

# 香港電腦奧林匹克競賽 2017/18 高級組

## 題目總覽

編號	名稱	執行時間限制	記憶體限制	子任務
S181	奇怪的奇數	1.000 s	256 MB	14 + 16 + 13 + 24 + 33
S182	聯賽	1.000 s	256 MB	7 + 13 + 20 + 25 + 10 + 25
S183	桌面圖案	1.000 s	256 MB	12 + 15 + 16 + 21 + 12 + 24
S184	Bogo 翻譯	1.000 s	256 MB	18 + 25 + 31 + 26

### 注意:

除非特別注明，否則輸入輸出將依照以下格式:

- 同一行中，數字與數字或字元之間需有一個空格。
- 同一行中，字元與字元之間並無空格。
- 每個字串需放在獨立的行。
- 輸出將自動被修正如下：每行最尾的連續的空格會被刪除，及在輸出最後補上換行符(如沒有)。其他格式問題則不會修正。

C++ 使用者請注意 `cin` / `cout` 可能導致輸入輸出樽頸使程式執行變慢。

有些題目可能需要使用 64 位元整數。在 Pascal 中它是 `int64`。在 C/C++ 中它是 `long long` 而其 `scanf` / `printf` 代號是 `%lld`。

所有題目均有細分多個子任務，你需要通過該子任務中的所有測試數據才能得到分數。

## S181 - 奇怪的奇數

執行時間限制: 1.000 s / 記憶體限制: 256 MB

愛麗絲認為一個人孤單地吃晚飯十分奇怪，所以她邀請她的  $N - 1$  位朋友到她家共進晚餐，換言之，共有  $N$  個人會在愛麗絲家用膳。

在她的朋友到訪前，愛麗絲炸了很多雞塊，並把雞塊放在  $N$  個碟上。由於愛麗絲不擅長除數，因此她不選擇平均分配雞塊，只是隨意地在每個碟上放一些雞塊。詳細來說，她在第  $i$  個碟上放了  $A_i$  件雞塊。

分配好雞塊後，愛麗絲把碟順時針地放到一張圓桌上，對於  $1 \leq i < N$ ，第  $i + 1$  個碟在第  $i$  個碟的左方，並且第 1 個碟在第  $N$  個碟的左方。

當愛麗絲擺好碟子後，她發現一些碟子中的雞塊件數為奇數（即單數），她認為吃奇數件雞塊十分奇怪，因此她打算通過一些操作，令每個碟上的雞塊件數都是雙數。

每一次，愛麗絲會實行以下兩個行動中的其中一個：

- 她在第  $i$  個碟中拿取一件雞塊並放進它左邊的碟子，愛麗絲需要  $C$  秒去進行這行動。
- 她在第  $i$  個碟中拿取一件雞塊並放進它右邊的碟子，愛麗絲需要  $D$  秒去進行這行動。

現在，愛麗絲想知道她最少需要多少秒才能令所有碟上擺放的雞塊件數都是雙數。留意即使一個碟子上沒有任何雞塊，只要沒有碟子上的雞塊件數是奇數，愛麗絲依然接受這個方案。

## 輸入

第一行有三個整數，分別是  $N, C, D$ 。

第二行有  $N$  個整數： $A_1, A_2, \dots, A_N$ 。

## 輸出

第一行且唯一一行輸出一個整數，代表愛麗絲最少需要用多少秒去令所有碟上擺放的雞塊件數都是雙數。如不可能滿足以上條件，輸出  $-1$ 。

## 樣例

輸入                      輸出

1	4 6 9	12
	1 0 3 2	

其中一個最優方案:



2	4 3 2	4
	1 1 1 1	

3	3 3 2	-1
	2 3 4	

4	6 2 3	6
	1 4 1 5 9 2	

## 子任務

對於所有數據:

$$2 \leq N \leq 10^5$$

$$0 \leq A_i \leq 18$$

$$1 \leq C, D \leq 10^4$$

估分                      約束條件

1	14	$N = 3$
2	16	恰好有兩個 $A_i$ 是單數。
3	13	所有 $A_i$ 都是單數。
4	24	$2 \leq N \leq 1000$
5	33	無額外約束。

## S182 - 聯賽

執行時間限制: 1.000 s / 記憶體限制: 256 MB

三隊球隊 Alpha、Beta 和 Gamma 參加一年一度的幻球聯賽。聯賽以單循環形式進行，即每兩隊皆對賽共一次。

在  $X$  隊和  $Y$  隊的比賽中，比分会以「 $X\ x - y\ Y$ 」或「 $Y\ y - x\ X$ 」紀錄，當中的  $x$  和  $y$  皆是非負整數。比分代表在這場比賽中， $X$  隊進了  $x$  球、失了  $y$  球，而  $Y$  隊進了  $y$  球、失了  $x$  球。由於幻球不是足球，比分之大小可能超出你的想像。

