



0_送分題 – Hello World

(30分)

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!
```

1_路燈 (Streetlights)

(2分 / 8分)

時間限制: 1 second

記憶體限制: 256 MB

問題敘述

路燈照明系統，是為了確保在天黑的時候，也可以有著足夠的燈光，至少在車輛行駛等狀況下，不至於發生危險。

由於前陣子在一個小村莊發生了過多的車禍事故，根據調查結果發現，是因為路燈設置的明顯不足，因此村長決定好好處理這件事情，以新增更多路燈來確保安全。但由於經費有限，因此如何妥善的設置，將成為村長要擔心的問題。

根據一些光學原理等物理知識和一些調查，村長發現每 15 公尺內一定要至少有一個路燈才會安全，意即對於路燈設置在 x 座標上， $(x - 15, x + 15)$ 是照明範圍，且必須要在 $x - 15$ 和 $x + 15$ 設置路燈才會安全。

也就是說如果一條路長為 100 單位（起點為 1、終點為 100，即 $[1, 100]$ 的閉區間），在沒有設立任何路燈的狀況下，將路燈設置在 15, 30, 45, 60, 75, 90 是最小達到安全的標準。

現在這一條路的長度 L ，且已經設立了 N 個路燈，試問最少還要新增多少路燈，使得 $[1, L]$ 這個區間達到安全門檻？

輸入格式

輸入的第一行會包含兩個正整數 L 、 N ，代表路道長度和路燈數量。

第二行有 N 個正整數 a_1, a_2, \dots, a_N 表示路燈位置。

輸出格式

輸出一個整數，代表最少需要新增的路燈數量。

資料範圍

- $1 \leq L \leq 10^{18}$ 。
- $1 \leq N \leq 2 \times 10^5$ 。
- $1 \leq a_i \leq L$ ($1 \leq i \leq N$) 。
- 保證沒有兩個以上的路燈在同一個位置上，也就是對所有 $1 \leq i < j \leq N$ 都有 $a_i \neq a_j$ 。

子任務

- 子任務 1 (2 points) $L \leq 10^5$ 、 $N \leq 1000$ 。
- 子任務 2 (8 points) 無額外限制。

範例

輸入範例 1

```
100 4  
60 15 90 30
```

輸出範例 1

```
2
```

輸入範例 2

```
50 1  
16
```

輸出範例 2

```
3
```

輸入範例 3

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

輸出範例 3

```
0
```

範例說明

範例 1 中，需新增的路燈位置為 45, 75 即可達到安全門檻。

範例 2 中，需新增的路燈位置可以是 1, 31, 46 以達到安全門檻。

範例 3 中，已經達到安全門檻，並不需要新增路燈。# Streetlights

2_教授的教授 (Professor_of_Professors)

(10分)

時間限制: 1 second

記憶體限制: 256 MB

題目敘述

國立長頸鹿大學有 N 位教授，其中每一位教授有最多一位指導教授，而且教授 i 說自己是 a_i 位教授的指導教授。注意到自己不能是自己的指導教授。

來自國立短頸鹿大學的鄭教授最近在研究長頸鹿大學的教授生態。他定義教授 x 是教授 y 的上司若且唯若存在若干位教授 $x = v_1, v_2, \dots, v_k = y$ 使得對於所有 $1 \leq i < k$ ，教授 v_i 是教授 v_{i+1} 的指導教授。注意到自己不算自己的上司。

在研究中，鄭教授發現在這 N 位教授中，恰好只有一位教授滿足其他教授都不是他的指導教授，而且這位教授是其他所有教授的上司。他稱呼這位特別的教授為雞教授。

令 c_i 為教授 i 的上司個數。鄭教授想知道 $\max_{1 \leq i \leq N} \{c_i\}$ 最小可以是多少？同時，他也想知道一個達到此最小值的指導教授分配方式。

身為鄭教授實驗室研究生的你，請幫鄭教授算出答案。

輸入格式

第一行輸入一個正整數 T ，代表接下來有 T 筆子測試資料。

對於每一筆測試資料：

第一行輸入一個正整數 N 。

第二行輸入 N 個非負整數 a_1, a_2, \dots, a_N 。

輸出格式

每一筆子測試資料輸出兩行。

第一行輸出一個整數，代表 $\max_{1 \leq i \leq N} \{c_i\}$ 的最小值。

第二行輸出 N 個整數 p_1, p_2, \dots, p_N ，代表達到此最小值的指導教授分配方式，其中如果教授 i 是雞教授，則 $p_i = 0$ ，否則教授 p_i 為教授 i 的指導教授。如果有多種達到最小值的分配方式，輸出任意一種即可。

資料範圍

- $1 \leq T \leq 1000$
- $2 \leq N \leq 3 \cdot 10^5$
- $0 \leq a_i \leq N - 1$
- $\sum_{i=1}^N a_i = N - 1$

- 保證所有子測試資料的 N 總和不超過 $3 \cdot 10^5$

測試範例

輸入範例 1

