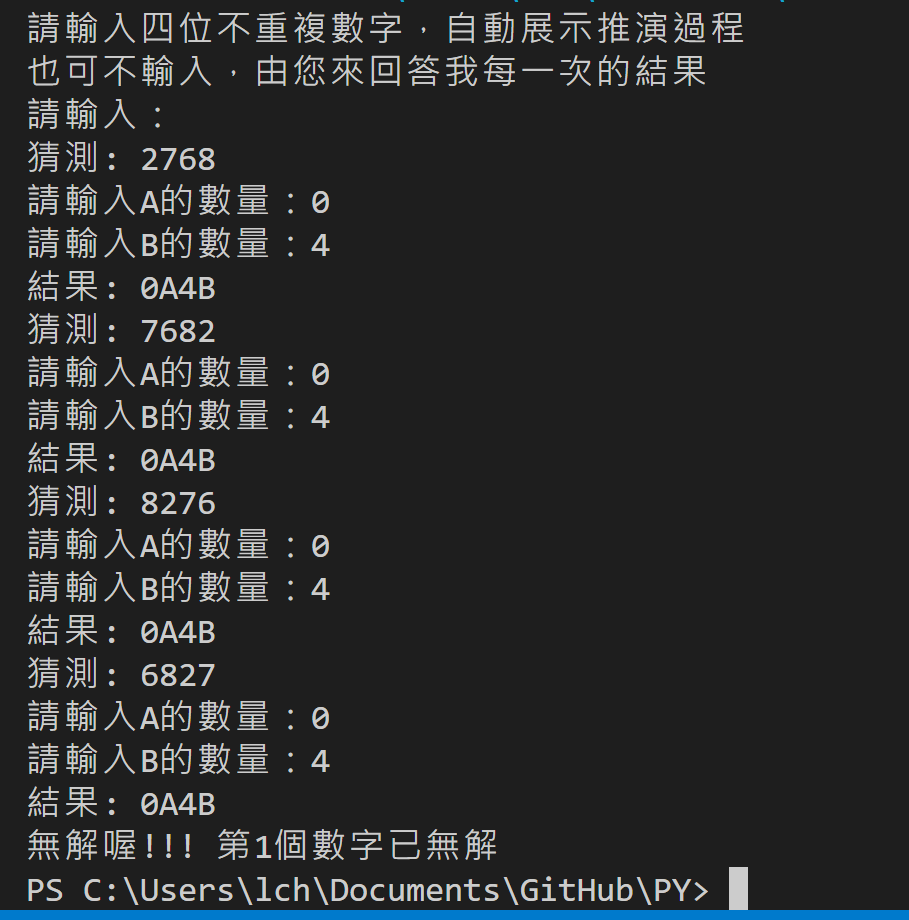
自動, 輸入答案, 由程式進行結果判斷



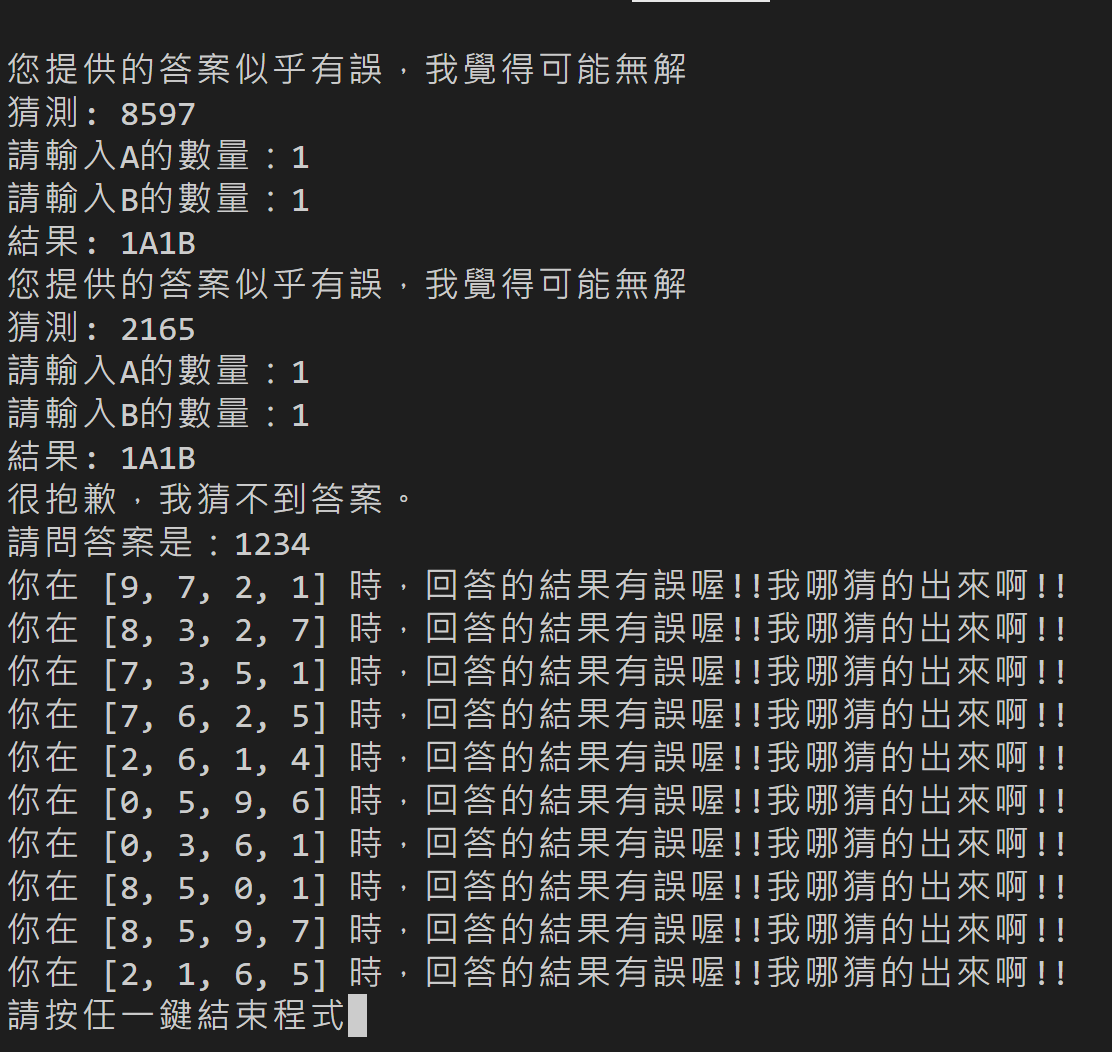
手動, 不輸入答案, 由使用者每一輪輸入A, B數量



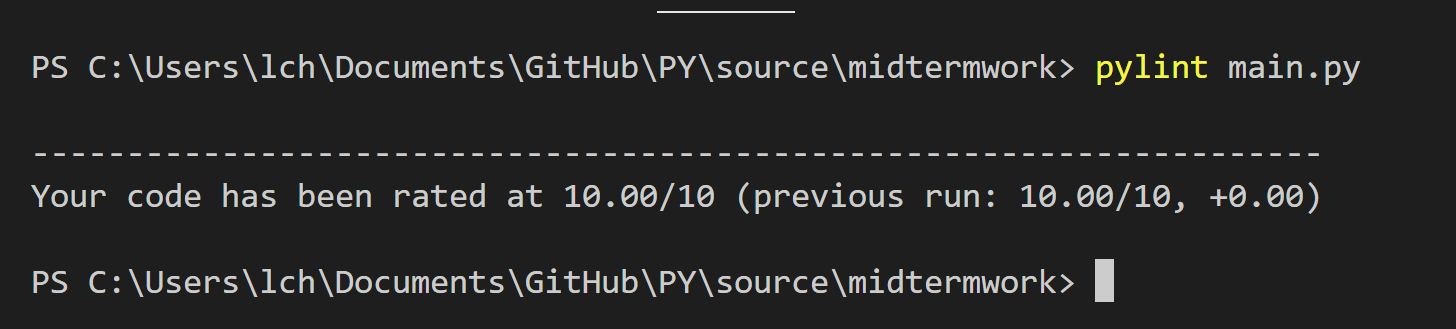
無解判斷



故意輸入無法判定結果的回答, 程式於第十回放棄, 與判斷使用者回覆的結果是否有不正確。



# 程式碼通過Pylint靜態分析



# 可以再改進的部份。

1. 由使用者來進行互動時的介面可以再改進。
2. 使用者的輸入可能不合理，增加了更多的難度。
3. 判定無解的邏輯，可以再進一步提早確認無解的情況。
4. 猜測數字時，可以再改用更有效率的方式，選擇有效的數字。