



視覺程式(Vision Program)

你正在為一個機器人實作一個視覺程式，每一次機器人的照相機拍攝一張照片並在機器人的記憶體中儲存成一個黑白影像。每一個影像是一個 $H \times W$ 像素的網格，列(row)由0至 $H - 1$ 編號，而行(column)由0至 $W - 1$ 編號。每一張影像恰有兩個黑色像素，其他像素都是白色。

此機器人能以簡單指令所構成的程式來處理影像，你被給予 H 與 W 的值，以及一個正整數 K ，你的目標是寫一個程序來為機器人產生一個程式，對於任何影像，該程式能決定兩個黑像素之間的距離是否恰好為 K ，在這裡，若一個像素在第 r_1 列第 c_1 行而另一像素在第 r_2 列第 c_2 行，則這兩個像素之間的距離為 $|r_1 - r_2| + |c_1 - c_2|$ 。在此公式中， $|x|$ 表示 x 的絕對值，也就是說，若 $x \geq 0$ ，則 $|x| = x$ ；若 $x < 0$ ，則 $|x| = -x$ 。

我們現在描述該機器人如何運作。

此機器人的記憶體是一個足夠大的格子陣列，由0開始編號。每一個記憶體格子可以儲存0或1，格子的內容一旦被設定就不會再被改變。影像是一列一列(row by row)的儲存在編號0到 $H \cdot W - 1$ 的格子中，第一列存在編號0到 $W - 1$ 的格子，而最後一列儲存在編號 $(H - 1)W$ 到編號 $H \cdot W - 1$ 的格子中。特別說明，若第 i 列第 j 行是黑色，則編號 $(i \cdot W + j)$ 格子內的值是1，否則為0。

此機器人的程式是由一序列的指令(instructions)組成，這些指令是由0開始的連續整數來編號，當程式執行時，這些指令是一個一個被執行。每一個指令讀取一個或多個格子(稱為指令的輸入(inputs))並產生一個單一的0或1(稱為指令的輸出(output))，而第 i 個指令的輸出被儲存於格子 $H \cdot W + i$ 。第 i 個指令只能讀取像素的格子或在它之前指令的輸出，也就是第0到 $H \cdot W + i - 1$ 的格子。

有四種指令：

- **NOT**: 有恰好一個輸入。如果輸入是0，它的輸出是1；否則它的輸出是0。
- **AND**: 有一個或多個輸入。它的輸出是1若且唯若所有的輸入都是1。
- **OR**: 有一個或多個輸入。它的輸出是1若且唯若至少一個輸入是1。
- **XOR**: 有一個或多個輸入。它的輸出是1若且唯若奇數個輸入是1。

如果兩個黑色像素的距離恰好是 K ，則此程式最後一個指令的輸出應該是1，否則應該是0。

實作細節(Implementation details)

你應實作以下程序：

```
void construct_network(int H, int W, int K)
```

- H, W : 每一個由機器人照相機所拍攝的影像尺寸。

- K : 一個正整數。
- 此程序應該產生一個機器人的程式。對於每一個機器人相機所拍攝的影像，此機器人的程式應該決定是否兩個黑色像素之間的距離恰好是 K 。

此程序應該呼叫一個或多個以下的程序來添加(append)指令到機器人的程式(機器人的程式一開始是空的):

```
int add_not(int N)
int add_and(int[] Ns)
int add_or(int[] Ns)
int add_xor(int[] Ns)
```

- 分別添加(Append)一個NOT, AND, OR, 或者是 XOR 指令。
- N (for `add_not`): 所添加NOT指令要讀取的輸入格子編號。
- Ns (for `add_and`, `add_or`, `add_xor`): 一個陣列，此陣列包含所添加AND, OR, 或者 XOR 指令要讀取的輸入格子編號。
- 每一個程序回傳(return)儲存該指令輸出的格子編號，對這些指令連續的呼叫則回傳由 $H \cdot W$ 開始的連續整數。

