

## 特殊位置

### 問題描述

考慮一個二維正整數陣列  $A$ ，有  $n$  列 (row) 及  $m$  行 (column)。陣列  $A$  中第  $i$  列第  $j$  行的元素  $A[i][j]$ 。兩個元素  $A[i][j]$  與  $A[s][t]$  的距離定為  $|i-s|+|j-t|$ ，也就是列註標差的絕對值加上與行註標差的絕對值。

對於一個元素  $A[i][j]=x$ ，若與其距離不超過  $x$  的元素總和 (包含  $A[i][j]$ ) 除以 10 的餘數與  $x$  除以 10 的餘數相同，則我們稱  $(i, j)$  為一特殊位置；注意，在計算此總和時，超出陣列範圍的元素不予計算。今給定一陣列，你的任務要找出陣列  $A$  中特殊位置的總數以及每個特殊位置；一個位置以兩個數字表示，分別代表其列註標與行註標。

舉例來說，考慮下列陣列  $A$ ：

1	3	4	1	3	1
1	1	4	1	3	1
1	1	3	2	5	3
4	3	3	1	4	1
5	2	1	1	1	1

其中  $A[0][0]=1$ ，與其距離 1 以內的只有三個元素：其本身、右方的 3 與下方的 1，其總和為  $1+3+1=5$ ，其除以 10 的餘數與  $A[0][0]$  除以 10 的餘數不同，故  $(0, 0)$  並非一特殊位置。而  $A[2][3]=2$ ，與其距離 2 以內有 13 個元素，如圖中陰影處所示，其元素總和為 32，除以 10 的餘數為 2，與  $A[2][3]$  除以 10 的餘數相同，故  $(2, 3)$  為一特殊位置。此例之特殊位置共有 4 個，分別為  $(0, 2)$ 、 $(1, 3)$ 、 $(2, 3)$  以及  $(3, 3)$ 。

### 輸入格式

輸入第一行有兩個正整數  $n$  與  $m$ ， $1 \leq n, m \leq 50$ ，代表陣列的列數與行數。接下去有  $n$  行，每行有  $m$  個整數，表示陣列的內容，其順序為由上而下，由左至右，請參考範例二題目中例子的輸入。同一行的兩數字之間以一個空白間隔。陣列內的數字均為小於 10 的正整數。

### 輸出格式

第一行是一個整數  $k$ ，表示所求之特殊位置總數。接下來  $k$  行，每行有兩個整數，依序表示一個特殊位置的列註標與行註標。輸出時，列註標較小的特殊位置先輸出，若兩個特殊位置的列註標相同，則先輸出行註標較小者。

範例一：輸入 1 5 3 1 4 5 1	範例一：正確輸出 2 0 0 0 2
----------------------------	-----------------------------

範例二：輸入 5 6 1 3 4 1 3 1 1 1 4 1 3 1 1 1 3 2 5 3 4 3 3 1 4 1 5 2 1 1 1 1	範例二：正確輸出 4 0 2 1 3 2 3 3 3
--	---

### 評分說明

輸入包含若干筆測試資料，每一筆測試資料的執行時間限制均為1秒，依正確通過測試筆數給分。其中：

第1子題組 60分： $n = 1$ 。

第2子題組 40分：無額外限制。

試題來源：程式實作 2023年6月

## 蒐集寶石

### 問題描述

小田設計了一個機器人來玩蒐集寶石的遊戲，遊戲的場地是一個  $M \times N$  的二維陣列，每個元素  $(i, j)$  對應在第  $i$  橫列第  $j$  直行的格子，其中橫列由上而下，直行由左往右，皆由 0 開始編號。某些格子是牆壁，機器人無法進入，其他的格子機器人可以進入，這些格子內有若干寶石，但某些格子的寶石數量可能是 0，場地四周視為被牆壁包圍，也就是機器人不能走出界外。遊戲開始時，機器人從編號  $(r, c)$  的格子開始，初始的方向為面向東方(右方)，初始得分與擁有的寶石數量皆為 0。機器人會記錄目前的得分以及行進的方向。機器人的行動規則如下，請計算出機器人在停止時所獲得的寶石數量，其中  $K$  是一個輸入的正整數：