```
3
5
2 1 0 1 0
10
3 0 2 0 1 1 0 0 2 0
9
0 0 0 0 0 0 0 0 8
```

輸出範例 1

```
2
0 1 2 1 4
3
0 5 1 6 9 1 3 3 1 9
1
9 9 9 9 9 9 9 9 0
```

範例說明

對於第一筆子測試資料，一種可行的方式是讓教授 1 成為雞教授且為教授 2, 4 的指導教授，而教授 2, 4 依序為教授 3, 5 的指導教授。在這樣的分配方式下，我們有 $(c_1, c_2, c_3, c_4, c_5) = (0, 1, 2, 1, 2)$ ，因此 $\max_{1 \leq i \leq n} \{c_i\} = 2$ 。

注意到答案不一定是唯一的，例如對於第一筆子測試資料：

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

也是一個可能的答案。

3_Code_Guardian

(15分)

時間限制: 1 second

記憶體限制: 256 MB

題目敘述

自從 OpenAI 推出 ChatGPT，人工智能領域在過去的半年內快速演進。許多工作已開始透過與AI協作來更高效地滿足需求。其中，與電腦的溝通成為協作的瓶頸。因此，利用特定領域語言 (DSL, domain specific language) 來加速提問 (prompt) 成為關鍵之一。例如，在 Midjourney 上使用美術專用語句來產生圖片（例如：請使用「印象派畫法」來...），或者在正在研發中的 Mojo 語言中，使用 autotune 可以根據處理器核心數自動進行執行優化。

基於這個想法，再加上「懶」是人類進步的原動力，你需要寫一個程式，利用人工智能來幫助自己進行函式註解和錯誤處理，以便未來撰寫的程式更易於閱讀且能夠良好處理錯誤。為了方便說明與處理，假設程式碼的每一層縮排都是2個空白，空行將被替換為 "emptyline"，需要錯誤處理的地方將被標記為 "safe"。而你的程式將對它們進行以下處理：

- 將 "safe" 轉換為 "!!!Analyze(<同一層未被解析的程式碼，去掉前後空白並以逗號隔開>)"。註：原本意思為 "ErrorCheck"，為了方便說明改為 "!!!"。
- 將 "emptyline" 轉換為 "//(Describe(程式碼區塊的起始行數，結束行數))"。註：原本意思為 "FunctionComment"，為了方便說明並節省打字，改為 "//"。為了簡化問題，只有前端沒有空白時才替換 emptyline，當然 function 後的 emptyline 不屬於 function。
- 若縮排不正確，則僅輸出 "ERR!"。
- 最後一行為 "end" 時輸入的程式結束

輸入的程式碼	輸出的程式碼	以實際程式取代 In# 較容易理解
<pre> emptyline 1n2 1n3 SPSP 1n4 SPSP 1n5 SPSPSPSP 1n6 SPSPSPSP 1n7 SPSPSPSP 1n8 SPSPSPSP safe SPSPSPSP 1n10 SPSPSPSP safe SPSPSPSP 1n12 SPSP 1n13 SPSP 1n14 SPSP safe 1n16 emptyline emptyline 1n19 SPSP 1n20 safe end </pre>	<pre> //Describe(2,16) 1n2 1n3 SPSP 1n4 SPSP 1n5 SPSPSPSP 1n6 SPSPSPSP 1n7 SPSPSPSP 1n8 SPSPSPSP !!!Analyze(1n6,1n7,1n8) SPSPSPSP 1n10 SPSPSPSP !!!Analyze(1n10) SPSPSPSP 1n12 SPSP 1n13 SPSP 1n14 SPSP safe 1n16 emptyline //Describe(19,21) 1n19 SPSP 1n20 !!!Analyze(1n19) </pre>	<pre> # main - create markdown output @timer def main(): result = [] for line in sys.stdin.readlines(): line = line.rstrip() line = line.replace(' ', '&#9428;') line = line+
 raise if len(line) >= MAX_LINE_SIZE result.append(line) raise if len(result) >= MAX_ARRAY_SIZE print('.',end='') fh = open('missclose.txt', 'w+') fh.write(''.join(result)) raise if filesize('missclose.txt') == 0 # end of main # Entry Point if name == 'main': main() raise if diskfull() </pre>

- 第 1 個 emptyline 轉成需要 AI describe code 2-16 行，第 2 個 emptyline 沒有轉換因為沒有 function，第 3 個則轉 19-21 行。

- 第1個 safe 轉成需要 AI analyze “ln6,ln7,ln8” 的內容，第 2 個 safe 則只需分析 ln10，第 3 個 safe 需分析同層未分析的內容 (ln4,ln5,ln13,ln14)，以及最後一行的 safe 則只需分析第二個 function ln19 的內容。

PS: 'sp' is '' (ASCII: 0x20) 字元

輸入格式

多行的文字，以完全空白行結束

輸出格式

輸出 “ERR!” 如果縮排錯誤
如無錯誤則輸出替換好的內容

資料範圍

英文字母、數字、空白、括弧及換行字元

測試範例

輸入範例 1

```
emptyline
ln2
ln3
ln4
ln5
    ln6
    ln7
    ln8
safe
ln10
safe
ln12
ln13
ln14
safe
ln16
emptyline
emptyline
ln19
    ln20
safe
end
```

輸出範例 1

```
//Describe(2,16)
ln2
ln3
ln4
ln5
ln6
ln7
ln8
!!!Analyze(ln6,ln7,ln8)
ln10
!!!Analyze(ln10)
ln12
ln13
ln14
!!!Analyze(ln4,ln5,ln13,ln14)
ln16
emptyline
//Describe(19,21)
ln19
ln20
!!!Analyze(ln19)
```

輸入範例 2

```
emptyline
emptyline
ln3
emptyline_safe_emptyline
ln5
safe
ln7
safe
ln9
ln10
ln11
safe
emptyline
ln14
end
```

輸出範例 2

```
emptyline
//Describe(3,12)
ln3
emptyline_safe_emptyline
ln5
!!!Analyze(ln5)
ln7
!!!Analyze(ln7)
ln9
ln10
ln11
!!!Analyze(ln9)
//Describe(14,14)
ln14
```

輸入範例 3

```
emptyline
emptyline
emptyline
safe
    safe
end
```

輸出範例 3

ERR!

範例說明

1-4行是原本應輸出如下，但因第5行內縮多一層所以只輸出 ERR!

