



0_送分題 (Hello World)

(30分)

時間限制: 1 second

記憶體限制: 256 MB

前言

比賽開始了！

趕快驗證一下，

網路是否設定正確？

上傳競賽程式是否順利？

程式解答是否用 STDOUT 輸出？

都沒問題，30分就到手了！繼續...衝！衝！衝！

題目敘述

請寫一個程式輸出Hello World!

輸入格式

本題無需輸入值

輸出格式

[A~Z][a~z]、空格，以及常用英文符號。

資料範圍

[A~Z][a~z]、空格，以及驚嘆號 “!”

測試範例

輸入範例 1

(無輸入值)

輸出範例 1

Hello world!

範例說明

輸入範例1, 無輸入值, 簡單而快樂的輸出Hello World!

1_No_1_can_solve_this_problem!

(2分 / 8分)

時間限制: 1 second

記憶體限制: 256 MB

題目敘述

就如同標題所說的，沒有 1 就可以解出這個問題！

給你一個正整數 N ，請輸出最小的非負整數 K ，使得 $N + K$ 的十進位表示法中沒有任何一個位數是 1。

輸入格式

輸入只有一個正整數 N 。保證輸入不會有任何的前綴 0。

輸出格式

輸出一個非負整數 K ，為題目中所要求的值。請注意，除非你的輸出是 0，否則你的輸出不應該有任何的前綴 0。

資料範圍

- $1 \leq N < 10^{100000}$

子任務

- 子任務 1 滿足 $N < 10^6$
- 子任務 2 無額外限制

測試範例

輸入範例 1

```
1
```

輸出範例 1

```
1
```

輸入範例 2

```
11
```

輸出範例 2

9

輸入範例 3

42

輸出範例 3

0

範例說明

範例輸入 1 中，只要把 1 加上 1，變成 2 之後十進位表示法中沒有任何一個位數是 1 了。

範例輸入 2 中，只要把 11 加上 9，變成 20 之後十進位表示法中沒有任何一個位數是 1 了。

範例輸入 3 中，42 本來就沒有任何一個位數是 1，故輸出 0。

2_能源危機 (Energy_Crisis)

(2 分/8 分)

時間限制: 1 second

記憶體限制: 256 MB

題目敘述

在未來的2050年，地球上的資源已經嚴重稀缺，而人類為了獲取能源，開發出了一種能量結晶稱為「數晶」。每個數晶上都寫著一個數字，在對的條件下能釋放巨大的能量，尤其當找到特定的乘積配對時，即能量乘數「K」，能量的釋放效率將達到最優。換句話說，如果可以找到兩個數晶上面所寫著的數字相乘為 K 即可獲得大量能量。

作為一名數晶工程師，你被賦予了這樣的任務：從一批大小為 N 的初始數晶集合 A_1, A_2, \dots, A_N 中 (A_i 代表第 i 個數晶上的數字，數字可以重複出現)，通過精確的操作來獲得能量乘數「K」中的能量。你的操作包括：

1. 能量增幅（花費 x ） — 選擇一個數晶，進行能量增幅使其值增加一。
2. 能量抑制（花費 y ） — 選擇一個數晶，進行能量抑制使其值減少一。
3. 數晶融合（花費 z ） — 選擇兩個數晶進行融合，融合後的新數晶將返回集合中，新數晶上的數字為被融合的兩個數晶的數字相加，被融合的兩個數晶會消失。

在操作完成後的數晶集合中，你希望能找到兩個數晶上的數字相乘為 K ，並且要計算出最少的花費。請你展現你身為數晶工程師的實力，寫一個程式來解決這個問題。

輸入格式

第一行輸入 2 個數字 N, K 分別代表初始數晶數量和能量乘數。

第二行輸入 3 個數字 x, y, z 分別代表各個操作所需的花費。

第三行輸入 N 個數字 A_1, A_2, \dots, A_N ，代表初始數晶集合中的數字。

輸出格式

輸出一個數字，代表讓集合中存在兩個數字相乘為 K 的最小花費。

資料範圍

- $2 \leq N \leq 10$ 。
- $1 \leq K \leq 1000$
- $1 \leq A_i \leq 1000$ ($\forall i \in [1, N]$)
- $1 \leq x, y, z \leq 10^7$

子任務

- 子任務 1 滿足 $x = y = 1, z = 10^7$ 。
- 子任務 2 無額外限制。

測試範例

輸入範例 1

```
3 10
1 1 1
2 2 2
```

輸出範例 1

```
2
```

輸入範例 2

```
5 20
1 1 10
1 2 3 5 7
```

輸出範例 2

```
1
```

範例說明

在範例 1 中，我們可以先把其中兩個數晶用融合操作合併得到一個數值為 4 的數晶，再把這個數晶的值加一，這樣就有一個數值為 2 和 5 的數晶可以相乘等於 10。

3_得字串上的詢問(Queries_on_Destring)

(10分)

時間限制: 2 seconds

記憶體限制: 1024 MB

題目敘述

劉得徵很喜歡 49 這個數字，所以他定義一個字串為得字串，如果這個字串的每一個字元都是 4 或 9。

劉得徵收到了朋友 LDC 紿他的一個得字串 s 。他為了在 IMO 拿到 49 分，對於所有正整數 i 滿足 $1 \leq i \leq Q$ ，劉得徵想要找到兩個正整數 x, y 使得

- $L_i \leq x \leq y \leq R_i$
- $s[x : y]$ 不是完全平方數
- 在 x, y 滿足以上兩個條件的情況下， $y - x + 1$ 越小越好

對於所有正整數 i 滿足 $1 \leq i \leq Q$ ，判斷劉得徵是否能找到滿足所有條件的兩個正整數 x, y 。如果可以的話，找到 $y - x + 1$ 的最小可能值以及有多少組 x, y 滿足所有條件。

題目用到的定義說明如下：

- 對於一個字串 t 以及兩個正整數 l, r 滿足 $1 \leq l \leq r \leq |t|$ ($|t|$ 為字串 t 的長度)，定義 $t[l : r]$ 為 t 從第 l 個字元開始第 r 個字元結束的子字串，即 $t_l t_{l+1} \dots t_r$ 。
- 對於一個每個字元都是數字的字串 t ，定義 t 是完全平方數如果 $\sum_{i=1}^{|t|} t_i \cdot 10^{|t|-i}$ 是完全平方數。
- 對於一個非負整數 x ，定義 x 是完全平方數如果以下兩個條件有至少一個成立
 - 存在非負整數 y 使得 $x = y^2$
 - $x = 4$

輸入格式

輸入第一行為一個正整數 $|s|$ 。

輸入第二行為一個得字串 s 。

輸入第三行為一個正整數 Q 。

對於所有正整數 i 滿足 $1 \leq i \leq Q$ ，輸入第 $i + 3$ 行為兩個正整數 L_i, R_i ，兩數之間以一個空格分開。

輸出格式

對於所有正整數 i 滿足 $1 \leq i \leq Q$ ，如果劉得徵無法找到滿足條件的正整數 x, y ，則輸出第 i 行為 -1，否則輸出第 i 行為兩個正整數，第一個正整數是 $y - x + 1$ 的最小值，第二個正整數是滿足所有條件的正整數 x, y 的組數。

資料範圍

- $1 \leq |s| \leq 10^6$

- s 是一個得字串
- $1 \leq Q \leq 10^6$
- $1 \leq L_i \leq R_i \leq |s|$

子任務

- 子任務 1 沒有其他限制 (10分)

測試範例

輸入範例 1

```
3
499
2
1 2
2 3
```

輸出範例 1

```
-1
2 1
```

範例說明

$i = 1$ 時，由於 4, 9, 49 都是完全平方數，所以找不到滿足條件的 x, y 。

$i = 2$ 時，由於 4, 9 都是完全平方數，但是 99 不是，所以 $x = 2, y = 3$ 是唯一的選擇。

4_羽球比賽(Badminton_Game)

(10 分)

時間限制: 1 second

記憶體限制: 256 MB

題目敘述

小桃與小櫻很喜歡打羽球，時不時會約出來比賽。羽球比賽會進行數輪，在第一輪開始前，雙方的分數都是零分。每一輪會由其中一方發球，發球完經過一連串的對打後會決定出哪一方獲勝，每輪獲勝的一方會獲得一分。詳細發球的規則為：

- 第一輪會由小桃發球，接下來每一輪由前一輪的獲勝方發球
- 若發球方當前的得分為偶數，則要在球場的右邊發球，否則要在左邊發球

舉例來說，以下為一個進行四輪的羽球比賽的例子：

輪數	發球方	發球時小桃得分	發球時小櫻得分	哪邊發球	該輪獲勝方
1	小桃	0	0	右	小桃
2	小桃	1	0	左	小櫻
3	小櫻	1	1	左	小櫻
4	小櫻	1	2	右	小桃

兩人昨天進行了一場 N 輪的羽球比賽，但到了今天，她們卻發現她們忘記最後的比分，只記得每一輪是在左邊發球還是右邊發球。小桃想要知道在這樣的情況下，她最高獲得的分數是多少，你能幫幫她嗎？因為有可能發球的資訊也是錯誤的，如果不存在一種合法的比賽過程可以得出這樣的發球資訊，也請回報這個錯誤。

輸入格式

輸入第一行有一個正整數 T ，代表接下來會有 T 筆測資資料。

每一筆測試資料會有兩行，其中第一行會有一個正整數 N ，代表羽球比賽有幾輪。

第二行會有一個長度為 N 的字串 S ，若 S_i 為 L 則代表第 i 輪是在左邊發球，而若 S_i 為 R 則代表第 i 輪在右邊發球。

輸出格式

請輸出 T 行，若第 i 筆測試資料的發球資訊不合法則在第 i 行輸出 -1，否則輸出小桃可以獲得的最高分數。

資料範圍

- $1 \leq T \leq 10000$
- $1 \leq N \leq 5 \times 10^5$
- $|S| = N$

- $S_i \in \{L, R\}$
- 所有測試資料的 N 的總和不超過 5×10^5

測試範例

輸入範例 1

```
4
4
RLLR
3
LRR
2
RR
7
RLRLRLR
```

輸出範例 1

```
3
-1
-1
7
```

範例說明

在第一筆範例測資中，若每輪的贏家分別為「小桃、小櫻、小桃、小桃」可以得出 RLLR 的發球結果。可以證明沒有比這更好的結果，因此答案為 3。

在第二筆範例測資中，因為第一輪一定會在右邊發球，所以不合法。

5_程式競賽是想像的世界 (Competitive_Programming_Is_A_World_of_Imagination)

(6分 / 9分)

時間限制: 1 second

記憶體限制: 256 MB

題目敘述

小李、小明跟小奕今年組隊參加 YTP 程式挑戰營，但最近他們的練習表現一直不太好。為了加深他們的團隊默契並轉換心情，他們決定一起去陽明山上走走。晚上，他們坐在草地上一起看星星。想到各路高手那麼多，再加上最近的練習狀況不好，小李喃喃自語：「我根本就不能想像我在 YTP 程式挑戰營得獎的樣子。」

這時，小波忽然從一旁出現，說：「程式競賽是想像的世界，想像不到的事情是做不到的！」

小明覺得他說的話非常有道理，於是提議大家一起來練習想像力。他看著天空上的 N 顆肉眼可見的星星，發現它們恰好是一個凸多邊形的所有頂點！更具體一點的說，整個天空可以被視為是一個二維平面，每顆星星都是這個二維平面上的一個點，第 i 顆星星的座標是 $p_i = (x_i, y_i)$ ，並且存在一個方式把所有星星重新編號以後，在 p_1, p_2 之間畫線、 p_2, p_3 之間畫線、……、 p_{N-1}, p_N 之間畫線、 p_N, p_1 之間畫線，會得到一個 N 個頂點的凸多邊形。

在小明說出了這個觀察之後，小李又突發奇想，用不同順序把星星連起來就可以得到不同形狀，不如用星星來畫星星吧！以下是小李想要的畫星星的方式：

1. 把 N 顆星星分成 $k \geq 1$ 組，每組至少包含 3 顆星星。一顆星星要在恰好一組之中。
2. 對於每一組之中的星星，將它們按照某種方式排列之後，在第 1 顆星星和第 2 顆星星之間連線、第 2 顆和第 3 顆之間連線、……、最後一顆再跟第 1 顆連線。

白話的說，就是小李想要畫 k 個筆畫，每個筆畫都從某個星星開始，經過一些星星之後，再回到起點的星星，每個星星都恰好被畫到一次（同為起終點的那個星星只算一次）。

很明顯的，小李會畫 N 條線段，小李希望：

- 對於任意一條線段，除了它自己和它的相鄰兩條線段之外，同一筆畫裡的其他線段都和它相交。一條線段的相鄰兩條線段是指那一筆畫裡的上一條和上一條線段，第一條線段的上一條線段是最後一條線段、最後一條線段的下一條線段是第一條線段。
- 對於任意一條線段，任意一個不是它所在的筆畫裡，都有至少一條線段和它相交。