1. 如果走到的格子內無寶石，則停止遊戲；否則
2. 將得分加上該格子目前的寶石數量，並取得一顆寶石。該格子的剩餘寶石數量就會減少一顆；
3. 如果總得分是  $K$  的倍數，則向右轉(順時針轉 90 度)；
4. 若前方是牆壁或者前方是界外，則向右轉；持續本步驟直到前方的格子在場地範圍內且非牆壁；
5. 進入前方的格子，並回到步驟 1 繼續。

以下圖為例，場地是一個  $4 \times 5$  的二維陣列，起點在  $(2, 1)$ ，起始方向朝東，假設  $K=4$ 。機器人一開始會行經  $(2, 1) \rightarrow (2, 2) \rightarrow (2, 3)$ ，此時得分為  $3+2+3=8$ ，為  $K$  的倍數，因此會右轉，方向改為朝南。接下來往南一步走到  $(3, 3)$ ，此時得分為 11，但往前是界外，因此會右轉朝西，此時前方  $(3, 2)$  的位置為牆壁，所以繼續右轉朝北。

2	0	1	1	1
2	-1	0	2	-1
0	3	2	3	0
-1	1	-1	3	0

再前進時到達  $(2, 3)$ ，這個格子一開始有 3 顆寶石，但之前經過一次拿走一顆，所以只剩下 2 顆，故總得分為  $11+2=13$ 。接著會往北走兩次，總得分分別為 15 與 16，在  $(0, 3)$  時右轉到  $(0, 4)$ ，然後會右轉兩次朝西，最後走到  $(0, 3)$  時，該格子的寶石已被取走，剩下 0 顆寶石，遊戲停止。總共獲得的寶石的數量即是停止前所走的總步數(7 步)加上起始位置的 1 顆，一共是 8 顆。

請注意，在原地右轉時並不算再次走到該格子。

### 輸入格式

第一行有五個整數  $M, N, K, r, c$  ( $1 \leq M \leq 100, 2 \leq N \leq 100, 2 \leq K \leq 20$ )，其中  $(r, c)$  為起始位置。接下來  $M$  行，每行有  $N$  個整數，為每一個格子由上而下，由左而右

的值，其中-1 表示牆壁，否則為寶石數量，寶石數量介於 0 到  $K-1$  之間，同一行兩個數值間以一個空白間隔。起始點  $(r, c)$  的位置保證不是牆，且起始點不會四周均無路可走，機器人最終一定會停下來。

### 輸出格式

輸出一個整數，為機器人所蒐集到的寶石數。

範例一：輸入 1 7 3 0 4 1 -1 2 1 2 1 0	範例一：正確輸出 5
---------------------------------------	---------------

範例一說明：經過的格子依序為  $(0, 4)$ 、 $(0, 5)$ 、 $(0, 4)$ 、 $(0, 3)$ 、 $(0, 2)$ ，接下來到達  $(0, 3)$  已無寶石。共取得 5 顆寶石。

範例二：輸入 4 5 4 2 1 2 0 1 1 1 2 -1 0 2 -1 0 3 2 3 0 -1 1 -1 3 0	範例二：正確輸出 8
---	---------------

範例二說明：此為題目中之範例。

### 評分說明

輸入包含若干筆測試資料，每一筆測試資料的執行時間限制均為 1 秒，依正確通過測試筆數給分。其中：

第 1 子題組 60 分： $M = 1$ 。請留意  $M = 1$  時，如遇到右轉必然會連續轉兩次，也就是說右轉即是迴轉。

第 2 子題組 40 分：無額外限制。

試題來源：程式實作 2024 年 10 月

## 字串解碼

### 問題描述

小虎對資訊隱藏很有興趣，他設計了一個編碼方式將一個字串中的字元順序調換，產生一個和原字串不相同的字串，藉以隱藏原本的資訊。首先，小虎定義了一個轉換操作  $t=f(s, e)$ ，即根據  $s$  與  $e$  計算出另一個字串  $t$ ，其中  $s$  是一個由大寫英文字母組成的字串， $e$  是一個 0/1 字串(由 0 與 1 組成)，兩者長度均為  $n$ 。轉換方式有兩大步驟：

