囚徒的挑戰 (Prisoner Challenge)

在一座監獄中，有 500 名囚犯。某一天，典獄長提供了他們一個出獄的機會。他在一個房間中放置了兩個裝了錢的提袋，提袋 A 與提袋 B。每個提袋裝了數量介於 1 與 N 枚錢幣，包含 1 與 N。提袋 A 與提袋 B 的錢幣數量「不相同」。典獄長在囚犯面前宣布了一項挑戰。囚犯們的目標是找出錢幣數量較少的提袋。

除了裝錢幣的提袋之外，這間房間還有一個白板。在任一時刻，白板上都必須寫著單一個數字。初始時，白板上的數字為 0。

接著，典獄長要求囚犯們一個接一個進入房間。進入房間的囚犯不知道在他之前有哪些囚犯進入過房間，也不知道在他之前有多少囚犯進入房間。每當一位囚犯進入房間後，他們會看目前寫在白板上的數字。看了這個數字後，他們必須從提袋 A 或提袋 B 二者中選擇一個。然後這位囚犯會「檢視」他所選的提袋，因此可得知其中的錢幣數量。接下來這位囚犯必須進行下列二「動作」之一：

- 選一個非負整數來覆寫白板上的數字，然後離開房間。請注意他們可以覆寫上與當前相同或不相同的數字。 在此動作後挑戰會繼續 (直到所有 500 位囚犯都進入過房間為止)。
- 辨識出比較少錢幣的提袋。這會使得挑戰直接結束。

典獄長不會讓離開房間的囚犯再次進入房間。

若任何一位囚犯辨識出較少錢幣的提袋，則這些囚犯們會贏得挑戰。若任何一位囚犯辨識錯誤，或者 500 位囚犯都進入了房間但卻無人嘗試去辨識出較少錢幣的提袋，則囚犯們輸掉這個挑戰。

在挑戰開始前，囚犯們在監獄大廳集合並為這個挑戰設計一個三步驟的共同「策略」。

- 他們選一個非負整數 x，是一個他們會想寫在白板上的最大數字。
- 對寫在白板上的任何數字 i $(0 \leq i \leq x)$，他們會決定看到該數字的囚犯該檢視哪一個提袋。
- 他們會決定當得知所選擇的提袋中錢幣的數量後，該囚犯要進何種動作。具體地說，對白板上寫的任何數字 i $(0 \leq i \leq x)$ 以及任何在被檢視的提袋中所包含的錢幣數量 j $(1 \leq j \leq N)$，他們會決定進行下列二動作之一：
  - 0 至 x 間 (包含 0 與 x) 的哪個數字應該被寫在白板上，或
  - 哪個提袋應被辨視為較少錢幣的提袋。

在贏得挑戰後，典獄長會在囚犯們再服役 x 天後釋放他們。

你的任務是為囚犯們設計一個策略，確保他們能夠贏得這個挑戰 (無關於提袋 A 與提袋 B 中的錢幣數量)。你的得分取決於 x 的值 (詳見 Subtasks 一節)。

實作細節 (Implementation Details)

你應實作下列程序：

```
int[][] devise_strategy(int N)
```

- N: 提袋能容納最大可能的錢幣數量。
- 此程序應回傳一陣列的陣列 s 來表示你的策略，s 中的每個陣列是一個由 $N+1$ 個整數構成的陣列。

x 的值是陣列 s 的長度減一。對每個 i 滿足 $0 \leq i \leq x$，陣列 $s[i]$ 表示當一個囚犯進入房間後從白板上讀到數字 i 應做的動作：

1. 若此囚犯應檢視提袋 A 則 $s[i][0]$ 為 0，若此囚犯應檢視提袋 B 則 $s[i][0]$ 為 1。
2. 令 j 為所選擇的提袋中所含的錢幣數量。此囚犯應進行下列的動作：
    - 若 $s[i][j]$ 的值為 -1，則此囚犯應辨識提袋 A 為較少錢幣的提袋。
    - 若 $s[i][j]$ 的值為 -2，則此囚犯應辨識提袋 B 為較少錢幣的提袋。
    - 若 $s[i][j]$ 的值為一非負整數，則此囚犯應將該數字寫在白板上。注意 $s[i][j]$ 必須至多為 x。
    - 此程序被呼叫恰好一次。

範例 (Example)