雖然小奕不太喜歡小波，不知道從哪跑出來還偷聽他們說話，但搞不好 YTP 程式挑戰營就考這個，所以請你幫他們畫星星。

輸入格式

第一行有一個奇數 N ，代表星星的數量。

接下來有 N 行，其中第 i 行有兩個整數 x_i, y_i ，代表第 i 顆星星的座標。

輸出格式

第一行輸出一個整數 k ，代表星星被分成幾組。

接下來輸出 k 行，其中第 i 行代表第 i 組星星，先輸出一個整數 $t_i \geq 3$ ，代表這組星星有幾顆星星，接下來輸出 t_i 個 1 到 N 之間的整數 $s_{i,1}, s_{i,2}, \dots, s_{i,t_i}$ ，代表這組之中的星星編號，按照把星星連起來的順序輸出。也就是說，這代表要把星星 $s_{i,1}, s_{i,2}$ 相連， $s_{i,2}, s_{i,3}$ 相連，……， $s_{i,t_i}, s_{i,1}$ 相連。

有多種畫星星的方法時，輸出任意一種滿足題目要求的皆可。

資料範圍

- $3 \leq N \leq 1000$
- N 是奇數
- $\forall 1 \leq i \leq N, |x_i|, |y_i| \leq 10^9$
- 星星剛好是一個凸多邊形的所有頂點
- 星星不一定會按照凸多邊形的頂點順序給出

子任務

- 子任務 1 滿足星星會按照它們形成的凸多邊形的頂點順序，以逆時針順序給出 (6 分)
- 子任務 2 沒有額外限制 (9 分)

測試範例

輸入範例 1

```
9
6 8
6 4
10 2
14 2
18 4
20 8
18 12
12 14
8 12
```

輸出範例 1

```
1
9 1 6 2 7 3 8 4 9 5
```

輸入範例 2

```
3  
4 9  
8 1  
1 1
```

輸出範例 2

```
1  
3 1 2 3
```

輸入範例 3

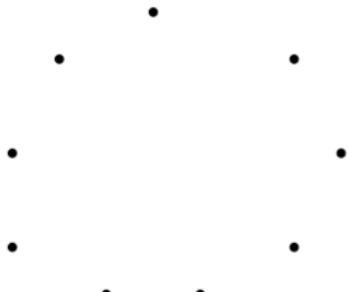
```
5  
4 1  
19 14  
17 17  
8 0  
13 0
```

輸出範例 3

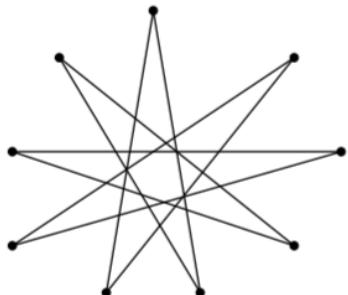
```
1  
5 1 5 3 4 2
```

範例說明

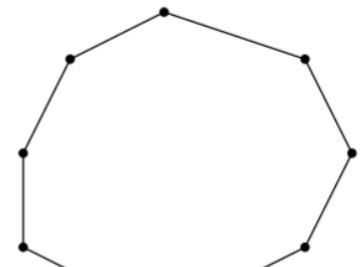
下圖是輸入範例 1 的說明，右側列出了一些「相鄰線段」的例子。(b) 是範例輸出 1 代表的畫星星的方式。



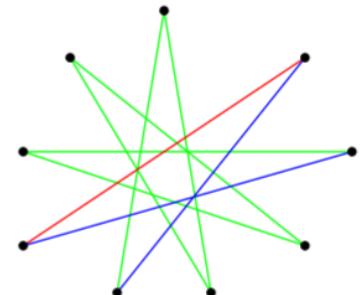
(a) 天上的星星



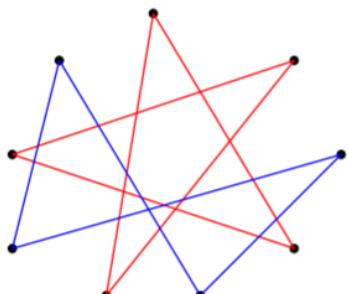
(c) 一種畫星星的方式



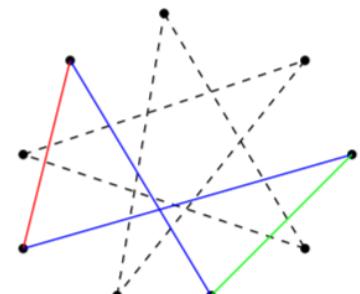
(b) 把星星連成凸多邊形的樣子



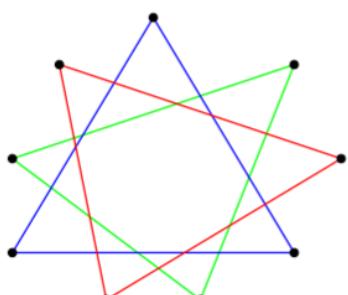
(d) 兩條藍色線段是紅色線段的相鄰線段，紅色線段要和所有綠色線段相交（不管起點星星是哪一個，任何線段的相鄰線段都一樣！）



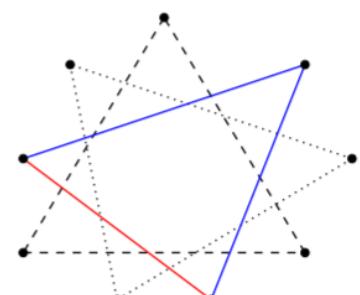
(e) 一種不正確的畫星星的方式，星星被分成兩組



(f) 兩條藍色線段是紅色線段的相鄰線段，紅色線段應該要和綠色線段相交。



(g) 另一種畫星星的方式，星星被分成三組



(h) 兩條藍色線段是紅色線段的相鄰線段

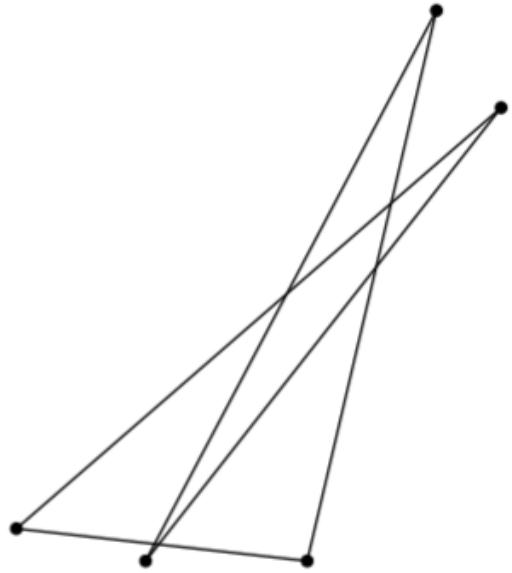
以下也是正確的答案，對應到上圖的 (g)。

```
3
3 1 4 7
3 2 5 8
3 3 6 9
```

下圖是輸出範例 2 和 3 畫出來的樣子：



(a) 範例 2



(b) 範例 3

6_蛋餅愛拉麵(Omelet_Loves_Ramen)

(5 分 / 10 分)

時間限制: 2 seconds

記憶體限制: 256 MB

題目敘述

愛拉麵的蛋餅打算要開一間拉麵店，並在經過長時間的籌備後即將開幕！這裡將會販賣由他精心研發的 M 種不同的拉麵。蛋餅會使用鍋子來煮拉麵，由於蛋餅的拉麵口味都偏重，若鍋子煮完一種拉麵沒有洗就去煮另外一種，口味就會混合而導致拉麵變不好吃。為了避免這種情況發生，當蛋餅需要煮第 x 種拉麵時，他會採用以下方法決定要用哪一個鍋子煮：

- 若有一個鍋子上一碗煮的拉麵是第 x 種，則他會直接用該鍋子煮
- 否則若有至少一個沒使用過的鍋子，他會直接拿其中一個煮
- 否則他會把最久沒使用的鍋子拿去洗乾淨，再用那個鍋子煮

舉例來說，若蛋餅有 2 個鍋子而他依序要煮第 1, 2, 3, 2, 3, 1 種拉麵，則：

拉麵種類	使用第幾個鍋子	要不要洗	說明
1	1	No	第 1 個鍋子沒有使用過
2	2	No	第 2 個鍋子沒有使用過
3	1	Yes	第 1 個鍋子最久沒被使用
2	2	No	第 2 個鍋子上一碗煮的是第 2 種
3	1	No	第 1 個鍋子上一碗煮的是第 3 種
1	2	Yes	第 2 個鍋子最久沒被使用

蛋餅已經預測出了拉麵店開幕這天會依序有 N 個客人到來，而第 i 個客人會點第 a_i 種拉麵，請幫幫蛋餅對於所有 K 從 1 到 M ，計算若他有 K 個鍋子則他會需要洗幾次鍋子。

輸入格式

輸入第一行有兩個正整數 N, M ，分別代表客人的數量以及拉麵的種類。

輸入第二行有 N 個正整數 a_1, a_2, \dots, a_N ，第 i 項代表第 i 個客人點的拉麵種類。

輸出格式

請輸出一行，該行有 M 個數字，第 i 個代表若蛋餅有 i 個鍋子他需要洗幾次鍋子。

資料範圍

- $1 \leq M \leq N \leq 5 \times 10^5$

- $1 \leq a_i \leq M$

子任務

- 子任務 1 滿足 $N \leq 2000$ (5 分)
- 子任務 2 沒有額外限制 (10 分)

測試範例

輸入範例 1

```
6 3  
1 2 3 2 3 1
```

輸出範例 1

```
5 2 0
```

輸入範例 2

```
6 3  
1 1 3 2 2 1
```

輸出範例 2

```
3 2 0
```

輸入範例 3

```
9 9  
8 6 4 1 9 7 5 3 2
```

輸出範例 3

```
8 7 6 5 4 3 2 1 0
```

輸入範例 4

```
20 7  
4 5 2 7 2 7 1 3 1 3 4 2 4 1 5 1 2 5 4 1
```

輸出範例 4

19 12 10 5 3 0 0

範例說明

第二筆範例測資 $K = 2$ 時的說明如下：

拉麵種類	使用第幾個鍋子	要不要洗	說明
1	1	No	第 1 個鍋子沒有使用過
1	1	No	第 1 個鍋子上一碗煮的是第 1 種
3	2	No	第 2 個鍋子沒有使用過
2	1	Yes	第 1 個鍋子最久沒被使用
2	1	No	第 1 個鍋子上一碗煮的是第 2 種
1	2	Yes	第 2 個鍋子最久沒被使用

因此蛋餅共會洗 2 次鍋子。

7_鴨鴨收集 (Duck_Collection)

(15 分)

時間限制: 1 second

記憶體限制: 256 MB

題目敘述

YTP 國上有 n 個池塘，每一個池塘內有無限多的鴨鴨，同一個池塘內的鴨鴨種類都是一樣的。

YTP 國的鴨鴨管理者「呱呱」在某一天決定要來玩一個收集鴨鴨的遊戲，這一個遊戲的目標是要收集到所有種類的鴨鴨，而由於有 n 個池塘，因此鴨鴨的種類當然也是 n 種。

這個遊戲每回合都可以發起一次請求，每一次請求可以選擇兩個池塘（可以重複）並在這兩個池塘中各拿一隻鴨鴨出來，而如果你選擇的池塘分別為 (i, j) ，那麼這一次請求的花費就會是 $c_{i,j}$ ，此外，這項遊戲注重先後順序，因此先拿池塘 i 再拿池塘 j 可能會與倒過來拿得到不一樣的花費，也就是說 $c_{i,j} = c_{j,i}$ 不一定成立。

我們的目標除了收集到所有種類的鴨鴨之外，我們也希望花費越少越好，請你設計一個程式計算出最小花費。

輸入格式

第一行輸入一個正整數 n ，代表鴨子的種類數以及池塘的個數。

接下來共有 n 行，每一行依序輸入 n 個正整數，第 i 行的第 j 正整數為 $c_{i,j}$ ，代表你在一回合中先拿了鴨子 i 再拿了鴨子 j 的花費。

輸出格式

請輸出一行，包含一個整數，代表「呱呱」要收集到所有鴨子所需要的最小花費。

資料範圍

- $1 \leq n \leq 18$ 。
- $1 \leq c_{i,j} \leq 10^9$ 。

測試範例

輸入範例 1

```
3
10 9 10
4 5 10
7 5 1
```

輸出範例 1

```
5
```

輸入範例 2

```
3
1 10 10
10 1 10
10 10 1
```

輸出範例 2

```
3
```

輸入範例 3

```
3
1 1 1
10 10 10
10 10 10
```

輸出範例 3

```
2
```

範例說明

在範例 1 中，我們各回合的拿法如下：

- 第一回合拿取第 2、1 種鴨子，花費為 4。
- 第二回合拿取兩隻第 3 種的鴨子，花費為 1。
加起來後得到了總花費為 5。

8_喜多風暴(Kita_Storm)