```
emptyline
emptyline
//Describe(4,4)
!!!Analyze()
```

4_優奈與委託 (Yuna_and_Quests)

(10分)

時間限制: 1 second

記憶體限制: 256 MB

題目敘述

優奈是一個冒險者，職業為熊的她總是穿著熊熊服裝在冒險者公會中接受各式各樣的委託，雖然因為奇特的服裝時常引起初次見面的人的歧視，但那又是另一個故事了。

優奈有兩種服裝與兩種召喚獸作為坐騎，服裝為白熊服裝與黑熊服裝，而召喚獸為熊緩與熊急。這兩種服裝與召喚獸有著不同的能力，可以幫助她完成公會的各種委託。

今天優奈接受了 n 個委託，為了讓進行委託的效率變好，她將每個委託用兩個數值 a_i, b_i 表示，分別代表該委託所需要的服裝種類與坐騎的種類，具體來說：

- 若 $a_i = 1$ ，則代表她需要穿黑熊服裝
- 若 $a_i = 2$ ，則代表她需要穿白熊服裝
- 若 $a_i = 3$ ，則代表該委託對於服裝種類沒有要求
- 若 $b_i = 1$ ，則代表她需要騎乘熊緩
- 若 $b_i = 2$ ，則代表她需要騎乘熊急
- 若 $b_i = 3$ ，則代表該委託對於坐騎沒有要求

一開始優奈穿著黑熊服裝，並騎著熊緩，她希望能夠按照順序完成 n 個委託。但由於更換衣服或更換坐騎很麻煩，她希望讓更換的總次數越小越好，請問她最少的次數為何？

輸入格式

輸入第一行有一個整數 n ，代表委託的數量。

接下來的 n 行，第 i 行會有兩個正整數 a_i, b_i ，代表第 i 個委託的數值。

輸出格式

請輸出一行，該行有一個數字，代表優奈最少需要的更換次數。

資料範圍

- $1 \leq n \leq 1000$
- $1 \leq a_i, b_i \leq 3$

測試範例

輸入範例 1

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

輸出範例 1

```
2
```

輸入範例 2

```
8  
3 3  
3 3  
3 3  
3 3  
3 3  
3 3  
3 3  
3 3
```

輸出範例 2

```
0
```

輸入範例 3

```
8  
2 2  
1 1  
2 2  
1 1  
2 2  
1 1  
2 2  
1 1
```

輸出範例 3

16

範例說明

在範例測資 1 中，優奈的其中一個最優解為在一開始將坐騎換為熊急，並在完成第一個委託後將服裝換為白熊服裝，這樣一來總更換次數為 2，可以證明不存在更小的答案。

5_Minimum_Distance

(3分 / 3分 / 4分)

時間限制: 1 second

記憶體限制: 256 MB

問題敘述

2D 平面上有 N 個點。定義這個 2D 平面上任兩點 $(a, b) \cdot (c, d)$ 的距離為 $\max(|a - c|, |b - d|)$ ，請找出一個非負整數座標 (x, y) 使得它跟這 N 個點間的最大距離最小。

輸入格式

輸入的第一行會包含一個正整數 N ，代表 2D 平面上有 N 個點。

在接下來的 N 行中，第 i 行會包含兩個非負整數 $x_i \cdot y_i$ ，代表第 i 個點的座標。

輸出格式

第一行請輸出一個非負整數 r ，代表這個座標跟這 N 個點間的最大距離。

第二行請輸出兩個非負整數 $x \cdot y$ ，代表該座標。如果有多個座標滿足條件，請輸出字典序最小的一個。（如果有多个滿足條件的座標，取 x 座標較小的；如果仍有多個滿足條件的座標，則再取 y 座標較小的。）

資料範圍

- $1 \leq N \leq 200\,000$ 。
- $0 \leq x_i, y_i \leq 10^9$ ($1 \leq i \leq N$) 。

子任務

- 子任務 1 (3 points) $1 \leq N \leq 5000 \cdot 0 \leq x_i, y_i \leq 1000$ ($1 \leq i \leq N$) 。
- 子任務 2 (3 points) $0 \leq x_i, y_i \leq 1000$ ($1 \leq i \leq N$) 。
- 子任務 3 (4 points) 無額外限制。

範例

輸入範例 1

```
2
1 2
2 1
```

輸出範例 1

```
1  
1 1
```

輸入範例 2

```
7  
801 554  
766 608  
825 81  
850 537  
897 460  
305 595  
666 448
```

輸出範例 2