如果  $x > y$ ，則  $X$  隊贏得比賽並獲得  $W$  分， $Y$  隊落敗而且沒有獲得積分。

如果  $x = y$ ，則兩隊打成平手並各獲得  $D$  分。

如果  $x < y$ ，則  $X$  隊落敗而且沒有獲得積分， $Y$  隊贏得比賽並獲得  $W$  分。

現在，每隊的總積分、總入球數和總失球數皆為已知，卻丟失了每場比賽的比分。你的任務是找出一套符合已知條件的完整比分。

## 輸入

第一行只包含兩個整數， $W$  和  $D$ ，分別是贏得一場比賽獲得的積分和在一場比賽打成平手獲得的積分。

第二行只包含三個整數， $P_A$ 、 $S_A$  和  $C_A$ ，分別是 Alpha 隊的總積分、總入球數和總失球數。

第三行只包含三個整數， $P_B$ 、 $S_B$  和  $C_B$ ，分別是 Beta 隊的總積分、總入球數和總失球數。

第四行只包含三個整數， $P_G$ 、 $S_G$  和  $C_G$ ，分別是 Gamma 隊的總積分、總入球數和總失球數。

## 輸出

如果有一套符合已知條件的完整比分，輸出三行。每行皆以「 $X\ x - y\ Y$ 」表示一場比賽的比分，當中的  $x$  和  $y$  分別是  $X$  隊和  $Y$  隊在該場比賽的入球數。

否則，請輸出 `Impossible`。

如果有多於一套符合已知條件的完整比分，你可以輸出任何一套。另外，比分亦可以以任何次序輸出。

## 樣例

	輸入	輸出
1	<pre>3 1 6 2 0 1 0 1 1 0 1</pre>	<pre>Alpha 1 - 0 Beta Alpha 1 - 0 Gamma Beta 0 - 0 Gamma</pre>
2	<pre>3 1 6 3 0 1 0 1 1 0 1</pre>	<pre>Impossible</pre>

## 子任務

對於所有數據:

$$0 \leq D \leq W \leq 5$$

$$0 \leq P_A, P_B, P_G \leq 2W$$

$$0 \leq S_A, S_B, S_G, C_A, C_B, C_G \leq 10^9$$

	估分	約束條件
1	7	$S_A = S_B = S_G = C_A = C_B = C_G = 0$
2	13	$W > 2D$ $P_A = P_B = P_G = 2D$ $S_A = C_A$ $S_B = C_B$ $S_G = C_G$
3	20	$0 \leq S_A, S_B, S_G, C_A, C_B, C_G \leq 20$
4	25	$0 \leq S_A, S_B, S_G, C_A, C_B, C_G \leq 10^6$
5	10	$W = 0$
6	25	無額外約束

## S183 - 桌面圖案

執行時間限制: 1.000 s / 記憶體限制: 256 MB

愛麗絲喜歡試用不同的作業系統。不同作業系統上的使用者介面都有不同，例如 macOS 缺少了 Microsoft Windows（微軟視窗）的「開始功能表」，所以有些 Microsoft Windows 使用者轉用 macOS 時會感到困惑。不過，不同的使用者介面也有相同之處。在 Microsoft Windows，你通常會見到一個桌面，桌面上顯示著不同的桌面圖案，例如「資源回收筒」、「我的電腦」及各類檔案；在 macOS，你也會看見一個桌面，上面顯示著不同的桌面圖案，例如已連接的硬盤及各類檔案。這兩種桌面的桌布款式亦可自訂。

「桌面」這功能很常見於使用者介面。愛麗絲正在試用的作業系統上的使用者介面亦有桌面，桌面上可放置圖案。這個作業系統名為 BWOS，源於它的介面只有黑色和白色。由於桌面上的圖案及桌布均是黑白色，有時候圖案並不可見。例如當使用者把一個黑色桌面圖案放在桌布上黑色的位置，這個圖案便不可見到。

要避免令圖案不可見，使用者可以選用一張合適的桌布然後把桌面圖案適當地排置好。在一個好的桌面圖案排位裏，重要的桌面圖案應要可見，而不可見的桌面圖案應是不重要的。

準確點來說，桌面是一個  $R$  行  $C$  列的長方網格，而每格都是一個單位正方形空格，可容納一個桌面圖案。第  $i$  行  $j$  列的桌面空格將標記為  $(i, j)$ ，故左上角的桌面空格將標記為  $(1, 1)$ ，右下角的桌面空格將標記為  $(R, C)$ 。