(15分)

時間限制: 1 second

記憶體限制: 512 MB

題目敘述

「台大的網路管理與系統管理 (NASA) 是一個積極新創、人才齊全、內容充實、作業特色鮮明的課程。在資訊系上具有重要影響力，在多個領域具有非常前瞻的科技實力，擁有世界一流的電腦設備與師資力量，各種排名均位於全球前列，並且擁有公開透明的討論管道、各種進修資源，以及服務學習抵免可以申請，歡迎大家來修 NASA。」

看完這份真摯感人的招募文的你，決定去參加 NASA 課程，而在某一次作業中你遇到了這一題：

「系上某實驗室的某台電腦居然大量廣播夾帶喜多能量的喜多封包，造成喜多風暴，使得整個系館的人都要被淨化成社牛了（其實好像也挺不錯的？），波奇醬一接收到喜多能量就自動進入自卑模式，不管你怎麼叫她都沒有回應，整個網管團隊只剩下你一人，好巧不巧，喜多風暴突然停止了，聰明的你能夠找出失控的電腦，阻止喜多風暴再次發生，拯救波奇醬和整個系館的人嗎！」 -取自 NASA 2023 HW2

在作業的 `.pka` 檔裡，系上實驗室的網路架構可以被視為一棵有根樹，節點的編號由 1 到 N ，而節點 1 是根節點。其中葉節點是實驗室的電腦，其他節點都是負責維護連線的交換機 (Switch)。失控的電腦編號為 x ，IP 為 `192.168.0.53`，而你在一個交換機上面可以執行以下指令

```
ping 192.168.0.53
show arp
show mac address-table
```

根據這些資訊，你可以知道失控的電腦在交換機的哪一個連接埠 (port) 上。換句話說，在節點 u 上，你會得到一個與 u 相鄰的點 a_u ，使得 x 到 u 的簡單路徑會經過 a_u 。壞心的助教為了增加作業難度，鎖住了某些節點的權限，讓你無法在這些節點上執行指令。但是，聰明的你發現這樣題目就不一定有唯一答案了。為了證明你的論點，你決定寫一個程式，找出所有可能是失控電腦的節點編號。注意，失控電腦一定是葉子節點。

輸入格式

第一行有一個數字 N ，代表樹上節點數量。

接下來 $N - 1$ 行每行有兩個數字 u, v ，代表樹邊的節點編號。樹邊本身是無向的。

最後一行有 N 個數字 a_1, a_2, \dots, a_N 。如果 $a_i = 0$ ，代表你沒有節點 i 的權限，否則 a_i 代表失控節點與節點 i 路徑中會經過的節點。

輸出格式

第一行輸出一個整數 k ，代表可能是失控電腦的點數。

第二行輸出 k 個整數 b_1, b_2, \dots, b_k ，代表每個失控節點的編號。請將 b_1, b_2, \dots, b_k 由小到大輸出。

資料範圍

- $2 \leq N \leq 2 \times 10^5$
- $1 \leq u, v \leq N$

- $0 \leq a_i \leq N$ ，若 $a_i \neq 0$ ，則節點 a_i 一定與節點 i 相鄰。
- 如果節點 p 是葉節點， $a_p = 0$
- 保證 a_i 的資訊不會產生矛盾

子任務

- 本題只有一個子任務

測試範例

輸入範例 1

```
7
1 2
1 3
2 4
2 5
3 6
3 7
2 4 1 0 0 0 0
```

輸出範例 1

```
1
4
```

輸入範例 2

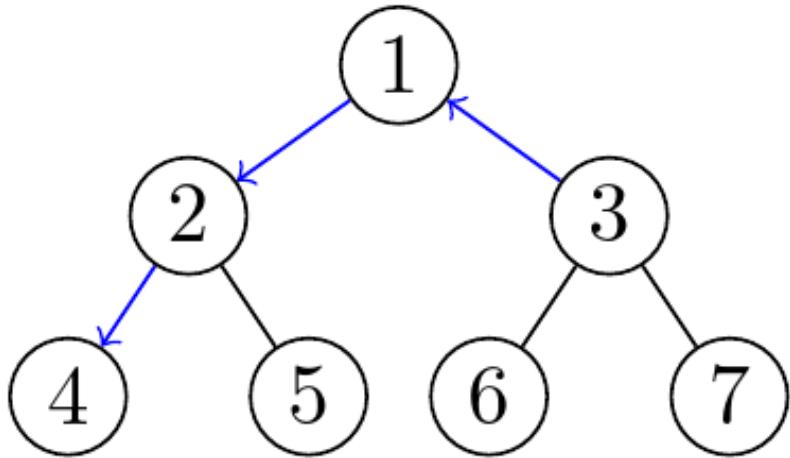
```
7
1 2
1 3
2 4
2 5
3 6
3 7
0 0 1 0 0 0 0
```

輸出範例 2

```
2
4 5
```

範例說明

範例一的圖示如下：



其中，藍色箭頭分別代表有辦法執行指令的交換機中，指向失控節點的方向。因此，只有編號為 4 的電腦可能是失控的。

9_迴文(Palindromes)

(20分)

時間限制: 1 second

記憶體限制: 256 MB

題目敘述

Kevin 最近喜歡上了迴文，也就是正著念和反著念都一樣的字串，例如 `aba`、`abba` 和 `b` 都是迴文，但 `ba`、`abc` 和 `abaa` 都不是。回過神來，他不管看見什麼東西都會嘗試想出一個跟迴文有關的問題。果不其然，這天 Kevin 就收到了一個 `01` 字串，他馬上就想到了一個新的問題：

「每次詢問一個區間，是否能夠快速回答這個區間內有多少種長相不同的非空迴文子序列呢？」

Kevin 一時之間不知道怎麼解決這個問題，因此他找上了同樣喜歡演算法的你，你能夠幫他解決他想到的問題嗎？

註：一個序列 S 的子序列，是指從 S 中刪除若干項（可以沒有）後，剩餘的項順序不變所得到的序列。

輸入格式

第一行包含兩個正整數 N, Q ，代表 Kevin 收到的 `01` 字串長度，以及詢問的數量。

第二行有一個長度為 N 的 `01` 字串 $s_1 s_2 \dots s_N$ ，代表 Kevin 收到的字串。

接下來 Q 行，第 i 行有兩個正整數 l_i, r_i ，代表第 i 筆詢問希望知道字串的區間 $[l_i, r_i]$ 中有多少種長相不同的迴文子序列。

輸出格式

輸出 Q 行，第 i 行代表第 i 筆詢問的答案。由於答案可能很大，對於每筆詢問請輸出答案除以 998244353 的餘數。

資料範圍

- $1 \leq |S| = N \leq 3000$
- $1 \leq Q \leq 3 \times 10^5$
- $s_i \in \{0, 1\}$
- $1 \leq l_i \leq r_i \leq N$

測試範例

輸入範例 1

```
6 4
011011
1 4
4 6
3 5
1 6
```

輸出範例 1

```
6
3
4
10
```

範例說明

在範例 1 中，整個字串中長相不同的迴文子序列有 0、1、00、11、010、101、111、0110、1111、11011 共十種，這也是第四筆詢問的答案。除此之外，每筆詢問也可以在對應的區間內找到對應種類量的迴文子序列。

10_有限井字遊戲(Expiring Tic-Tac-Toe)

(20分)

時間限制: 1 second

記憶體限制: 256 MB

題目敘述

Alice 是一個非常喜歡打日麻的人，她時不時就會從身邊找齊四個人來湊一桌四人的半莊。這一天，Alice 一如往常的在找人打日麻，在她找好三個人後，她跑去找了 Bob 來一起打日麻。不過，由於 Bob 在準備期末考，因此拒絕了 Alice 的邀請。好不容易只差一個人就能湊齊四人半莊，Alice 不甘心放棄，於是一直纏著 Bob 不放。禁不住 Alice 的死纏爛打，Bob 於是決定與 Alice 下一盤棋，只有 Alice 贏了，他才會答應陪 Alice 打日麻。

Bob 決定要跟 Alice 下一場「有限井字遊戲」，遊戲玩法跟一般的井字遊戲很像。由 Alice 先手，兩個人輪流在 3×3 的棋盤上放上自己的棋子，先讓自己的棋子連成一直行、橫排、或斜直線的玩家獲勝。不過與一般井字遊戲不同的點在於，每個被玩家所放置的棋子都會在該玩家的三個回合後失效，也就是說，每個玩家在第 x 回合所放置的棋子都會在第 $x + 3$ 回合放置棋子時失效。特別地，第 x 回合和第 $x + 3$ 回合不可把棋子放在相同格子上。此外，如果遊戲到第 45510 回合還沒結束，則會被判定為和局。特別的，在某一回合判斷是否有玩家勝出前，如果有棋子將要失效，則棋子先失效，才進行勝負的判定。

Alice 和 Bob 在下了幾回合後有點累了，他們不想再花時間認真思考策略來把棋下完。因此，他們找到你，希望你能根據他們下出來的殘局幫忙判斷，如果兩人接下來都以最佳策略下棋，且 Alice 會下下一步棋，則棋局結果為何？