```
296  
601 312
```

範例說明

範例1中， $(1, 1), (1, 2), (2, 1), (2, 2)$ 與輸入中兩點的最遠距離皆為1，其中字典序最小的點為 $(1, 1)$ 。

6_胖胖貓的減肥計畫 (Weight_Loss_Plan)

(15分)

時間限制: 1 second

記憶體限制: 256 MB

問題敘述

在喵喵學院當中，有一個非常喜歡競程的橘貓。他平時最喜歡做的事情就是坐在電腦前面吃零食寫題目。隨著他寫出來的題目越來越多，零食消耗的速度也越來越快，胖胖貓也變得越來越肥。因此喵喵學院的大家也開始戲稱他為胖胖貓。

胖胖貓為了不讓大家繼續嘲笑他，他決定開始健身。他在 YouTube 上查詢到了許多健身影片，並整理了出 N 個減肥動作。這些減肥動作之間有 $N - 1$ 組關聯性，每組關聯性代表哪兩個減肥動作之間是互相關聯，且這些關聯性為雙向的。關聯性之間沒有傳導性，假如 A 和 B 之間有關聯性， B 和 C 之間有關聯性，並不代表 A 和 C 之間也有關聯性。因為這些減肥動作是胖胖貓精挑細選的，因此對於任意兩個減肥動作 s 和 t ，一定找得到一系列的減肥動作 e_1, e_2, \dots, e_k 使得 $e_1 = s \wedge e_k = t$ ，且對所有 $1 \leq i \leq k - 1$ ， e_i 和 e_{i+1} 都有關聯性。

一份健身組合是由任意一個動作開始，並完成至少 2 個不重複的減肥動作所組成，其中每兩個相鄰的動作之間都必須要有關聯性。每完成一對有關聯性的動作都會改變胖胖貓的開心度。

令「組合中最高的開心度」為 M 、「組合中最低的開心度」為 m 、「組合中動作的數量」為 n ，我們定義一份健身組合的困難度為 $\lfloor \frac{M}{2} \rfloor \oplus (m \times 2) \oplus n$ ，其中 \oplus 是位元 XOR。

胖胖貓為了避免減肥的過程太過困難或無聊，他決定選擇兩個不同的健身組合來健身。兩個健身組合不相同若且唯若兩者包含的減肥動作中，至少有一個減肥動作不相同。最後，胖胖貓很好奇是否存在兩個符合上述條件，且難度相同的健身組合。

輸入格式

輸入的第一行包含一個正整數 N ，代表減肥動作的數量。

接下來一共有 $N - 1$ 行，每行有三個正整數 U_i, V_i, W_i ，代表完成這組關聯性後胖胖貓的開心度會變成 W_i 。

輸出格式

若有解答，輸出四個整數 a, b, c, d ，代表 a 到 b 的健身組合與 c 到 d 的健身組合困難度相同且動作不完全相同。如果 (a, b, c, d) 是一組解，輸出 $(b, a, c, d), (a, b, d, c), (b, a, d, c)$ 也都算正確。若無解則輸出 -1 。

資料範圍

- $3 \leq N \leq 2 \times 10^5$ 。
- $1 \leq U_i, V_i \leq N$ (對所有 $1 \leq i \leq N - 1$)。
- $1 \leq W_i \leq 10^6$ (對所有 $1 \leq i \leq N - 1$)。
- 保證對於任意兩個減肥動作 s 和 t ，一定找得到一系列的減肥動作 e_1, e_2, \dots, e_k 使得 $e_1 = s \wedge e_k = t$ ，且對所有 $1 \leq i \leq k - 1$ ， e_i 和 e_{i+1} 都有關聯性。

子任務

- 子任務 1 (100 points) 無額外限制。

範例

輸入範例 1

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

輸出範例 1

```
1 2 2 3
```

輸入範例 2

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

輸出範例 2

```
-1
```

範例說明

在範例一中：

健身組合 $1 \rightarrow 2$ 的最高開心度 $M = 1$ ，最低開心度 $m = 1$ ，動作數量為 $n = 2$ ，因此困難度為 $\lfloor \frac{1}{2} \rfloor \oplus (1 \times 2) \oplus 2 = 0$ 。

健身組合 $2 \rightarrow 1 \rightarrow 3$ 的最高開心度為 $M = 2$ ，最低開心度為 $m = 1$ ，動作數量為 $n = 3$ ，因此困難度為 $\lfloor \frac{2}{2} \rfloor \oplus (1 \times 2) \oplus 3 = 0$ 。

因為兩個健身組合完成的動作不完全相同，且困難度相同，因此這是一組合法的解。

注意 $(2, 3, 3, 2)$ 不是一組合法的解，因為完成的動作 $\{2, 1, 3\}$ 和 $\{3, 1, 2\}$ 相同。

在範例二中：

$1 \rightarrow 2$ 的困難度為 $\lfloor \frac{1}{2} \rfloor \oplus (1 \times 2) \oplus 2 = 0$ 。

$1 \rightarrow 3$ 的困難度為 $\lfloor \frac{5}{2} \rfloor \oplus (5 \times 2) \oplus 2 = 10$ 。

$2 \rightarrow 3$ 的困難度為 $\lfloor \frac{5}{2} \rfloor \oplus (1 \times 2) \oplus 3 = 3$ 。

沒有相同的困難度，因此輸出 -1 。

7_典型的 TOI 題目 (TOI_Problem)

(15分)

時間限制: 1 seconds

記憶體限制: 256 MB

題目敘述

LDC 正在準備一場 TOI 模擬賽，她最近想到了一道符合 TOI 風格的題目。題目敘述如下：

有 N 個正整數對 a_i, b_i ($1 \leq i \leq N$)。有多少從正整數映射到正整數的函數 f 滿足以下兩個條件：

- 對於所有正整數 i 滿足 $1 \leq i \leq N$ ， $f(a_i) = b_i$
- 對於所有正整數 x, y ， $f(x) \cdot \min(x, f(y)) = x \cdot f(\min(x, y))$ 皆成立

如果答案是無限大，輸出 `-1`，否則輸出答案模 998244353 的餘數。

LDC 現在找你做這題，以估計難度。請你寫出能做出這題的程式。

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 。

對於所有整數 i 滿足 $1 \leq i \leq N$ ，測試資料第 $i + 1$ 行為兩個正整數 a_i, b_i ，兩數之間以一個空格分開。

輸出格式

對於每一筆測試資料輸出一行。如果答案是無限大，輸出 `-1`，否則輸出答案模 998244353 的餘數。

資料範圍

- $1 \leq T \leq 2 \cdot 10^5$
- $1 \leq N \leq 2 \cdot 10^5$
- $1 \leq a_i, b_i \leq 10^{18}$
- $a_i \neq a_j (i \neq j)$
- 在一筆輸入檔案中，所有測試資料的 N 的總和 $\leq 2 \cdot 10^5$

測試範例

輸入範例 1

