

# 互動式問題

## (Interactive Task)

- 你的程式必須依照題目的說明 (Protocol) ，  
與伺服器端的裁判程式 (Jury Program) 互動  
完成題目要求計算的任務。
  - 裁判程式會依照你繳交的程式的行為 (截至目前為止的輸出) ，動態產生之後的測資。  
一來一往，直到任務完成為止。

# 互動式問題

## (Interactive Task)

### 原理

- 兩個程式透過 Pipe 連結、溝通、傳遞訊息
  - 程式 A 的輸出(output)  
成為程式 B 的輸入(input)， 反之亦然
  - I/O 的對象不再是檔案，而是另一隻程式
  - 兩者皆需要依照規範好的協議 (Protocol) 進行溝通

## Problem 1

- 目標：

依照互動的結果，猜測 Jury 程式儲存的密碼

練習重點：

1. 熟悉互動式問題的 I/O
2. 練習 Binary Search

## Problem 1

### 需注意的事項

- I/O 的格式需符合題目的規範

ex. 輸出的結尾需要換行 ( `'\n'` )

- 讀取 Jury Program 的回覆時，

建議以字串 token 的方式讀取，再進行字串判讀，避免空白或換行可能導至的問題。

## Problem 2

- 目標：

以有效率的方法，  
計算序列裡每個元素的 “Best Partner”

### 練習重點：

1. 建構解決序列相關問題所需的思維；培養評量程式效能以及設計最佳計算方法的習慣。
2. 建構基礎 DP (Dynamic Programming) 解題思維

## Problem 2

- 觀察：

令  $BP[i]$  為第  $i$  個元素的 Best Partner

則  $i < j \rightarrow BP[i] \leq BP[j]$