輸入格式

輸入有三行，每行包含三個整數。

第 i 行的第 j 個整數 $a_{i,j}$ 代表目前棋盤上第 i 行第 j 列的狀態。

$a_{i,j} > 0$ 代表該格為 Alice 的棋子，且會在 Alice 接下來下第 $a_{i,j}$ 個棋子時失效；

$a_{i,j} < 0$ 則代表該格為 Bob 的棋子，且會在 Bob 接下來下第 $-a_{i,j}$ 個棋子時失效；

$a_{i,j} = 0$ 則代表該沒有棋子。

保證輸入的棋盤還沒產生贏家，且尚未進行超過十回合。

輸出格式

輸出一個字串代表該盤面在雙方都使用必勝策略後的結果：

- "Alice" (不含引號) 代表 Alice 會獲勝
- "Bob" (不含引號) 代表 Bob 會獲勝
- "Draw" (不含引號) 代表雙方會和局。

資料範圍

- $-3 \leq a_{i,j} \leq 3$
- 保證 $1, 2, 3, -1, -2, -3$ 在所有 a_{ij} 中各出現一遍

測試範例

輸入範例 1

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

輸出範例 1

```
Alice
```

輸入範例 2

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

輸出範例 2

```
Bob
```

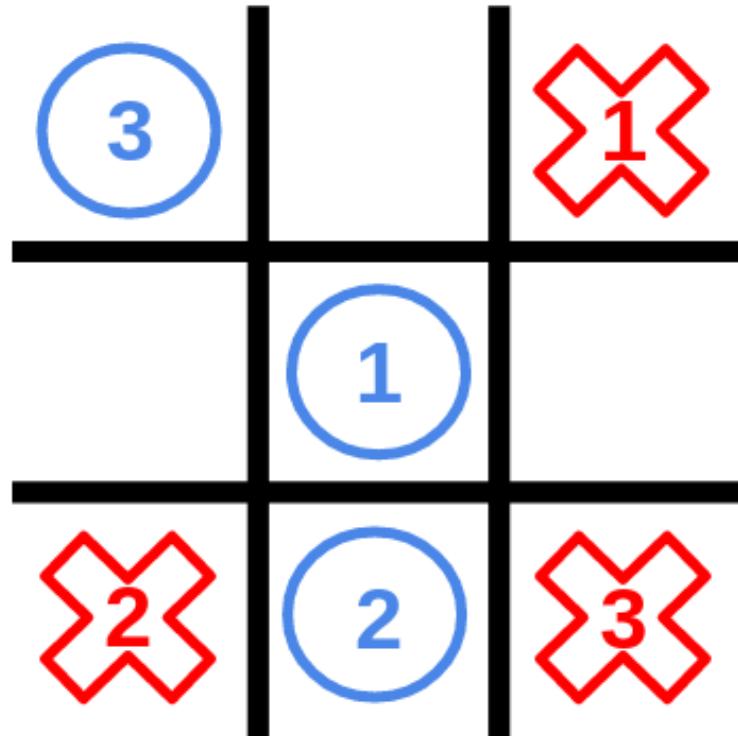
輸入範例 3

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

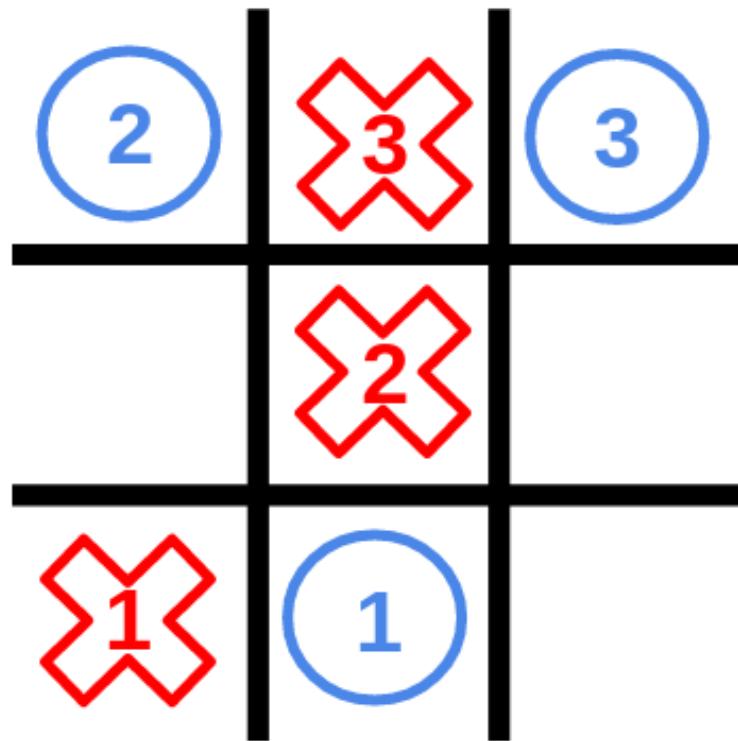
輸出範例 3

```
Draw
```

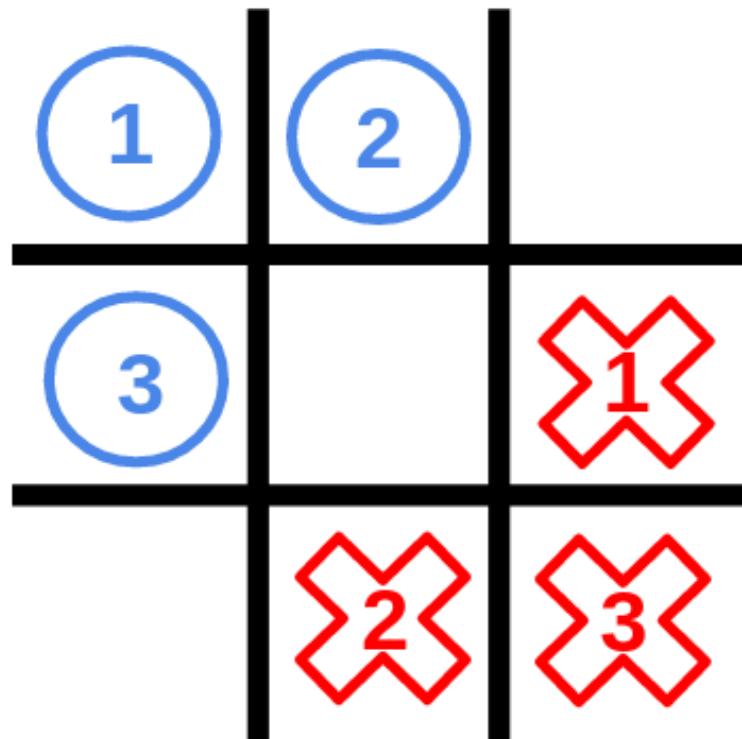
範例說明



在範例一中，可以看到只要 Alice 將旗子下在中上的位置，下回合必定能將棋字下在左上獲勝獲勝。



在範例二中，無論 Alice 下在什麼位置，Bob 必定能將棋子下在中下獲勝。



在範例三中，雙方每回合都要阻止對方下回合勝出，從而形成和局。

11_Stairs

(6 分/6 分/8 分)

時間限制: 1 second

記憶體限制: 1024 MB

題目敘述

NJ 有 N 個階梯，編號從 $1 \sim N$ 。

每個階梯都由一些柱子組成，每根柱子都有自己的高度 y ，跟它所處的位置 x 。

我們很隨意地將這些柱子形成的集合稱作該階梯的「柱集」。

每個階梯都有無限長，其中，這個階梯中座標為 k 的位置的高度為，該階梯的柱集中，所有 $x \leq k$ 的柱子的最大 y 值。

一開始，每個階梯的柱集皆為空。

NJ 想瘋狂擺弄他的階梯們，具體地說，NJ 會修改這些階梯們的柱集，NJ 能做的操作如下：

- 1 $id\ x\ y$
- 2 $a\ b$
- 3 $id\ k$

對於第 1 種操作，NJ 會在編號為 id 的階梯的柱集裡加入一個新的高 y ，位於 x 的柱子。

對於第 2 種操作，NJ 會將編號 b 的柱集中的柱子全數移到編號 a 的階梯的柱集中。

對於第 3 種操作，NJ 想知道編號 id 的階梯，位置 k 的高度是多少。

輸入格式

輸入的第一行會包含兩個正整數 N 、 Q 。

接下來的 Q 行，每行包含題敘所述之其中一種操作。

輸出格式

對每個第三種操作，輸出 NJ 想知道的高度，及一個換行字元。

如果該 NJ 想知道的集合是空的，請輸出 -1 。

資料範圍

- $1 \leq N, Q \leq 200\,000$ 。
- $op \in \{1, 2, 3\}$ 。
- $1 \leq x, y, k \leq 1\,000\,000\,000$ 。
- $1 \leq a, b, id \leq N$ 。
- $a \neq b$ 。

子任務

- 子任務 1 滿足 $N = 1$ 。
- 子任務 2 滿足 $N \leq 200$ 。
- 子任務 3 無額外限制。

測試範例

輸入範例 1

```
4 7
1 3 424849053 592732474
1 1 787225144 158123046
2 4 3
3 4 534558786
2 2 4
3 2 806668402
3 3 679670154
```

輸出範例 1

```
592732474
592732474
-1
```

輸入範例 2

```
10 13
1 7 205588596 580018058
2 5 7
1 6 511038935 266614169
1 6 486422924 865076365
1 10 445734405 877372041
3 10 843207541
2 4 9
2 7 8
1 1 708876592 402376498
2 9 5
2 2 6
2 6 5
3 6 323976805
```

輸出範例 2

```
877372041
-1
```

範例說明

範例1中：

第一個詢問需回答第4個柱集中，座標小於等於534558786的柱子裡高度最高的，為操作一放入的柱子，高度為592732474。

第二個詢問需回答第2個柱集中，座標小於等於806668402的柱子裡高度最高的，為操作一放入的柱子，高度為592732474(此時此柱子被操作五移動到第2個柱集中了)

第三個詢問，因此時第3個柱集中不存在座標小於等於679670154的柱子，因此回答-1。

12_拍照 (Taking_Photos)

(4 分/16 分)

時間限制: 2 seconds

記憶體限制: 1024 MB

題目敘述

「外賣～怎麼吃呢？我們今天來吃外賣～今天吃什麼外賣呢？咖哩拌飯～咖哩拌飯～」

拍照是我的習慣，外賣是我的日常。每天我都會拿著相機記錄我的外賣人生。久而久之，相機的壽命也即將結束。在它僅存的「機生」中，只能再拍攝 K 張照片。現在已知城市中有 N 個咖哩拌飯，身為咖哩拌飯愛好者，我想一個不漏的拍攝到所有的咖哩拌飯。這些咖哩拌飯所在的位置可以看成一個二維平面，相機每次能夠拍攝到一塊矩形區域，其中矩形的四個角落都是整數座標。為了要讓咖哩拌飯看起來更好吃，我打算近距離拍攝這些咖哩拌飯，但這也會減少拍攝到的面積。定義一個點的面積為 0。現在我想知道，在只能拍攝 K 次且拍到所有咖哩拌飯的情況下，照片覆蓋的總面積最小是多少？

輸入格式

輸入的第一行包含兩個正整數 N 和 K ，代表咖哩拌飯的數量和可以拍照的次數。

接下來一共有 N 行，每行有兩個正整數 X_i 、 Y_i ，代表咖哩拌飯的座標。

輸出格式

輸出一個整數，代表最小照片覆蓋的總面積。

資料範圍

- $1 \leq N \leq 3000$ 。
- $1 \leq K \leq N$ 。
- $1 \leq X_i, Y_i \leq 10^7$ ($1 \leq i \leq N$) 。
- $X_i < X_{i+1}$ 且 $Y_i < Y_{i+1}$ ($1 \leq i \leq N - 1$) 。

子任務

- 子任務 1 滿足 $N \leq 500$ 。
- 子任務 2 無額外限制。

測試範例

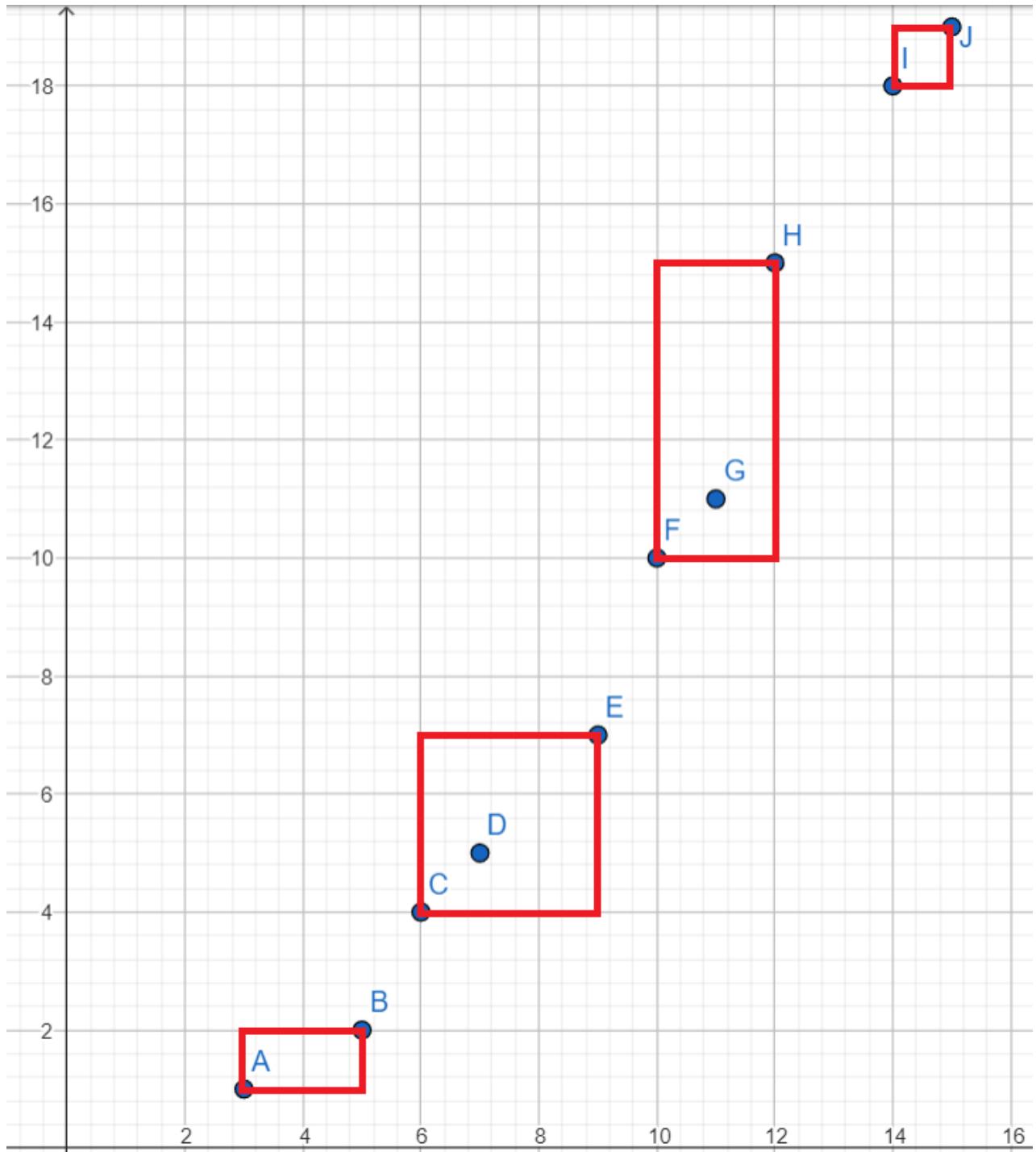
輸入範例 1

```
10 4
3 1
5 2
6 4
7 5
9 7
10 10
11 11
12 15
14 18
15 19
```

輸出範例 1

22

範例說明



最小覆蓋面積方式如上圖，面積總和為 $2 \times 1 + 3 \times 3 + 2 \times 5 + 1 \times 1 = 22$ 。

13_神聖聯盟 (Sacred_Alliance)

(3 分/3 分/14 分)

時間限制: 5 seconds

記憶體限制: 1024 MB

題目敘述

這個世界上有 N 個國家，國家之間可能相互連結形成道路。如果國家 u 和國家 v 之間有一條道路，我們就稱它們是鄰國。需要注意的是，一個國家不會是自己的鄰國。

我們保證所有國家之間都能透過道路相互聯通。

其中有 T 個國家被怪物統治，剩下的 $N - T$ 個國家由人類組成。

多個人類的國家可以聯合成一個神聖聯盟。對於一個神聖聯盟 S ，給定一個數字 p ，如果對於神聖聯盟中的任何一個國家 x ，都滿足 $\frac{|N(x) \cap S|}{|N(x)|} \geq p$ ，我們就稱神聖聯盟 S 的強度為 p 。這裡 $N(x)$ 代表與國家 x 相連的鄰國所形成的集合。