總共有  $K$  張桌布可選用，由 1 至  $K$  編號。每張桌布都有  $R$  行和  $C$  列的方塊，每個方塊都是一個黑色或白色的單位正方形。桌布方塊的標記方法與桌面空格的標記方法相同，而每一個桌布方塊都對應著一個桌面空格。因此，選用一張桌布後，桌面空格  $(i, j)$  的背景顏色會是那張桌布的桌布方塊  $(i, j)$  的顏色。

總共有  $N$  個桌面圖案，每個圖案可以是黑色或者白色，置於一個桌面空格內。兩個桌面圖案不可以置於同一個桌面空格。當一個桌面圖案的顏色與它所置的桌面空格顏色不相同，該桌面圖案便是可見的，否則便是不可見的。例如，一個白色的桌面圖案置於黑色的桌面空格，它就是可見的；如果它置於白色的桌面空格，它就是不可見的。

除非  $N$  等於  $R \times C$ ，否則桌面上存在至少一個空的桌面空格。要排置桌面圖案，使用者可以重覆地將一個桌面圖案拖曳至一個空的桌面空格，但千萬不可以拖曳至已置有桌面圖案的桌面空格，因為這樣會令 BWOS 立即崩毀。每一次拖曳只會移動一個桌面圖案。如果  $N$  等於  $R \times C$ ，那麼所有桌面空格都已置有桌面圖案，因此所有桌面圖案都不可以移動。

每個桌面圖案都編配了一個重要值。愛麗絲想選用一張桌布然後排置桌面圖案，以最大化所有可見桌面圖案的重要值總和。她不在乎哪一張桌布被選用，但不想拖曳多於 6400 次。你可以幫助她嗎？

## 輸入

第一行有三個整數  $N$ 、 $R$  和  $C$ 。

以下  $N$  行中的第  $i$  行有四項資料  $r_i$ 、 $c_i$ 、 $d_i$  和  $v_i$ 。資料  $r_i$ 、 $c_i$  和  $v_i$  是整數，而  $d_i$  是英文字母 **B**（代表黑色）或者 **W**（代表白色），意思是位於桌面空格  $(r_i, c_i)$  的桌面圖案的顏色和重要值分別是  $d_i$  和  $v_i$ 。保證  $1 \leq r_i \leq R$ 、 $1 \leq c_i \leq C$ 、及不會有兩個桌面圖案置於同一個桌面空格。

下一行是一個整數  $K$ 。

然後有  $K$  組輸入，第  $i$  組輸入對應著桌布  $i$ 。每組輸入有  $R$  行，每行是一個有  $C$  個英文字母的字串，這些英文字母是 **B**（代表黑色）或者 **W**（代表白色）。在這  $R$  行中的第  $i$  行的第  $j$  個英文字母代表桌布方塊  $(i, j)$  的顏色。

## 輸出

第一行輸出三個整數  $S$ 、 $W$  和  $M$ ，分別是最大的可見桌面圖案的重要值總和、所選用的桌布編號及拖曳的執行次數。注意， $1 \leq W \leq K$  及  $0 \leq M \leq 6400$ 。

然後輸出  $M$  行，其中第  $i$  行有四個以空格分隔的整數  $r1_i$ 、 $c1_i$ 、 $r2_i$  和  $c2_i$ ，意思是在第  $i$  次拖曳，位於桌面空格  $(r1_i, c1_i)$  的桌面圖案被移至桌面空格  $(r2_i, c2_i)$ 。拖曳至原空格（即  $r1_i = r2_i$  及  $c1_i = c2_i$ ）是可接受的。

如有多於一個答案，你可以輸出任何一個。

## 計分

在一個子任務中：

- 如果你的程序對所有測試數據輸出正確的最大可見桌面圖案重要值總和、桌布編號及拖曳步驟，你會得到該子任務的 100% 分數。
- 否則的話，如果你的程序對所有測試數據輸出正確的最大可見桌面圖案重要值總和，並且以正確格式輸出任何桌布編號及不會令 BWOS 崩毀的拖曳步驟，而且  $1 \leq W \leq K$  及  $0 \leq M \leq 6400$ ，你會得到該子任務的 40% 分數。
- 否則的話，你不會得到任何分數。

## 樣例

輸入

輸出

<b>1</b>	<pre>1 2 2 1 1 W 10 1 WW WW</pre>	<pre>0 1 0</pre>
----------	-----------------------------------	------------------