考慮以下呼叫：

```
devise_strategy(3)
```

令 v 表示在進入房間後該囚犯從白板上看到的數字。一個正確的策略如下：

- 若 $v = 0$ (包含初始值)，檢視提袋 A。
    - 若它包含 1 枚錢幣，辨識提袋 A 為較少錢幣的提袋。
    - 若它包含 3 枚錢幣，辨識提袋 B 為較少錢幣的提袋。
    - 若它包含 2 枚錢幣，將 1 寫在白板上 (覆寫 0)。
- 若 $v = 1$，檢視提袋 B。
    - 若它包含 1 枚錢幣，辨識提袋 B 為較少錢幣的提袋。
    - 若它包含 3 枚錢幣，辨識提袋 A 為較少錢幣的提袋。
    - 若它包含 2 枚錢幣，將 0 寫在白板上 (覆寫 1)。注意此情況可能永遠不會發生，因為我們能推得兩個提袋都包含 2 枚錢幣，並非允許的情形。

要回報這個策略此程序應回傳 `[[0, -1, 1, -2], [1, -2, 0, -1]]`。此被回傳的陣列長度為 2，因此針對此回傳值 x 的值為 $2 - 1 = 1$。

限制 (Constraints)

- $2 \leq N \leq 5000$

子任務 (Subtasks)

1. (5 points) $N \leq 500$，x 的值不得超過 500。
2. (5 points) $N \leq 500$，x 的值不得超過 70。
3. (90 points) x 的值不得超過 60。

若在任一測試中，`devise_strategy` 所回傳的陣列不為一正確的策略，則該子任務你分數為 0。

在子任務 3 中你可能得到部分分數。令 m 為此子任務對於所有測試所回傳的陣列之 x 的最大值。對於此子任務，你的分數將以下表計算：

Condition	Points
$40 \leq m \leq 60$	20
$26 \leq m \leq 39$	$25 + 1.5 \times (40 - m)$
$m = 25$	50
$m = 24$	55
$m = 23$	62
$m = 22$	70
$m = 21$	80
$m \leq 20$	90

範例評分程式 (Sample Grader)

範例評分程式以下列格式讀取輸入：

- line 1: N
- line $2 + k$ $(0 \leq k)$: $A[k]$ $B[k]$
- last line: -1

除了第一列與最後一列外，每一列皆描述一個情境 (scenario)。我們將第 $2 + k$ 列描述之情境稱為情境 k。在情境 k 中，提袋 A 裝了 $A[k]$ 枚錢幣，且提袋 B 裝了 $B[k]$ 枚錢幣。

範例評分程式首先呼叫 `devise_strategy(N)`。x 之值將被設定為回傳的陣列大小減一。接著，若範例評分程式偵測到從 `devise_strategy` 回傳的陣列不滿足實作細節中描述的條件，此程式將輸出下列其中之一錯誤訊息並終止程式執行：

- `s is an empty array`: s 為空陣列 (空陣列無法代表任何合法的策略)。
- `s[i] contains incorrect length`: 存在一個註標 i $(0 \leq i \leq x)$ 使得 $s[i]$ 的長度不等於 $N + 1$。
- `First element of s[i] is non-binary`: 存在一個註標 i $(0 \leq i \leq x)$ 使得 $s[i][0]$ 並非 0 也並非 1。
- `s[i][j] contains incorrect value`: 存在註標 i 與 j $(0 \leq i \leq x, 1 \leq j \leq N)$ 使得 $s[i][j]$ 不介於 -2 與 x 之間。

否則，範例評分程式將會產生兩個輸出。

首先，範例評分程式將以下列格式輸出你的策略執行成果：

- line $1+k$ $(0 \le k)$: 第 k 個情境中，你的策略執行成果。若執行你的策略致使一名囚犯辨識出提袋 A 中的錢幣數量較少，那麼輸出為字元 A。若執行你的策略致使一名囚犯辨識出提袋 B 中的錢幣數量較少，那麼輸出為字元 B。若執行你的策略並不會讓任何一名囚犯辨識出一個錢幣數量較少的提袋，那麼輸出為字元 X。

其次，範例評分程式會在目前的資料夾中以下列格式寫入一個檔名為 `log.txt` 的檔案：

- line $1+k$ $(0 \le k)$: $w[k][0]$ $w[k][1]$ ...

第 $1+k$ 列包含一個整數序列，描述情境 k 中依序被寫在白板上的數字。精確地說，$w[k][l]$ 為第 $l+1$ 個進入房間的囚犯寫在白板上的數字。