1. 若  $e$  中的 1 的個數為奇數，則將  $s$  平分成兩半，然後將前半段與後半段交換，若  $s$  的長度為奇數，則正中間的字元不變。若  $e$  中的 1 的個數為偶數，則  $s$  不改變。
2. 從  $i=0$  到  $n-1$ ，依序根據  $e$  的第  $i$  個字元  $e[i]$  與  $s$  來決定  $t$  的第  $i$  個字元  $t[i]$ ：  
若  $e[i] = 0$ ，則取出  $s$  的第一個字元給  $t[i]$ ，並刪除  $s$  的第一個字元；若  $e[i] = 1$ ，則取出  $s$  的最後一個字元給  $t[i]$ ，並刪除  $s$  的最後一個字元。

以  $s = \text{BCAAD}$  與  $e = 10110$  為例，因為  $e$  中有 3 個 1，所以第一步將  $s$  的前半段 BC 與後半段 AD 交換， $s$  變成 ADABC。第二步驟共有 5 次操作，詳細如下表，每次操作參考的字元  $e[i]$  以及操作後的  $s$  與  $t$  記錄於表的第  $i$  個直欄。

$i$	0	1	2	3	4
$e[i]$	1	0	1	1	0
$s$	ADAB	DAB	DA	D	空
$t$	C	CA	CAB	CABA	CABAD

小虎選定了  $m$  個 0/1 字串 ( $e_0, e_1, \dots, e_{m-1}$ ) 做為編碼表，針對原始字串  $s_0$  的編碼方式為進行上述的轉換  $m$  次，第  $j$  次轉換的結果為  $s_j = f(s_{j-1}, e_{j-1})$ ，最後得到的  $s_m$  就是編碼結果。

小虎興高采烈的將這個編碼方式分享給好友小龍，並吩咐小龍以後傳訊息時都要用這個編碼方式。今天小虎收到了小龍傳來的一則編碼後的訊息，但卻突然發現他不知道如何還原原始的訊息。請寫一支程式，幫助小虎進行訊息的解碼，意即針對一個轉換後的字串，參考小虎的編碼表，計算出原始字串。

### 輸入格式

輸入的第一行有 2 個不超過 100 的正整數  $m$  與  $n$ ，分別為編碼表中 0/1 字串的個數以及每個 0/1 字串的長度。接下來  $m$  行，依序是  $e_0, e_1, \dots, e_{m-1}$ 。最後一行是一個長度為  $n$  的字串，表示小虎收到的訊息，該字串由英文大寫字母組成。

### 輸出格式

一個字串，為編碼前的原始字串。

<p><b>範例一：輸入</b></p> <p>1 5 10110 CABAD</p>	<p><b>範例一：正確輸出</b></p> <p>BCAAD</p>
---	-------------------------------------

**範例一說明：**此為題目中所述之範例。

<p><b>範例二：輸入</b></p> <p>3 6 111110 101101 000000 RETYWQ</p>	<p><b>範例二：正確輸出</b></p> <p>QWERTY</p>
---	--------------------------------------

**範例二說明：**  $e_2 = 000000$  有 0 個 1，可知  $s_2$  轉換為  $s_3$  的過程中不做前後半對調，且由左而右依序取字母後可得  $s_3 = RETYWQ$ ，故  $s_2 = RETYWQ$ 。 $e_1 = 101101$  有 4 個 1，故  $s_1$  轉換為  $s_2$  的過程中不需前後半對調，直接進行轉換的第二步驟可得  $s_2$ ，故  $s_1 = EWQYTR$ 。 $e_0 = 111110$  有 5 個 1，故  $s_0$  轉換為  $s_1$  的過程中需要前後半對調，對調後的結果再由右而左取字母得到  $s_1$ ，故原始字串  $s_0 = QWERTY$ 。

### 評分說明

輸入包含若干筆測試資料，每一筆測試資料的執行時間限制均為 1 秒，依正確通過測資筆數給分。其中：

第 1 子題組 60 分： $m = 1$ 。

第 2 子題組 40 分：無額外限制。

試題來源：程式實作 2022 年 6 月