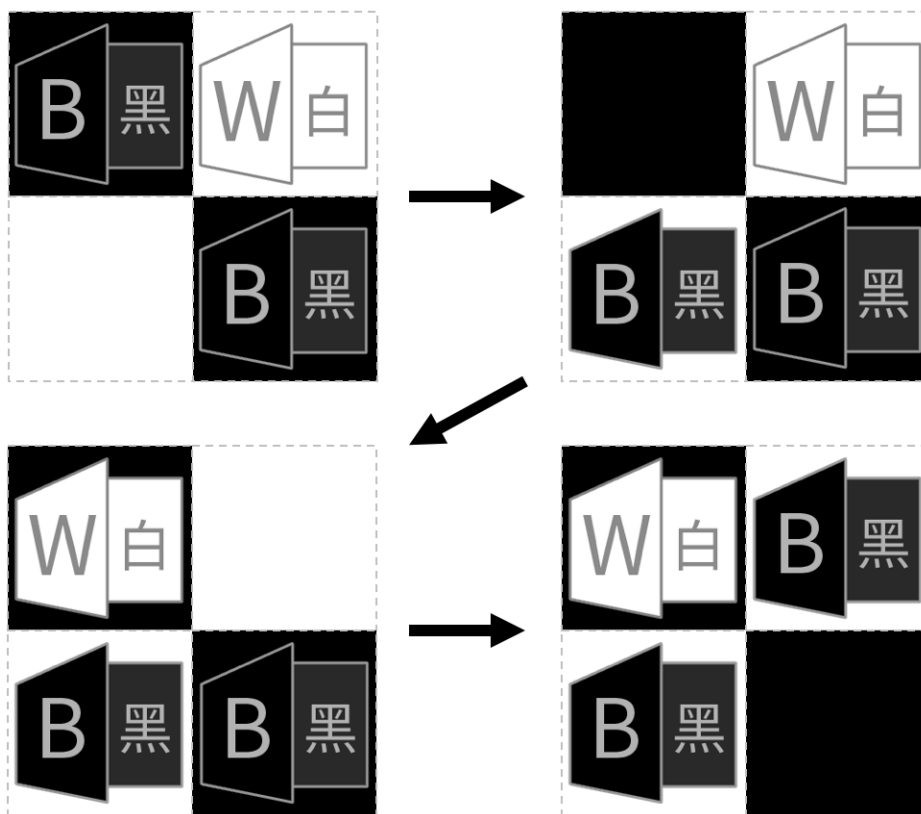
2

1 2 2	10 2 1
1 1 W 10	1 1 2 1
2	
WW	
WW	
WW	
BW	

3

3 2 2	30 1 3
1 1 B 10	1 1 2 1
1 2 W 10	1 2 1 1
2 2 B 10	2 2 1 2
1	
BW	
WB	

下圖演示答案：



4

3 2 2	30 1 0
1 1 B 10	
1 2 W 10	
2 2 B 10	
1	
BW	
WB	

此樣例只得 40% 分數



## 子任務

對於所有數據:

$$1 \leq N \leq R \times C$$

$$1 \leq R, C \leq 80$$

$$0 \leq v_i \leq 10^5$$

$$1 \leq K \leq 100$$

	估分	約束條件
<b>1</b>	12	$R, C \leq 40$ $K = 1$ 所有桌布方塊都有相同顏色
<b>2</b>	15	$R, C \leq 40$ $K = 1$
<b>3</b>	16	$R, C \leq 40$ $N = R \times C$
<b>4</b>	21	$R, C \leq 40$ $v_i = 1$
<b>5</b>	12	$R, C \leq 40$
<b>6</b>	24	無額外約束

## S184 - BOGO 翻譯

執行時間限制: 1.000 s / 記憶體限制: 256 MB

查理是 Bogo 的實習生，他正在開發一個新功能「Bogo 翻譯」。查理不擅長編寫程式，所以專誠來請你幫忙。

由於這是份實習專案，系統只能將語言  $A$  翻譯成語言  $B$ 。要進行翻譯，系統應該：

1. 將一句語言  $A$  句子中每個單詞獨立地翻譯至語言  $B$ 。句子裏的單詞之間都以恰好一個空格分隔，而每個單詞都由小楷英文字母構成。
2. 根據句子結構的差異重組句子。

系統有一個資料庫，資料庫儲存了  $N$  組單詞  $(WordA_i, WordB_i)$ ， $i = 1, 2, \dots, N$ 。  $WordA_i$  是語言  $A$  的單詞，而  $WordB_i$  是語言  $B$  的單詞。留意，所有  $WordA_i$  都互不相同，但  $WordB_i$  有可能出現相同。所有翻譯任務都會使用同一個資料庫。