作為人類方的一員，你想知道人類方能夠組成多強的神聖聯盟。

輸入格式

輸入的第一行包含兩個正整數 N 、 M 、 T ，分別代表國家的總數量、道路的數量、由怪物所統治的國家數量。

輸入的第二行包含 T 個正整數 a_1, a_2, \dots, a_T ，代表著由怪物所統治的國家的編號。

接下來的 M 行，每一行包含兩個正整數 u_i, v_i ，代表國家 u_i 和國家 v_i 之間有一條道路相連。

輸出格式

若最強的神聖聯盟的強度為 $\frac{p}{q}$ ，其中 p 、 q 是正整數且 $\gcd(p, q) = 1$ ，則請輸出 $p \times q^{-1} \bmod 10^9 + 7$ 。

資料範圍

- $2 \leq N \leq 10^5$ 。
- $N - 1 \leq M \leq \min\left(\frac{N(N-1)}{2}, 2 \times 10^5\right)$ 。
- $0 \leq T < N$ 。
- $1 \leq u_i, v_i \leq N$ for $i = 1, 2, \dots, N$ 。
- $u_i \neq v_i$ for $i = 1, 2, \dots, N$ 。

子任務

- 子任務 1 滿足 $N \leq 15$ 。
- 子任務 2 滿足 $N \leq 26$ 。
- 子任務 3 無額外限制。

測試範例

輸入範例 1

```
3 2 1  
3  
1 2  
2 3
```

輸出範例 1

```
500000004
```

輸入範例 2

```
4 6 2  
3 4  
1 2  
1 3  
1 4  
2 3  
2 4  
3 4
```

輸出範例 2

```
33333336
```

輸入範例 3

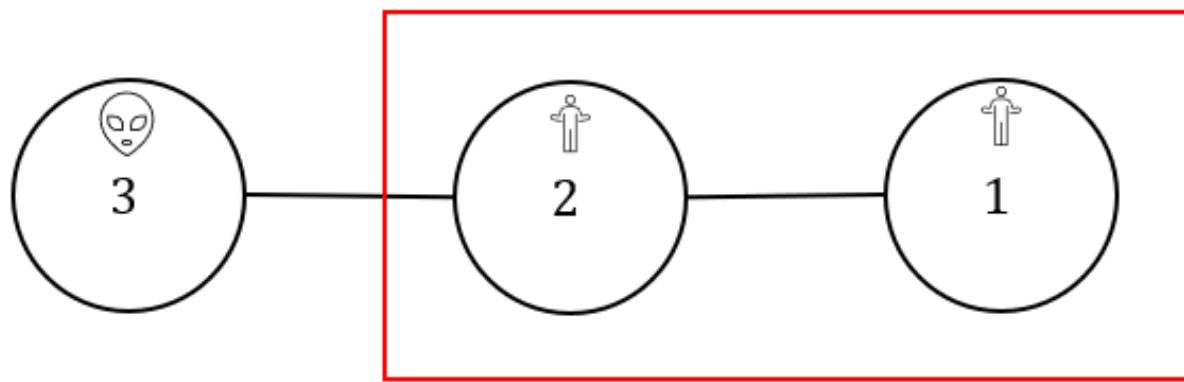
```
6 6 3  
1 2 3  
1 4  
2 4  
3 4  
4 5  
4 6  
5 6
```

輸出範例 3

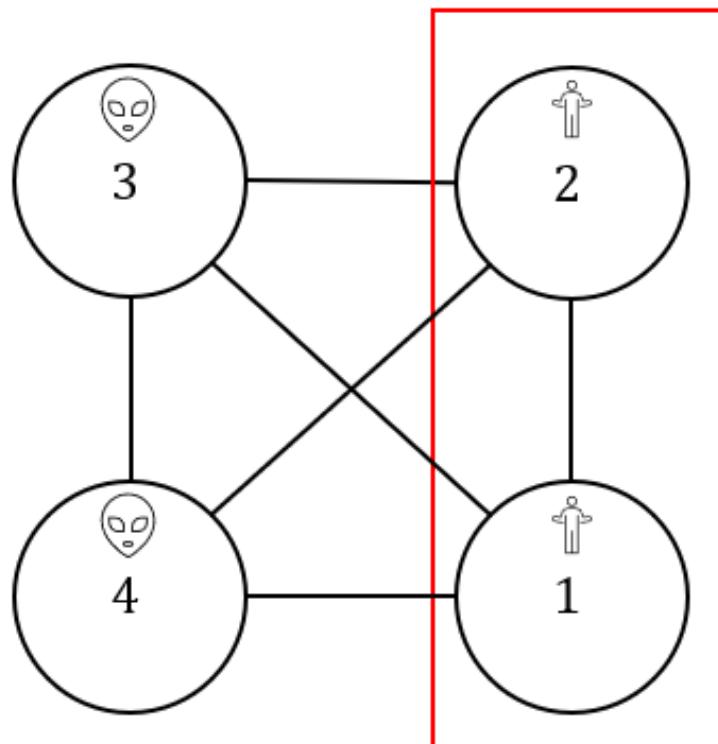
```
500000004
```

範例說明

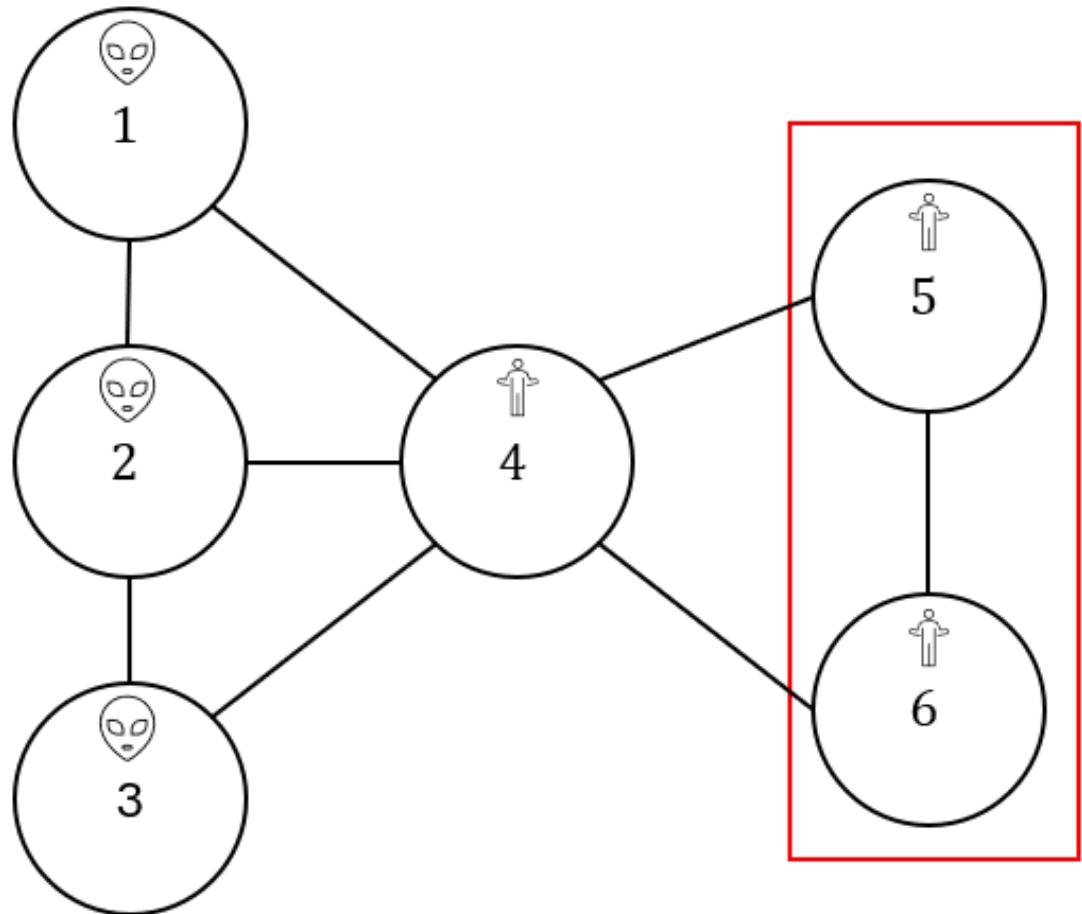
在範例 1 中，國家 1 和國家 2 組成的神聖聯盟中，國家 1 有 $\frac{1}{1}$ 比例的鄰國在聯盟中，國家 2 有 $\frac{1}{2}$ 比例的鄰國在聯盟中，因此聯盟的強度為 $\frac{1}{2}$ 。



在範例 2 中，國家 1 和國家 2 組成的神聖聯盟中，國家 1 有 $\frac{1}{3}$ 比例的鄰國在聯盟中，國家 2 有 $\frac{1}{3}$ 比例的鄰國在聯盟中，因此聯盟的強度為 $\frac{1}{3}$ 。



在範例 3 中，國家 4 和國家 5 組成的神聖聯盟中，國家 1 有 $\frac{1}{2}$ 比例的鄰國在聯盟中，國家 2 有 $\frac{1}{2}$ 比例的鄰國在聯盟中，因此聯盟的強度為 $\frac{1}{2}$ 。



14_凸包面積 (Convex_Hull_Area)

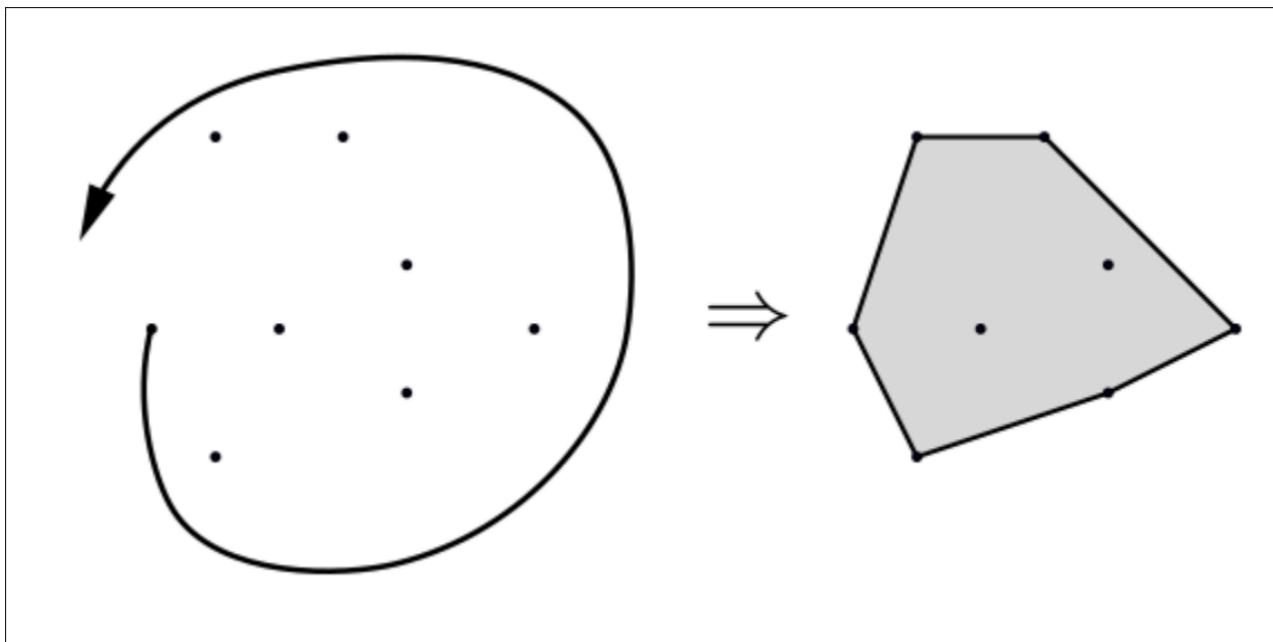
(20 分)

時間限制: 1 second

記憶體限制: 512 MB

題目敘述

「凸包」是一個在計算幾何問題中常使用到的工具，對於二維平面上給定的 N 個點，我們說這些點的凸包為一個最小面積的凸多邊形，使得這個凸多邊形可以圍住所有的點。



計算二維平面上給定 N 個點的面積同樣也是一個經典的問題，身為喜歡解決演算法問題的你，肯定不能滿足於這樣簡單的問題。現在我們隨機的從這 N 個點的 2^N 種子集合中均勻隨機的選一個出來，請問這個子集合的凸包面積期望值是多少呢？（空集合的凸包面積視為 0。）