```
2
1
1 1
4
8 21
7777714 49
625 52725
10000000000000000000 864197532
```

輸出範例 1

```
1
0
```

範例說明

對於第一筆範例，只有 $f(x) = x$ 滿足所有條件。
答案記得要模 998244353。

8_保護費 (Protection_Fees)

(5分 / 15分)

時間限制: 3 seconds

記憶體限制: 512 MB

問題敘述

身為一個成熟的大學生，想要在校園裡舒適的生活，勢必要向各方勢力繳納高額的保護費，你繳的越多，你的大學生活就會越安穩。

現在你身為一位國立清寒大學的新生，進到學校的第一件事當然是想辦法繳保護費。你知道學校裡有多方勢力盤據，且你的學校是可以用一張簡單無向圖 G 來表示，每個節點都是一個領地且一定會被恰一方勢力所佔據，領地之間有一些邊互相連接。如果你不小心同時繳納保護費給兩個勢力且有一條邊橫跨兩方勢力的領地，他們將會發生衝突，而你將非常慘。

所以繳納保護費也是一件充滿學問的大事，一個不小心就會讓你的大學生活陷入悲劇。

清寒大學裡總共有 K 方勢力，編號由 1 到 K 。想要請第 t 方勢力保護你，就必須繳納 c_t 金額的保護費。

因為你認為繳的越多就得到越多，所以請你求出在不發生衝突的情況下，你最多可以繳納多少保護費。

輸入格式

第一行輸入三個非負整數 N 、 M 、 K ，代表無向圖 G 的點數、邊數和學校中的勢力個數。

第二行輸入 N 個正整數 a_1, a_2, \dots, a_N ，分別代表佔據 i 號節點的勢力。

在接下來的 M 行中，每行會包含兩個正整數 u_i 、 v_i ，代表第 i 條邊連接著圖 G 上 u_i 與 v_i 兩個點。

最後一行輸入 K 個正整數 c_1, c_2, \dots, c_K ，分別代表 i 號勢力所收取的保護費金額。

輸出格式

輸出一個整數，代表不發生衝突的情況下，能繳納的保護費金額最大值。

資料範圍

- $1 \leq N \leq 100\,000$ 。
- $0 \leq M \leq 100\,000$ 。
- $1 \leq K \leq 44$ 。
- $1 \leq a_i \leq K$ ($1 \leq i \leq N$) 。
- $1 \leq u_i, v_i \leq N$ ($1 \leq i \leq M$) 。
- $1 \leq c_i \leq 10^9$ ($1 \leq i \leq K$) 。
- 輸入的圖是簡單圖，亦即不存在自環或重邊。

子任務

- 子任務 1 (5 points) $K \leq 20$ 。
- 子任務 2 (15 points) 無額外限制。

範例

輸入範例 1

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

輸出範例 1