對於每一個翻譯任務，系統應將待譯句子中所有有出現的  $WordA_i$  轉換成  $WordB_i$ 。有些單詞未必能在  $WordA_i$  中找到，也許它們是專有名詞。這些單詞不應進行轉換。

下一步，系統應根據兩個句子結構的差異重組句子，這兩個句子結構分別以兩個字串表示： $PattA$  及  $PattB$ 。它們專為待譯句子而設；假設待譯句子中的單詞數目是  $x$ ，則兩個句子結構均由  $x$  個不相同的大楷英文字母組成。 $PattB$  是  $PattA$  的一個排列（permutation）。如果  $PattA[i] = PattB[j]$ ，則待譯句子中的第  $i$  個單詞應變成譯後句子的第  $j$  個單詞。

舉例：

- $WordA_1 = \text{charlie}$ ,  $WordA_2 = \text{i}$ ,  $WordA_3 = \text{am}$
- $WordB_1 = \text{charli}$ ,  $WordB_2 = \text{watashiwa}$ ,  $WordB_3 = \text{desu}$
- 待譯句子： $\text{i am charlie}$
- 每個單詞獨立翻譯後： $\text{watashiwa desu charlie}$
- 句子結構： $PattA = \text{SVO}$ ,  $PattB = \text{SOV}$
- 重組句子後： $\text{watashiwa charlie desu}$

查理能否繼續被錄用，全看系統的效率。請你幫他實現 Bogo 翻譯，並將  $M$  句語言  $A$  的句子翻譯成語言  $B$ 。

## 輸入

第一行有一個整數  $N$ ，資料庫裏單詞組的數目。

接下來會有  $N$  組輸入，每組兩行。第  $i$  組的第一行是  $WordA_i$ ，第二行是  $WordB_i$ 。所有  $WordA_i$  都互不相同。

下一行有一個整數  $M$ ，翻譯任務的數目。

然後是個  $M$  個翻譯任務。每個翻譯任務都以三行描述：第一行是一句語言  $A$  的句子，第二行是字串  $PattA$ ，第三行是字串  $PattB$ 。

## 輸出

你的輸出應有  $M$  行，第  $i$  行是將第  $i$  個翻譯任務根據上述方法翻譯而成的語言  $B$  句子。

## 樣例

輸入

輸出

<b>1</b>	<pre>4 d m r f m s f l 1 d r m d m d m r m f f m r f ABCDEFGHIJKLMN ABCDEFGHIJKLMN</pre>	<pre>m f s m s m s f s l l s f l</pre>
----------	--	--

此樣例對應子任務 1 及 2

<b>2</b>	<pre>5 dont m i ngo love zungji you nei like zungji 2 i love hkoi SVO SVO you dont like noip SNVO SNVO</pre>	<pre>ngo zungji hkoi nei m zungji noip</pre>
----------	--	--

此樣例對應子任務 2

<p>3</p> <pre> 0 5 charlie loves bogo MVF FVM single A A isnt the task cool WXYZ XYWZ to be or not to be ABCDEF EFCDAB senior is easier than junior SIETJ JIETS </pre>	<pre> bogo loves charlie single the task isnt cool to be or not to be junior is easier than senior </pre>
--	---

此樣例對應子任務 3

<p>4</p> <pre> 5 charlieli chariri i watashiwa am desu coding koodinguga like sukidesu 3 charlieli G G i am charlieli XYZ XZY i like coding SVO SOV </pre>	<pre> chariri watashiwa chariri desu watashiwa koodinguga sukidesu </pre>
--	---

## 子任務

對於所有數據：

$$0 \leq N \leq 300$$

$$1 \leq M \leq 10000$$

資料庫裏的每個單詞 ( $\text{Word}A_i$ 、 $\text{Word}B_i$ ) 及所有句子中的單詞均由最少 1 個、最多 15 個小楷英文字母組成。

每句句子有最少 1 個、最多 26 個單詞。

所有  $M$  句句子的單詞總數不多於 10000。

測試數據裏的句子語法未必正確，甚至未必以現實語言寫成。

### 估分

### 約束條件

- |   |    |   |
|---|----|---|
| 1 | 18 | 資料庫裏所有單詞 ( $\text{Word}A_i$ 、 $\text{Word}B_i$ ) 及所有句子中的單詞均由恰好 1 個小楷英文字母組成。 |
| 2 | 25 | 對於每個翻譯任務， $\text{Patt}A = \text{Patt}B$ 。                                   |
| 3 | 31 | $N = 0$   |
| 4 | 26 | 無額外約束條件   |