輸入格式

第一行包含一個整數 N ，表示點的數量。

接下來有 N 行，其中第 i 行包含兩個整數 x_i 、 y_i ，表示第 i 個點座落於座標 (x_i, y_i) 。

輸出格式

輸出一個實數，代表凸包的面積期望值。你輸出的答案只要與正確解答的相對或絕對誤差不超過 10^{-6} 皆會視為正確。

資料範圍

- $1 \leq N \leq 300$ 。
- $|x_i|, |y_i| \leq 10^6$ ($1 \leq i \leq N$) 。

- 所有給定的點兩兩相異。

測試範例

輸入範例 1

```
3  
0 0  
2 0  
0 1
```

輸出範例 1

```
0.1250000000000000
```

輸入範例 2

```
4  
0 0  
1 0  
2 0  
1 1
```

輸出範例 2

```
0.1875000000000000
```

輸入範例 3

```
10  
44 8  
35 12  
-46 -27  
16 -17  
-13 0  
-36 -10  
-37 -13  
-36 10  
35 18  
-23 -39
```

輸出範例 3

```
1485.1406250000000000
```

範例說明

輸入範例 1 總共有 $2^3 = 8$ 種子集合，其中子集合 $\{(0, 0), (2, 0), (0, 1)\}$ 可以形成面積為 1 的凸包。由於一種子集合被選中的機率為 $\frac{1}{2^3} = \frac{1}{8}$ ，凸包的面積期望值即為 $1 \times \frac{1}{8} = \frac{1}{8}$ 。

輸入範例 2 總共有 $2^4 = 16$ 種子集合，其中子集合 $\{(0, 0), (1, 0), (1, 1)\}$ 、 $\{(0, 0), (2, 0), (1, 1)\}$ 、 $\{(1, 0), (2, 0), (1, 1)\}$ 、 $\{(0, 0), (1, 0), (2, 0), (1, 1)\}$ 分別可以形成面積為 $\frac{1}{2}, 1, \frac{1}{2}, 1$ 的凸包。由於一種子集合被選中的機率為 $\frac{1}{2^4} = \frac{1}{16}$ ，凸包的面積期望值即為 $(\frac{1}{2} + 1 + \frac{1}{2} + 1) \times \frac{1}{16} = \frac{3}{16}$ 。

輸入範例 3 僅用作一個輸入大小稍大的測試資料供選手自我測試用。

15_典型的 IOI 題目(A_Typical_IOI_Problem)

(10分 / 10分)

時間限制: 2 seconds

記憶體限制: 1024 MB

題目敘述

今年當上阿根廷 IOI 國手的 LDC，最近有一個煩惱：IOI 的特色是有不少困難的數學題，生成函數及多項式操作只是基本，IOI 金牌選手幾乎都精通分析及代數！專精資料結構、網路流及實作的她難以適應 IOI 題型，於是她向她的老師 Frank 要了一題 IOI 練習題。題目敘述如下：

Frank 有 N 位學生，分別編號 $1, 2, \dots, N$ 。 N 位學生的人際關係可以由 $1, 2, \dots, N$ 的一個排列 p_1, p_2, \dots, p_N 和一個長度為 N 且每個字元都是 `0` 或 `1` 的字串 t 表示，其中定義一個長度為 n 的陣列 a_1, a_2, \dots, a_n 為 $1, 2, \dots, n$ 的排列如果 $1, 2, \dots, n$ 都出現在陣列 a 裡恰好一次。舉例來說， $N = 3, p = [3, 1, 2], t = \text{010}$ 為一組合法的人際關係。

他們準備要考德文期末考，於是 Frank 準備了 N 張考卷給助教 LDC 要她發給學生，每位學生都應該要被發到一張考卷。由於學生太多，她用以下的步驟發考卷：

1. 如果她手上沒有考卷了（也就是每一位學生都已經拿到考卷），結束這個流程。
2. 把手上的所有考卷交給編號最小的沒有考卷的人。
3. 喝一杯牛奶。
4. 等待直到有學生舉手。當有學生舉手的時候，把那位學生還給她的考卷拿走，然後回到步驟 1。

當學生 i 拿到 x 張考卷的時候（無論是 LDC 發的或是其他學生傳的），學生 i 會依序進行以下的操作：

1. 把 1 張考卷留給自己
2. 如果以下三個條件至少有一個成立，學生 i 會舉手，然後把 $x - 1$ 張考卷還給 LDC（LDC 在拿到考卷之後學生 i 就會把手放下），否則把 $x - 1$ 張考卷傳給學生 p_i
 - 學生 p_i 目前已經拿到考卷
 - $x = 1$
 - $t_i = 1$

不過 LDC 對學生們的人際關係只有有限的認識。她有的資訊可以用一個長度為 N 的字串 s 表示，其中每個字元都是 `0`、`1` 或 `?`。對於所有整數 i 滿足 $1 \leq i \leq N$ ，如果 $s_i \neq ?$ ，那麼 LDC 知道 $t_i = s_i$ ，否則她不知道 t_i 是什麼。而對於 p_1, p_2, \dots, p_N ，LDC 完全沒有任何資訊。

對此，假設 p_1, p_2, \dots, p_N 為均勻隨機的從所有 $1, 2, \dots, N$ 的排列選一個，而且對於所有整數 i 滿足 $1 \leq i \leq N$ 且 $s_i = ?$ ，有 $\frac{1}{2}$ 的機率 $t_i = 1$ ， $\frac{1}{2}$ 的機率 $t_i = 0$ 。假設所有的選擇都是獨立事件，問 LDC 發完考卷之後喝牛奶的杯數的期望值。因為 LDC 不喜歡小數，請輸出答案模 998244353 的值。

數學不好的 LDC 當然不會做這題，請你幫幫她，寫出能做出這題的程式。

若一個有理數 $\frac{Q}{P}$ 滿足 P, Q 都是整數、 $\gcd(P, Q) = 1$ 且 P 不被 998244353 整除，定義 $\frac{Q}{P} \pmod{998244353}$ 為唯一的整數 R 滿足 $0 \leq R < 998244353$ 且 $RP \equiv Q \pmod{998244353}$ 。

能被證明所求的期望值是一個有理數，且寫成最簡分數 $\frac{Q}{P}$ 後分母 P 不被 998244353 整除。

輸入格式

輸入第一行為一個整數 T ，代表有 T 筆測試資料。

每一筆測試資料的格式如下：

測試資料第一行為一個整數 N 。

測試資料第二行為一個長度為 N 的字串 s ，其中每一個字元都是 `0`、`1` 或 `?`。

輸出格式

對於每一筆測試資料輸出一行。那一行只有一個數字，為 LDC 喝牛奶的杯數的期望值。

資料範圍

- $1 \leq T \leq 5000$
- $1 \leq N \leq 5000$
- $|s| = N$
- 在一筆輸入檔案中，所有測試資料的 N 的總和 ≤ 5000

子任務

- 子任務 1 滿足在一筆輸入檔案中，所有測試資料的 N 的總和 ≤ 400 (10分)
- 子任務 2 沒有其他限制 (10分)

測試範例

輸入範例 1

```
2
2
?1
8
100?0?1?
```

輸出範例 1

```
249561090
522444260
```

範例說明

對於第一筆範例， t 有兩種可能：`01` 和 `11`，而 p 有兩種可能： $\{1, 2\}$ 和 $\{2, 1\}$ 。

如果 $t = \text{01}$ 且 $p = \{1, 2\}$ ，發考卷的流程如下：

1. LDC 發 2 張考卷給學生 1。

2. LDC 喝了一杯牛奶。
3. 學生 1 把 1 張考卷留給自己。
4. 由於學生 $p_1 = 1$ 已經拿過考卷，學生 1 舉手，並把 1 張考卷還給 LDC。
5. LDC 發 1 張考卷給學生 2。
6. LDC 喝了一杯牛奶。
7. 學生 2 把 1 張考卷留給自己。
8. 由於學生 $p_2 = 2$ 已經拿過考卷，學生 2 舉手，並把 0 張考卷還給 LDC。
9. 由於 LDC 手上沒有考卷了，流程結束。

在這個情況下，LDC 喝了 2 杯牛奶。

如果 $t = 01$ 且 $p = \{2, 1\}$ ，LDC 會喝 1 杯牛奶。

如果 $t = 11$ 且 $p = \{1, 2\}$ ，LDC 會喝 2 杯牛奶。

如果 $t = 11$ 且 $p = \{2, 1\}$ ，LDC 會喝 2 杯牛奶。

每一種情況發生的機率都是 $\frac{1}{4}$ ，所以 LDC 期望喝了 $\frac{2+1+2+2}{4} = \frac{7}{4}$ 杯牛奶。因為 $249561090 \times 4 \equiv 7 \pmod{998244353}$ ，所以輸出 249561090。

16_蛋餅愛涉水 (Omelet_Loves_Wading)

(3 分/10 分/12 分)

時間限制: 6 seconds

記憶體限制: 512 MB

題目敘述

「怎麼又忽然下大雨，爛台北。」

台北盆地經常下大雨，這也導致喜歡四處吃拉麵的蛋餅必須涉水而行。然而熱愛拉麵的蛋餅一點也不覺得必須在雨中前往遙遠的拉麵店是一件麻煩的事，甚至覺得這是一項有趣的挑戰，因而樂在其中，也因此練就了就算在颱風天，走路時完全不會淋到雨、也不會把鞋子弄濕的能力。擁有這項能力的蛋餅，自然不能錯過每年的淋著雨大賽。注意到這件事情的主辦單位，很擔心蛋餅太強了，會讓其他參賽者完全沒有表現的機會，因此希望降低比賽的難度。

比賽場地由 N 個廣場以及 $N - 1$ 條雙向道路組成，每條道路連接兩個廣場，並且保證從任意一個廣場，都能經過這些道路抵達其他任何一個廣場。主辦單位要在場地中選擇 K 個廣場，並在這些廣場中設置「避難點」。在這場比賽中，每個參賽者會有一個隨機的初始位置，目標是在大雨中盡快抵達任意一個避難點。因此，一個廣場的難易度被定義為從這個廣場走到設有避難點的廣場，最少需要經過幾條道路。而整場比賽的難易度，就是所有廣場的難易度的最大值。

主辦單位想要知道對於 $K = 1, 2, \dots, N$ ，比賽的最低難易度分別是多少。

輸入格式

第一行包含一個整數 N ，表示廣場的數量。

接下來有 $N - 1$ 行，其中第 i 行包含兩個整數 u_i 、 v_i ，表示第 i 條道路連接第 u_i 跟 v_i 個廣場。

輸出格式

輸出 N 行，其中第 i 個包含一個整數，表示當 $K = i$ 時比賽的最低難易度。

資料範圍

- $1 \leq N \leq 3 \times 10^5$ 。
- $1 \leq u_i, v_i \leq N$ ($1 \leq i \leq N - 1$) 。
- 保證任兩個廣場都被若干條道路連接。

子任務

- 子任務 1 滿足 $N \leq 3000$ 。
- 子任務 2 滿足 $N \leq 50000$ 。
- 子任務 3 無額外限制。

測試範例

輸入範例 1

```
9  
1 2  
1 3  
3 4  
3 5  
4 6  
4 7  
7 8  
7 9
```

輸出範例 1

```
3  
2  
2  
1  
1  
1  
1  
1  
0
```

輸入範例 2

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

輸出範例 2

```
2  
1  
1  
1  
0
```

輸入範例 3

```
16  
5 3  
16 11  
10 11  
14 16
```

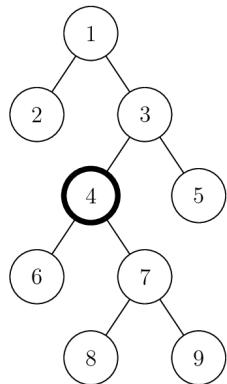
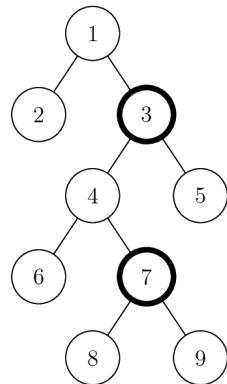
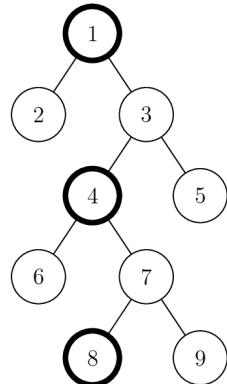
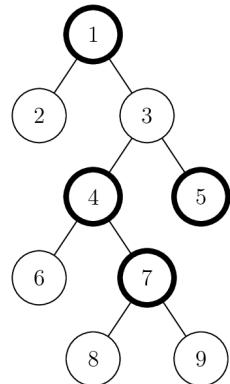
```
13 15  
9 13  
9 12  
6 7  
4 6  
1 8  
8 10  
15 2  
4 3  
1 2  
5 1
```