```
4
```

範例說明

在範例1中，我們可以繳交保護費給 1, 3 號勢力，以繳納最多保護費且不發生衝突

9_費馬最後定理 (Fermat_Last_Theorem)

(1分 / 2分 / 4分 / 13分)

時間限制: 5 seconds

記憶體限制: 512 MB

問題敘述

古希臘的阿基米德得知現代德國的高斯能夠快速求解 $1 \sim N$ 的總和，深感不服。

於是，阿基米德將問題升級，改為求解 $\sum_{i=1}^N \lfloor \frac{i}{N} \rfloor$ 的和。為此，他需要先證明對於任何有理數，都存在一個比它小的整數。透過這個定理，他證明了著名的阿基米德定理。

然而，不久之後，阿基米德發現答案永遠是一個定值。為增加難度，他改變了分數的上下位置，要求求解 $\sum_{i=1}^N \lfloor \frac{N}{i} \rfloor$ 。

當他剛把這個式子寫下時，牛頓路過，並利用 $\lfloor \frac{N}{i} \rfloor$ 的取值以及富比尼定理秒殺了這個問題。之後，他利用這個結果證明了 $\int_{i=1}^{\infty} \frac{1}{x} dx$ 發散，最終發明了微積分。

不願落後的阿基米德再次提高難度，變成了給定 Q 筆詢問，每次給定 L 和 R ，要求 $\sum_{i=L}^R \lfloor \frac{N}{i} \rfloor$ 。但英國的凱萊認為這仍然不夠美感，大筆一揮，將問題改為給定 N 個矩陣 M_i ，每次詢問要求計算 $\prod_{i=L}^R M_i$ 的結果。

明天要去決鬥伽羅瓦聽聞了這個問題，但他覺得在整數下的運算太無聊了，因此將加法和乘法改為在 $GF(2)$ 下的運算，也就是 $a + b$ 變成 $(a + b) \bmod 2$ ， $a \times b$ 變成 $(a \times b) \bmod 2$ 。

題目漸漸變得完整：給定 N 個矩陣 M_0, M_1, \dots, M_{N-1} ，接下來會有 Q 筆詢問，每次詢問 L 和 R ，要求計算 $\prod_{i=L}^R M_i$ 的結果，並且其中的運算是在模 2 下進行的。

在大家忙著計算的時候，身為業餘數學家的費馬負責將算出來的答案記下來。但因為詢問數量實在太多，幾乎用光他筆記本中所有空白處。為此他要大家將答案全部 XOR 起來後再告訴他。

可是，最終筆記本剩餘的空間還是太小，因此儘管費馬已經對費馬最後定理找到了一個精妙的證明，卻依然沒有足夠的空位寫下。

而三百多年後，英國數學家懷爾斯才發現費馬需要的空頁超過一百頁，就是後話了。

輸入格式

輸入的第一行包含兩個正整數 N 和 k ，代表矩陣的數量和每個矩陣的大小。

接下來的 N 個區塊中，每個區塊包含 k 行，描述了一個 $k \times k$ 的矩陣 M_i 。

接下來一行包含一個正整數 Q ，表示詢問的數量。

接下來兩行，分別包含三個正整數 A_1, B_1, x_0 和 A_2, B_2, y_0 ，表示生成詢問區間的公式。

對於 $i = 1, 2, \dots, Q$ ，詢問的區間 $[L_i, R_i]$ 可以由以下方式生成：

$$x_i = (A_1 \times x_{i-1} + B_1) \bmod (10^9 + 7)$$

$$y_i = (A_2 \times y_{i-1} + B_2) \bmod (10^9 + 7)$$

$$L_i = \min(x_i \bmod N, y_i \bmod N)$$

$$R_i = \max(x_i \bmod N, y_i \bmod N)$$

輸出格式

令 $P_i = \prod_{j=L_i}^{R_i} M_j$ ，其中運算是在模 2 下進行的。

請輸出 $\bigoplus_{i=1}^Q P_i$ ，其中 \oplus 是 XOR 運算。

資料範圍

- $1 \leq N \leq 3 \times 10^5$ 。
- $1 \leq k \leq 3$ 。
- $0 \leq M_i$ 裡的元素 ≤ 1 ($0 \leq i \leq N - 1$)。
- $1 \leq Q \leq 3 \times 10^7$ 。
- $1 \leq A_1, B_1, x_0, A_2, B_2, y_0 < 10^9 + 7$ 。

子任務

- 子任務 1 (1 point) $N, Q \leq 3000$ 。
- 子任務 2 (2 points) $N, Q \leq 3 \times 10^5$ 。
- 子任務 3 (4 points) $Q \leq 3 \times 10^6$ 。
- 子任務 4 (13 points) 無額外限制。

範例

輸入範例 1

```
3 3
000
000
000
111
111
111
100
000
000
3
1 1 2
1 1 2
```

輸出範例 1

```
011
111
111
```

輸入範例 2

```
3 2
10
00
01
01
10
00
3
604078599 524693017 85841332
671146413 219874579 658380675
```

輸出範例 2