機器人的程式可包含至多10 000個指令，這些指令可讀取的輸入數量總和至多為1 000 000。換句話說，呼叫`add_and`，`add_or`以及`add_xor` 的 Ns 陣列總長度加上呼叫`add_not`的次數不可以超過1 000 000。

在添加最後一個指令之後，`construct_network`程序應該要返回(return)，而此機器人的程式將會以若干影像來測試評定。如果對每一個測試影像，最後一個指令的輸出是1若且唯若兩個黑色像素的距離是 K ，那麼你的解就被評定為通過此測資。

評判你的解可能會產生一些英文的訊息，這些訊息解釋如下：

- **Instruction with no inputs:** 一個空陣列被做為`add_and`, `add_or`, 或者是`add_xor`的輸入。
- **Invalid index:** 一個不正確(可能是負值)的格子編號被做為`add_and`, `add_or`, `add_xor`, 或者是`add_not`的輸入。
- **Too many instructions:** 你的程序試圖添加超過10 000指令。
- **Too many inputs:** 指令總共讀取超過1 000 000個值。

Example

假設 $H = 2$, $W = 3$, $K = 3$ ，只有兩種可能的影像其黑色像素的距離恰好是3。

0	1	2
3	4	5

0	1	2
3	4	5

- Case 1: 黑色像素是0與5
- Case 2: 黑色像素是2與3

一個可能的解是經由以下的呼叫來建構機器人的程式：

1. `add_and([0, 5])`，此呼叫添加一個指令，該指令輸出1若且唯若上述case 1 為真，它的輸出儲存在編號6的格子。
2. `add_and([2, 3])`，此呼叫添加一個指令，該指令輸出1若且唯若上述case 1 為真，它的輸出儲存在編號7的格子。
3. `add_or([6, 7])`，此呼叫添加一個指令，該指令輸出1若且唯若上述兩個case其中之一成立。

限制(Constraints)

- $1 \leq H \leq 200$
- $1 \leq W \leq 200$
- $2 \leq H \cdot W$
- $1 \leq K \leq H + W - 2$

Subtasks

1. (10 points) $\max(H, W) \leq 3$
2. (11 points) $\max(H, W) \leq 10$
3. (11 points) $\max(H, W) \leq 30$
4. (15 points) $\max(H, W) \leq 100$
5. (12 points) $\min(H, W) = 1$
6. (8 points) 每一個影像的第0列第0行的像素是黑色。
7. (14 points) $K = 1$
8. (19 points) 無額外限制。

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: $H \ W \ K$
- line $2 + i$ ($i \geq 0$): $r_1[i] \ c_1[i] \ r_2[i] \ c_2[i]$
- last line: -1

除了第一行與最後一行之外，每一行代表一個影像的兩個黑色像素。第 $2 + i$ 行所表示的影像我們稱為影像 i ，其中一個黑色像素是在第 $r_1[i]$ 列第 $c_1[i]$ 行，另外一個黑色像素在第 $r_2[i]$ 列第 $c_2[i]$ 行。

範例評分程式首先呼叫`construct_network(H, W, K)`，如果`construct_network`違反題目描述中的某些限制，則範例評分程式印出列在實作細節中所說的錯誤訊息後離開。

否則，範例評分程式產生兩個輸出。

首先，範例評分程式以下列格式印出機器人程式的輸出：

- line $1 + i$ ($0 \leq i$): 對於影像 i ，機器人程式的最後一個指令的輸出 (1 or 0)。

其次，範例評分程式在目前的資料夾中以下列格式寫一個檔名為 `log.txt` 的檔案：

- line $1 + i$ ($0 \leq i$): $m[i][0] \ m[i][1] \ \dots \ m[i][c - 1]$

以影像 i 做為輸入時，在 line $1 + i$ 的序列描述機器人程式被執行後儲存在記憶體格子中的值。更明確地說， $m[i][j]$ 是格子 j 的值。請注意， c 的值(序列長度)等於 $H \cdot W$ 加上機器人程式中的指令數。