輸出範例 3

```
5  
5  
3  
2  
2  
1  
1  
1  
1  
1  
1  
1  
1  
1  
1  
1  
1  
0
```

範例說明

下圖為範例 1 中， K 為 1 到 4 時，其中一種使比賽難度最小的設置避難點的方法。粗框代表該廣場要設置避難點。

$K = 1, ans = 3$  $K = 2, ans = 2$  $K = 3, ans = 2$  $K = 4, ans = 1$ 

17_優奈與祕銀(Yuna_and_Mithril)

(8分 / 17分)

時間限制: 4 seconds

記憶體限制: 512 MB

題目敘述

在一個異世界裡，祕銀是最貴重的素材，用它打造的工具都會鋒利無比。要讓許多玩家都有祕銀的工具可以使用，就需要礦工與鐵匠鋪的共同努力。這世界上一共有 n 家鐵匠鋪，編號從 0 到 $n - 1$ ，然而因為加工祕銀所需要的技術很高，只有其中的 k 間能夠加工祕銀。而每天早上礦工會從礦洞挖數塊祕銀原礦，並在中午運輸給鐵匠鋪，讓鐵匠鋪能夠加工並在下午進行販售。由於祕銀是相當昂貴的素材，為了避免店家為了祕銀原礦而吵架，對於每一塊祕銀原礦，他們會等機率從 k 間可加工祕銀的鐵匠鋪選擇一間並將原礦分給該店。又由於祕銀是相當熱門的素材，每天早上挖到的原礦都一定會在當天售完。

優奈是生活在這異世界裡的冒險者，她想要要幫她的搭檔菲娜製作一把秘銀小刀，讓她更順利的完成肢解的工作。然而優奈不知道哪些鐵匠鋪是可以加工祕銀的，所以她首先會先等機率選擇一家鐵匠鋪作為起點，若她目前在第 i 間鐵匠鋪，她會進行以下流程：

- 若該鐵匠鋪沒有祕銀，則她會到移動到第 $(i + 1) \bmod n$ 間鐵匠鋪並繼續進行流程。
- 否則會在該鐵匠鋪製作祕銀小刀，並開心的回家。

現在你已經知道了哪 k 間店是可以加工祕銀的，但是礦工每天早上會挖到的祕銀原礦數量是未知數，只知道一天最多挖 m 塊原礦。所以你想要對於所有 i 從 1 到 m ，輸出若該天早上礦工挖到了 i 塊祕銀原礦，優奈期望需要拜訪幾家店才能獲得祕銀小刀。由於這個數可能是分數，若該數為 $\frac{P}{Q}$ ，請輸出 $PQ^{-1} \pmod{998244353}$ ，其中 Q^{-1} 滿足 $QQ^{-1} \equiv 1 \pmod{998244353}$ 。

輸入格式

輸入第一行會有三個正整數 n, m, k ，分別代表鐵匠鋪有幾間、每天挖的原礦數量上限與有幾間鐵匠鋪是可以加工祕銀的。

輸入第二行會有 k 個嚴格遞增的整數 a_1, a_2, \dots, a_k ，代表可加工祕銀的鐵匠鋪編號。

輸出格式

請輸出一行，該行有 m 個數字，其中第 i 個代表若礦工挖到 i 塊祕銀原礦，優奈期望需要拜訪幾家店才能成功獲得祕銀小刀。

資料範圍

- $1 \leq n < 998244353$
- $1 \leq m, k \leq 10^5$
- $k \leq n$
- $0 \leq a_1 < a_2 < \dots < a_k < n$

子任務

- 子任務 1 滿足 $m, k \leq 2000$ (8 分)
- 子任務 2 沒有其他限制 (17 分)

測試範例

輸入範例 1

```
6 6 2
1 5
```

輸出範例 1

```
499122180 831870297 499122179 332748120 748683267 457528664
```

輸入範例 2

```
864197532 1 1
565656562
```

輸出範例 2

```
931220943
```

輸入範例 3

```
20 10 5
0 3 7 14 19
```

輸出範例 3

```
499122187 411276681 367353928 350583422 846511216 333512644 47264079 475033985 232520342
449759810
```

範例說明

在範例測資 1 中：

- 若 $i = 1$ ，則有 $\frac{1}{2}$ 的機率在第 1 間鐵匠鋪會有祕銀，有 $\frac{1}{2}$ 的機率在第 5 間鐵匠鋪會有祕銀，因此答案為 $\frac{1}{2} \times \frac{2+1+6+5+4+3}{6} + \frac{1}{2} \times \frac{5+4+3+2+1+6}{6} = \frac{7}{2}$ 。因為 $2^{-1} \equiv 499122177 \pmod{998244353}$ ，所以需要輸出 $7 \times 499122177 \equiv 499122180 \pmod{998244353}$ 。
- 若 $i = 2$ ，則有 $\frac{1}{4}$ 的機率只有第 1 間鐵匠鋪有祕銀，有 $\frac{1}{4}$ 的機率只有第 5 間鐵匠鋪會有祕銀，而有 $\frac{1}{2}$ 的機率第 1 與第 5 間鐵匠鋪都有祕銀，因此答案為 $\frac{1}{4} \times \frac{2+1+6+5+4+3}{6} + \frac{1}{4} \times \frac{5+4+3+2+1+6}{6} + \frac{1}{2} \times \frac{2+1+4+3+2+1}{6} = \frac{17}{6}$ 。因為 $6^{-1} \equiv 166374059 \pmod{998244353}$ ，所以需要輸出

$$17 \times 166374059 \equiv 831870297 \pmod{998244353}.$$

- 若 $i = 3$ ，答案為 $\frac{5}{2}$ 。
- 若 $i = 4$ ，答案為 $\frac{7}{3}$ 。
- 若 $i = 5$ ，答案為 $\frac{9}{4}$ 。
- 若 $i = 6$ ，答案為 $\frac{53}{24}$ 。

在範例測資 2 中：

- 若 $i = 1$ ，答案為 $\frac{864197533}{2}$ 。

18_Nachoneko 做衣服 (Nachoneko_and_Beautiful_Clothes)

(1 分/1 分/8 分/15 分)

時間限制: 5 seconds

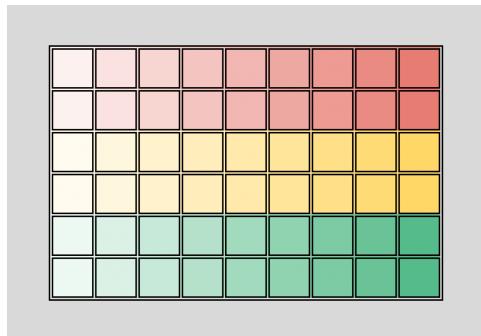
記憶體限制: 1024 MB

題目敘述

Nachoneko 媽媽正在為可愛的孩子們做女裝，而第一步就是要先找出適合的顏色們。

她拿出了一張布料色卡，打算從裡面剪下一些布料來做實驗。

這張布料色卡可以被想像成一個高 h 公分寬 w 公分的矩形，為了方便溝通，我們將其分為 h 個 row 跟 w 個 column，再給由上往下數的第 r 個 row、由左往右數的第 c 個 column 的 1×1 矩形編號為 (r, c) 。



一張 $(h, w) = (6, 9)$ 的布料色卡。

其中最右上角的紅色矩形編號為 $(1, 9)$ 、最左下角的白綠色矩形編號為 $(6, 1)$ 。

現在，Nachoneko 想要從裡面剪下 n 個高度為 1 的矩形，每個矩形可以以三個參數 $(r[i], c[i][0], c[i][1])$ 來表示，代表這個矩形的兩個端點座標是 $(r[i], c[i][0])$ 跟 $(r[i], c[i][1])$ 。愛惜布料的她不希望這 n 個區間以外的布料被剪下來（為此，她只會需要 r 是奇數的那些區間），不過這些區間內的布料可以被剪成任意份。

Nachoneko 紿了你三種可以做的操作，你能夠告訴她應該要怎樣才能把布料都剪下來嗎？

1. **combine** $c[0] \ c[1]$ ：把整張布料的第 $c[0]$ 個 column 到第 $c[1]$ 個 column 壓縮到一起，請注意布料的編號並不會因此而改變。
2. **cut** $r \ c$ ：把跟編號 (r, c) 的布料重疊的所有布料都剪下來。
3. **spread**：把整張布攤開，你必須且只能執行該操作 1 次，且這必須是你的最後一個操作。

因為 Nachoneko 的體力不好，所以如果你讓她做超過 $q_{\max} = 300\,000$ 次操作她就會躺在地上跟地板融為一體，沒辦法再給你 AC。

輸入格式

- line 1: **subtask**: 代表測資所屬的子任務編號。
- line 2: $h \ w \ n$

- line $3 + i$ ($0 \leq i \leq n - 1$) : $r[i] \ c[i][0] \ c[i][1]$

輸出格式

- line 1: q
- line $2 + i$ ($0 \leq i \leq q - 1$) : $\text{Operation}[i]$
- 你輸出的 q 必須滿足 $1 \leq q \leq q_{\max}$ 。
- **Operation** 的格式以及限制如下：
 1. **combine** $c[0] c[1]$
 - 你的輸出必須滿足 $1 \leq c[0] \leq c[1] \leq w$ 。
 2. **cut** $r c$
 - 你的輸出必須滿足 $1 \leq r \leq h$ 及 $1 \leq c \leq w$ 。
 3. **spread**
 - 你必須且只能執行該操作 1 次（即便你沒有進行任何 **combine** 操作），且這必須是你的最後一個操作。
- 請在每一個操作之後換行，且**不要在行尾輸出多餘的空格**，否則你可能會得到 Wrong Answer。

資料範圍

- $1 \leq \text{subtask} \leq 3$ 。
- $q_{\max} = 300\,000$ 。
- $1 \leq h, w \leq 120\,000$ 。
- $1 \leq n \leq 500$ 。
- $1 \leq r[i] \leq h$ ($0 \leq i \leq n - 1$)。
- $r[i]$ 是奇數 ($0 \leq i \leq n - 1$)。
- $1 \leq c[i][0] \leq c[i][1] \leq w$ ($0 \leq i \leq n - 1$)。
- 輸入的數字都是整數。

子任務

- 子任務 1 滿足 $h \leq 100$ 、 $w \leq 100$ 。
- 子任務 2 滿足 $c[i - 1][1] < c[i][0]$ ($1 \leq i \leq n - 1$)。
- 子任務 3 滿足 $n \leq 100$ 。
- 子任務 4 無額外限制。

在子任務中你可獲得部分分數。令 q_1, q_2, \dots, q_k 是你在該子任務中的每一筆測資所輸出的 q 值； q'_1, q'_2, \dots, q'_k 是該子任務中每一筆測資的理論最小操作次數；且 $Q = \max\{q_1, q_2, \dots, q_k\}$ 。你在四個子任務的得分將依下表計算：

操作次數	子任務 1、2 得分
$6001 \leq Q \leq 300\,000$	0.5
$Q \leq 6000$	1

操作次數	子任務 3 得分
$2001 \leq Q \leq 300\,000$	1
$Q \leq 2000$ 且存在測資使 $q_i > q'_i$	3
$Q \leq 2000$ 且所有測資皆 $q_i \leq q'_i$	8

操作次數	子任務 4 得分
$180\,001 \leq Q \leq 300\,000$	2
$30\,001 \leq Q \leq 180\,000$	4
$10\,001 \leq Q \leq 30\,000$	6
$Q \leq 10\,000$ 且存在測資使 $q_i > q'_i$	10
$Q \leq 10\,000$ 且所有測資皆 $q_i \leq q'_i$	15

測試範例

輸入範例 1

```
1
10 10 5
1 3 7
3 4 5
5 10 10
7 1 10
9 3 9
```

不考慮 subtask 的話，該範例輸入符合子任務 1, 3, 4 的限制。

輸出範例 1

```

10
combine 4 5
cut 3 5
combine 3 7
cut 1 5
cut 5 10
combine 6 9
cut 9 8
combine 1 10
cut 7 1
spread

```

可以證明不存在 $q \leq 9$ 的解。

輸入範例 2

```

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

```

不考慮 subtask 的話，該範例輸入符合子任務 1, 3, 4 的限制。

輸出範例 2

```

3
combine 1 5
cut 1 3
spread

```

可以證明不存在 $q \leq 2$ 的解。

輸入範例 3

```

2
1 100 3
1 1 2
1 3 98
1 99 100

```

輸出範例 3

```

3
combine 1 100
cut 1 1
spread

```

部份分範例

輸入

```
2
1 100 3
1 1 2
1 3 98
1 99 100
```

不考慮 subtask 的話，該範例輸入符合子任務 2, 3, 4 的限制。

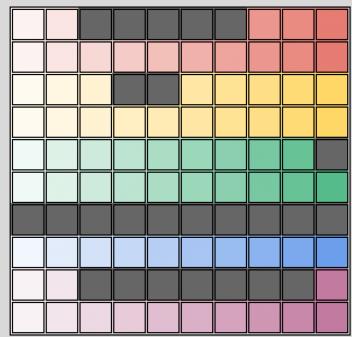
輸出

```
10
combine 1 2
cut 1 1
cut 1 2
combine 3 98
combine 31 41
cut 1 3
combine 99 100
cut 1 99
cut 1 1
spread
```