```
10
00
```

範例說明

輸入範例 1 的矩陣如下：

$$M_1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, M_2 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}, M_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (1)$$

輸入範例 2 中的矩陣如下：

$$M_1 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, M_2 = \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix}, M_3 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \quad (2)$$

請注意， \oplus 運算符號表示位元 XOR 運算。

對於兩個 $k \times k$ 的矩陣 U 、 V ，若 $W = U \oplus V$ ，則對於 $1 \leq i \leq k$ 、 $1 \leq j \leq k$ ：

$$W[i][j] = (U[i][j] \oplus V[i][j])$$

10_BST (Building_Spanning_Tree)

(2分 / 2分 / 16分)

時間限制: 3 seconds

記憶體限制: 512 MB

問題敘述

BST 通常會是 Binary Search Tree 的縮寫，但在這一題中，卻是指 Building Spanning Tree。

給定一張有 N 個點與 M 條邊的簡單無向圖 G ，以及其上的一棵生成樹 T ，其中點的編號為 $1 \sim N$ ，並且對於 $i = 1 \sim M$ ，第 i 條邊為 (u_i, v_i) 。

要求構造一個邊的排列方式 S ，滿足以下條件：

1. 每條 G 中的邊在 S 中恰好出現一次。
2. 維護一張有 N 個點的圖 H ，點的編號為 $1 \sim N$ ，且一開始 H 中所有點之間都沒有邊相連。將 S 中的邊 (a_j, b_j) 依序取出來檢查，若 a_j 、 b_j 尚不連通，就將這條邊加入 H 中，否則略過這條邊。要求最終 H 等於 T 。
3. 在滿足上述條件的邊序列中， S 的字典序最小。

請注意，雖然 G 是無向圖，但對於每一條題目給定的邊 (u_i, v_i) ，在邊序列中不能夠將端點的順序交換。舉例來說，若題目給了一條邊 $(2, 1)$ ，只能夠將其視為 $(2, 1)$ 、而不能夠視為 $(1, 2)$ 。

關於字典序的定義如下：

- 對於兩個長度為 M 的邊序列 A 、 B ，我們說 A 的字典序比較小，若且唯若存在 k 滿足 $A_k < B_k$ ，且對於所有的 $p < k$ 都有 $A_p = B_p$ 。
- 對於兩條邊 (u_i, v_i) 、 (u_j, v_j) ，我們說 $(u_i, v_i) < (u_j, v_j)$ ，若且唯若 $u_i < u_j$ 或者是 $u_i = u_j$ 且 $v_i < v_j$ 。

輸入格式

輸入的第一行會包含兩個正整數 N 、 M ，代表圖 G 的點數與邊數。

在接下來的 M 行中，每行會包含兩個正整數 u_i 、 v_i ，代表第 i 條邊連接著圖 G 上 u_i 與 v_i 兩個點。

最後一行會包含 $N - 1$ 個正整數 k_1, k_2, \dots, k_{N-1} ，代表 T 由 $(u_{k_1}, v_{k_1}), (u_{k_2}, v_{k_2}), \dots, (u_{k_{N-1}}, v_{k_{N-1}})$ 這些邊所組成。

輸出格式

請輸出題目要求的序列 S ，每條邊佔據一行。

資料範圍

- $2 \leq N \leq 100\,000$ 。
- $N - 1 \leq M \leq \min\left(200\,000, \frac{N \times (N - 1)}{2}\right)$ 。
- $1 \leq u_i, v_i \leq N$ ($1 \leq i \leq M$)。

- $u_i \neq v_i$ ($1 \leq i \leq M$) 。
- $1 \leq k_j \leq M$ ($1 \leq j \leq N - 1$) 。
- 保證 G 是簡單連通圖（不存在重邊及自環）, 並且 T 是 G 的生成樹。

子任務

- 子任務 1 (2 points) $N \leq 500$ 、 $M \leq 500$ 。
- 子任務 2 (2 points) $M \leq N + 500$ 。
- 子任務 3 (16 points) 無特別限制。

範例

輸入範例 1

```
5 8
5 2
2 3
3 4
4 1
1 2
3 5
3 1
4 5
1 2 3 4
```

輸出範例 1

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

輸入範例 2

```

6 7
1 2
5 3
4 6
2 4
3 1
6 5
1 6
1 2 3 4 5

```

輸出範例 2

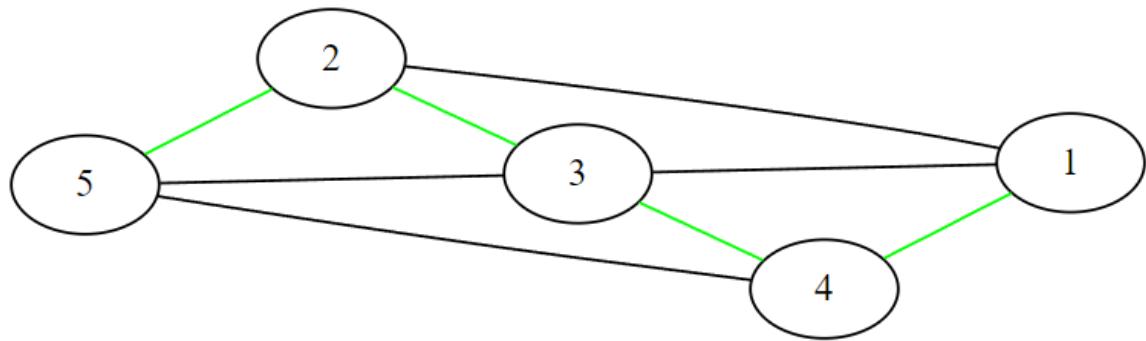
```

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

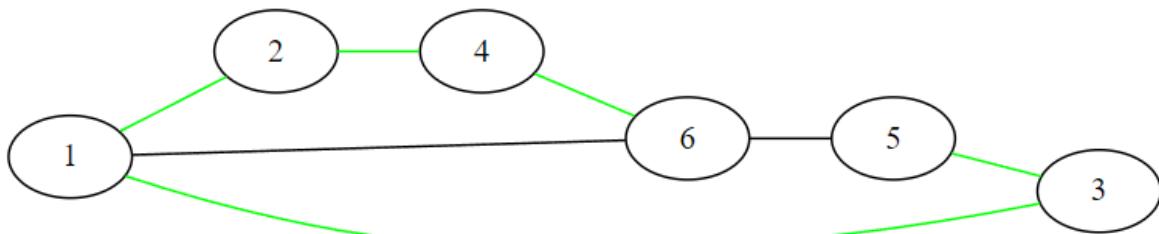
```

範例說明

第一筆測資中的圖如下所示，其中綠色的邊代表生成樹 T 上的邊。



第二筆測資中的圖如下所示，其中綠色的邊代表生成樹 T 上的邊。



11_我推的字串 (Oshi_No_String)

(20分)

時間限制: 3 seconds

記憶體限制: 512 MB

題目敘述

紗利奈是一個對身邊的各種事物都充滿興趣的女孩，她對電視上的偶像充滿興趣，也對每天幫她看病的醫生充滿興趣。最近她開始接觸了各種字串，並對它們產生了興趣。她發現，在許多不同的字串當中，有些字串是她特別喜歡的。

在接觸了許多字串後，紗利奈發現了對於字串的喜歡規則：

- 越長的字串，紗利奈越喜歡
- 對於同樣長度的兩個字串，紗利奈喜歡字典序比較小的那一個字串
- 紗利奈不喜歡同一個字母出現超過 k 次的字串

特別地，紗利奈會對照以下字元順序作為字典

序： abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789_? 。

在得知自己的喜好後，紗利奈希望對於一個字串，她能夠提取出一段她最喜歡的字串作為她推的字串。她提取字串的方法如下：對於一個長度為 n 的字串 S ，選擇一對 $[l, r]$ 作為字串的開頭及結尾。如果 $l \leq r$ ，那她會提取 $s_l s_{l+1} \dots s_r$ ，否則的話，她會提取 $s_r s_{r-1} \dots s_n s_1 \dots s_l$ 。

由於紗利奈只想要專心的推字串，她希望你能幫忙她找出每個字串中，她推的字串。

輸入格式

第一行包含一個 T ，代表接下來有 T 筆測資。

每筆測資有一行，包含一個字串 S 和一個正整數 k ，如題目所敘。

輸出格式

對於每筆測資，輸出符合題目要求的字串。

資料範圍

- $1 \leq k \leq |S| \leq 3 \times 10^5$ ， $|S|$ 代表字串 S 的長度
- $T, \sum |S| \leq 3 \times 10^5$

測試範例

輸入範例 1

