

嘉年華獎券 (tickets)

鈴告 (Ringo) 正在參加新加坡的某個嘉年華。他的袋中有若干張獎券。他想將該些獎券用在一個有獎遊戲攤檔。每張券染了 n 種顏色之一，且其上印了一個非負整數。不同券上的整數可能重複。按照嘉年華的奇妙傳統， n 保證是 **偶數**。

鈴告的袋中，每種顏色的券都有 m 張，合計 $n \cdot m$ 張。 i 號色的 j 號券上印有整數 $x[i][j]$ ($0 \leq i \leq n - 1$ 和 $0 \leq j \leq m - 1$)。

該有獎遊戲一共有 k 輪，編號為 0 至 $k - 1$ 。每輪按以下順序進行：

- 鈴告從袋中揀一 **疊** 共 n 張獎券，其中每種顏色的券各一張。然後，他將該疊券交給檔主 (game master)。
- 檔主記下該疊券上的整數 $a[0], a[1] \dots a[n - 1]$ 。該 n 個整數的順序不重要。
- 檔主在幸運大抽獎箱中抽出一張特別卡，並記下該卡上的整數 b 。
- 對於每個 i 從 0 到 $n - 1$ ，檔主會計算 $a[i]$ 和 b 之差的絕對值。記 S 為該些絕對值之和。
- 此輪，檔主送給鈴告價值 S 的獎品。
- 之後該疊券會被棄置，並且此後不能再在遊戲中使用。

經過 k 輪後，鈴告袋中餘下的券將全部棄置。

仔細觀察後，鈴告驚覺該有獎遊戲有蟲惑！幸運大抽獎盒中，原來藏有一部打印機。每一輪，檔主都會求出某個整數 b ，以使該輪的獎品價值最小。檔主所揀的值將打印在當輪的特別卡上。

已知以上一切資訊，鈴告想將獎券巧妙分配到各輪遊戲。換言之，他想每輪揀一疊券，以使獎品總值最大。

編程實作細節

你當實作以下子程序：

```
int64 find_maximum(int k, int[][] x)
```

- k : 輪數。
- x : 一個 $n \times m$ 數組，描述每張券上的整數。每種色的各張券上的整數已按不減順序排好。
- 此子程序會調用恰好一次。
- 此子程序應當調用 `allocate_tickets` (見下段) 恰好一次，描述 k 疊券，每輪一疊。該分配方法應當使獎品總值最大。
- 此子程序應當回傳獎品的最大總值。

子程序 `allocate_tickets` 定義如下：

```
void allocate_tickets(int[][] s)
```

- s : 一個 $n \times m$ 數組。若 i 號色的 j 號券用在第 r 輪，則 $s[i][j]$ 的值應為 r 。若無用該張券，則為 -1 。
- 對於 $0 \leq i \leq n-1$ ，在 $s[i][0], s[i][1], \dots, s[i][m-1]$ 之中， $0, 1, 2, \dots, k-1$ 每個數須出現恰好一次，而其他項皆須為 -1 。
- 若有多個能使獎品總值最大的分配方案，則回答任何一個即可。

範例

範例一

考慮以下調用：

```
find_maximum(2, [[0, 2, 5], [1, 1, 3]])
```

即：

- 共有 $k = 2$ 輪；
- 第 0 種色的券上，分別印有整數 0, 2 和 5；
- 第 1 種色的券上，分別印有整數 1, 1 和 3。

取到最大獎品總值的分配方案可以是：

- 第 0 輪，拎告揀 0 號色的 0 號券（寫有整數 0）和 1 號色的 2 號券（寫有整數 3）。今輪獎品最小可能值為 3。例如檔主可揀 $b = 1$: $|1 - 0| + |1 - 3| = 1 + 2 = 3$ 。
- 第 1 輪，拎告揀 0 號色的 2 號券（寫有整數 5）和 1 號色的 1 號券（寫有整數 1）。今輪獎品最小可能值為 4。例如檔主可揀 $b = 3$: $|3 - 1| + |3 - 5| = 2 + 2 = 4$ 。
- 故獎品總值會是 $3 + 4 = 7$ 。

要回答該分配方案，子程序 `find_maximum` 應如此調用 `allocate_tickets`:

- `allocate_tickets([[0, -1, 1], [-1, 1, 0]])`

最後，子程序 `find_maximum` 應當回傳 7。

範例二

考慮以下調用：

```
find_maximum(1, [[5, 9], [1, 4], [3, 6], [2, 7]])
```

即：

- 僅有一輪，

- 0 號色的券分別寫有整數 5 和 9；
- 1 號色的券分別寫有整數 1 和 4；
- 2 號色的券分別寫有整數 3 和 6；
- 3 號色的券分別寫有整數 2 和 7；

取到最大獎品總值的分配方案可以是：

- 第 0 輪，拎告揀 0 號色的 1 號券（寫有整數 9）、1 號色的 0 號券（寫有數字 1）、2 號色的 0 號券（寫有數字 3）和 3 號色的 1 號券（寫有數字 7）。今輪獎品最小可能值為 12，在檔主揀 $b = 3$ 時取到： $|3 - 9| + |3 - 1| + |3 - 3| + |3 - 7| = 6 + 2 + 0 + 4 = 12$.

要回答該分配方案，子程序 `find_maximum` 應如此調用 `allocate_tickets`:

- `allocate_tickets([[-1, 0], [0, -1], [0, -1], [-1, 0]])`

最後，子程序 `find_maximum` 應當回傳 12.

限制

- $2 \leq n \leq 1500$ 且 n 為偶數。
- $1 \leq k \leq m \leq 1500$
- $0 \leq x[i][j] \leq 10^9$ （對於 $0 \leq i \leq n - 1$ 和 $0 \leq j \leq m - 1$ ）
- $x[i][j - 1] \leq x[i][j]$ （對於 $0 \leq i \leq n - 1$ 和 $1 \leq j \leq m - 1$ ）

子任務

1. (11 分) $m = 1$
2. (16 分) $k = 1$
3. (14 分) $0 \leq x[i][j] \leq 1$ （對於 $0 \leq i \leq n - 1$ 和 $0 \leq j \leq m - 1$ ）
4. (14 分) $k = m$
5. (12 分) $n, m \leq 80$
6. (23 分) $n, m \leq 300$
7. (10 分) 没有其他限制。

樣例評分程式

樣例評分程式按以下格式讀入：

- 第 1 行： $n \ m \ k$
- 第 $2 + i$ 行 ($0 \leq i \leq n - 1$): $x[i][0] \ x[i][1] \ \dots \ x[i][m - 1]$

樣例評分程式按以下格式印出你的答案：

- 第 1 行：`find_maximum` 的回傳值
- 第 $2 + i$ 行 ($0 \leq i \leq n - 1$): $s[i][0] \ s[i][1] \ \dots \ s[i][m - 1]$