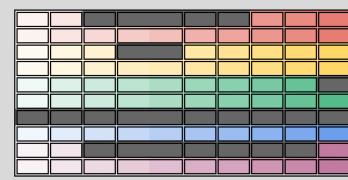
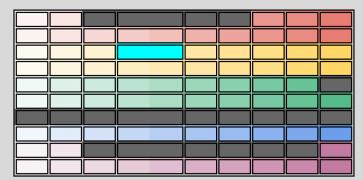
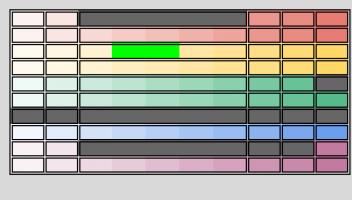
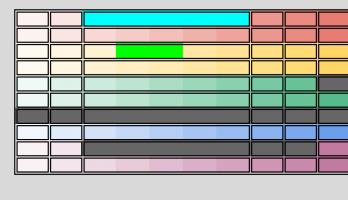
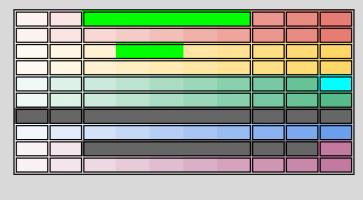
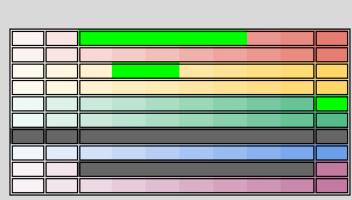
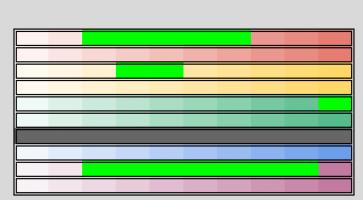
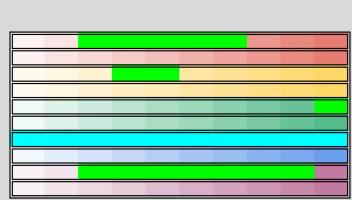
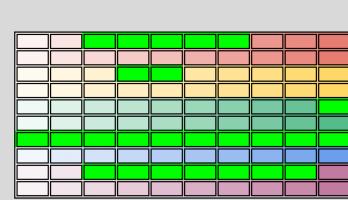
最佳解的 $q = 3$ 。如果你的輸出長這樣，將只能在子任務 3, 4 拿到部份分。

範例說明

範例 1 圖示如下，其中灰色區塊表示所有要剪下來的布料、天藍色區塊為當前 `cut` 操作剪下的布料、亮綠色區塊為已經被剪下來的布料們、而分隔線消失代表著對應的 column 經由 `combine` 操作被壓縮在一起。



範例 1 的初始狀態

`combine 4 5``cut 3 5``combine 3 7``cut 1 5``cut 5 10``combine 6 9``cut 9 8``combine 1 10``cut 7 1``spread`

對於第六次的 `combine 6 9` 操作，由於該操作之前 column 3 到 7 已經被壓縮到一起了，因此該操作相當於 `combine 3 9`。

請記得你的最後一個操作必須是 `spread`。

範例 2 圖示如下，其中灰色區塊表示所有要剪下來的布料：



即便是被指定了多次的位置（如 (1, 3)） ，你也只需要對這個位置執行一次 `cut` 操作就好。

範例 3 中，即便沒有必要，你還是可以重複剪下同一個位置，也可以重複壓縮已經被壓縮過的區間。

提示：子任務 4 的部份分數比子任務 3 更好拿。

後記

多虧了你的幫忙，Nachoneko 媽媽成功的為孩子們做出了最適合的衣服，P 喵也如願以償的穿上了女裝。真是可喜可賀！

19_有趣的樹上結構 (Interesting_Tree_Structure)

(2 分/10 分/3 分/4 分/6 分)

時間限制: 5 seconds

記憶體限制: 1024 MB

題目敘述

一天放學後，Rikki 在學校中庭的樹上找到了翹了練習正在睡覺的貓貓樂奈。為了掌握樂奈神出鬼沒的行蹤，Rikki 用了抹茶巴菲戰術來引誘她。樂奈告訴 Rikki 她只會在 **有趣的樹** 上睡覺，且只會在 **快要壞掉的樹** 旁跟小貓玩，而分辨這些樹的規則如下：

1. 這棵樹上有 $2k$ 個樹洞，編號為 $1, 2, \dots, 2k$ 。
2. 任兩個不同的樹洞 u 、 v 之間可能沒有邊，也可能有一條單向邊，但不會有重邊，也不會兩個方向都有單向邊。
3. 樂奈根據樹的形狀，會給出 k 個「希望從編號 a_i 的樹洞能走到編號 b_i 的樹洞」這樣的需求。
4. 在所有滿足前三個條件的樹洞關係圖中，所有使用到的邊的數量最少（令這個數量為 r ）的樹，就是 **有趣的樹**。
5. 所有使用到的邊數量為 r 的樹中，只要不同時滿足前三個條件，都是 **快要壞掉的樹**。
6. 所有使用到的邊數量不為 r 的樹，都是 **無趣的樹**。

燈幫助 Rikki 整理出了校園中樹上樹洞的連通情況 $E = \{(u_j, v_j)\}_{j=1}^m$ ，但她們都解決不了這個問題，於是燈跟 Rikki 找上了會寫程式的喵喵來幫忙。

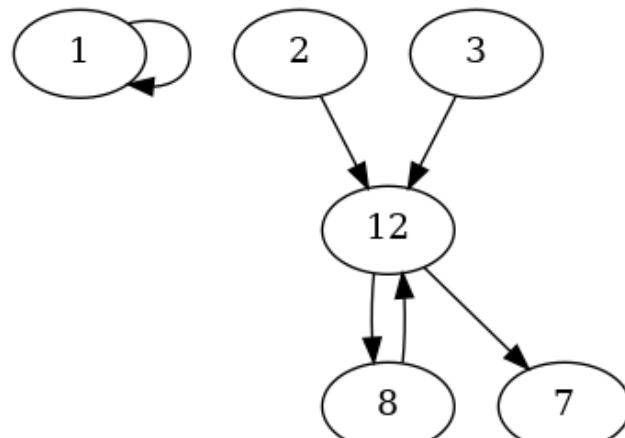
~~不出意外的，~~ 嘴巴又把題目丟給了你。現在給你樂奈的需求以及樹洞的連通情況，請你幫 Rikki 分類這些樹。

有 t 組測資。

P.S. 樹洞之間會形成一張有向圖。雖然樹洞長在樹上，但這張圖不一定是樹。

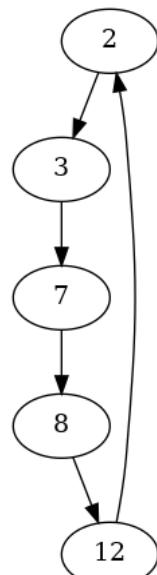
例子請參考下一頁。

以 $k = 6$ 且 $\{(a_i, b_i)\}_{i=1}^k = \{(1, 1), (2, 12), (12, 7), (3, 12), (12, 8), (8, 12)\}$ 為例，需求關係圖如下圖所示：

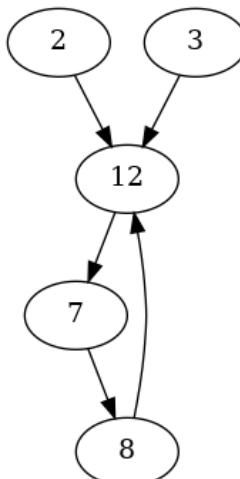


需求關係圖

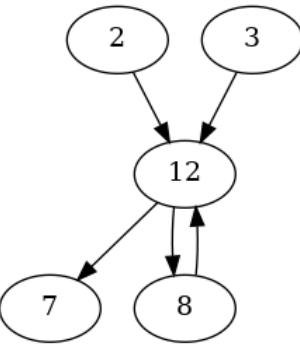
考慮以下幾種不同的樹洞結構：



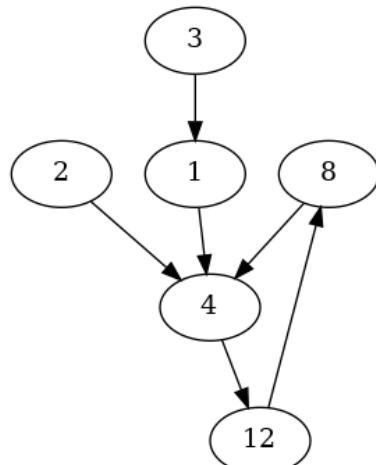
第一棵樹



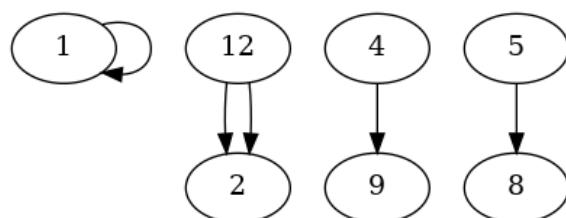
第二棵樹



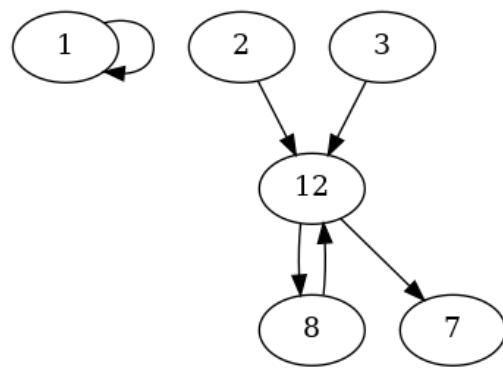
第三棵樹



第四棵樹



第五棵樹



第六棵樹

	第一棵 樹	第二棵 樹	第三棵樹	第四棵 樹	第五棵樹	第六棵 樹
滿足第二個條件	True	True	False	True	False	False
滿足所有樂奈的需 求	True	True	True	False	False	True
使用到的邊數量	5	5	5	6	5	6
樹的分類	有趣的 樹	有趣的 樹	快要壞掉的 樹	無趣的 樹	快要壞掉的 樹	無趣的 樹

輸入格式

- line 1: t
- block c ($1 \leq c \leq t$): TestCase_c

其中， TestCase_c 的格式為：

- line 1: k
- line $1 + i$ ($1 \leq i \leq k$): $a_i \ b_i$
- line $k + 2$: m
- line $k + 2 + j$ ($1 \leq j \leq m$): $u_j \ v_j$

輸出格式

- line c ($1 \leq c \leq t$): Answer_c

其中， Answer_c 的格式為下列三者之一：

- 有趣的樹：**Omoshiree**
- 快要壞掉的樹：**Kowaresou**
- 無趣的樹：**Tsumannee**

資料範圍

- $1 \leq t \leq 10$ 。
- $1 \leq k \leq 100\,000$ 。
- $1 \leq a_i, b_i \leq 2k$ ($1 \leq i \leq m$)。
- $1 \leq m \leq 100\,000$ 。
- $1 \leq u_j, v_j \leq 2k$ ($1 \leq j \leq m$)。
- 輸入的數字都是整數。

子任務

- 子任務 1 滿足 $a_i < b_i$ ($1 \leq i \leq k$) 、 $u_j = v_j = 1$ ($1 \leq j \leq m$)。
- 子任務 2 滿足 $u_j = v_j = 1$ ($1 \leq j \leq m$)。
- 子任務 3 滿足 $k \leq 800$ 、 $m \leq 800$ 。
- 子任務 4 滿足 $k \leq 20\,000$ 、 $m \leq 20\,000$ 。
- 子任務 5 無額外限制。

測試範例

輸入範例 1

```
2
4
1 2
2 1
3 1
1 3
3
1 2
2 3
3 1
8
1 3
1 7
2 7
4 4
3 7
1 6
1 7
1 1
4
1 3
3 6
6 2
2 7
```

該範例輸入符合子任務 3, 4, 5 的限制。

輸出範例 1

```
Omoshiree
Omoshiree
```

輸入範例 2

```
3
5
1 2
2 4
3 6
4 8
5 10
9
1 2
2 3
3 4
4 5
5 6
6 7
7 8
8 9
9 10
4
1 2
2 7
8 7
7 2
4
1 2
2 7
8 7
7 2
3
1 1
1 2
6 6
1
2 1
```

該範例輸入符合子任務 3, 4, 5 的限制。

輸出範例 2

```
Tsumannee
Kowaresou
Kowaresou
```