```
3
cbadcabd 1
aaaedbaaaceea 2
hoshinoaiiii 2
```

輸出範例 1

```
abdc
aacee
hoshinoai
```

輸入範例 2

```
1
woF5ZIpk_?MDOVGq9ES02yKc7iRTP41zBaCXjxuUn3tmh8bWH6sLrYAN1QgdJvef 2
```

輸出範例 2

```
aCXjxuUn3tmh8bWH6sLrYAN1QgdJvefwoF5ZIpk_?MDOVGq9ES02yKc7iRTP41zB
```

輸出範例 3

```
rasu_mediaMuteki_no_egao
...(省略)...
shiteruA
```

範例說明

在範例 1 中，`cbadcabd` 可以提取出長度最長的子字串長度為 4，有 `cbad`, `badc`, `dcab`, `cabd`, `abdc`, `acba`，其中字典序最小的子字串為 `abdc`。

在範例 2 中，任意一個子字串都能被提取，而符合要求的子字串為：`aCXjxuUn3tmh8bWH6sLrYAN1QgdJvefwoF5ZIpk_?MDOVGq9ES02yKc7iRTP41zB`。

範例 3 之完整測資，可於下載附件中的TestCases資料夾取得。

12_給你的禮物 (Present_for_You)

(4分 / 16分)

時間限制: 3 seconds

記憶體限制: 256 MB

題目敘述

Vivy 手上有一個正整數 N 和實數 $S = \sqrt{1} + \sqrt{2} + \dots + \sqrt{N}$ 。

S 可以用許多正整數乘以正整數開根號的總和來表示，Vivy 認為 $(a_1, a_2, \dots, a_k, b_1, b_2, \dots, b_k)$ 是一個合法表示方法，若 $S = \sum_{i=1}^k a_i \sqrt{b_i}$ 且對於所有 $1 \leq i \leq k$, a_i, b_i 為正整數。

Vivy 覺得一個合法表示方法 $(a_1, a_2, \dots, a_k, b_1, b_2, \dots, b_k)$ 的漂亮程度為 $\sum_{i=1}^k a_i^2 b_i$ 。

Vivy 想知道所有 S 的合法表示方法中，漂亮程度的最大可能值 M 除以 21610411 的餘數為多少，可是她忙著阻止 ChatGPT 消滅全人類，沒空把這個數字算出來。身為一個正在被她拯救的人類，幫她算出這個數字當作是給她的謝禮吧。

輸入格式

輸入只有一個正整數 N 。

輸出格式

輸出一個整數，代表 $M \bmod 21610411$ 。

資料範圍

- $1 \leq N \leq 10^{12}$

子任務

- 子任務 1 滿足 $N \leq 10^6$ (4分)
- 子任務 2 沒有其他限制 (16分)

測試範例

輸入範例 1

20

輸出範例 1

356

範例說明

可以證明當 S 被表示成，

$$\begin{aligned} S = & 10\sqrt{1} + 6\sqrt{2} + 3\sqrt{3} + 3\sqrt{5} + \sqrt{6} + \sqrt{7} + \sqrt{10} + \sqrt{11} \\ & + \sqrt{13} + \sqrt{14} + \sqrt{15} + \sqrt{17} + \sqrt{19} \end{aligned}$$

漂亮程度會有最大值 356。

13_結束樂團出遊！ (Kessoku_Band_at_Enoshima)

(10分/15分)

時間限制: 10 seconds

記憶體限制: 512 MB

題目敘述



終於放暑假了！一想到接下來有一個多月的時間可以不用去學校社交，小孤獨就不禁期待起接下來可以天天待在家裡當吉他英雄的生活。不過，更令小孤獨期待的是，這個暑假她交到朋友了！有了朋友，就代表她終於可以體驗到，在暑假跟著朋友們一起出去玩的生活了！就這樣，懷著這樣的期待，小孤獨迎來了屬於她的第一個現充的暑假...。

時間來到開學前一天，經過了一整個暑假，結束樂團的朋友都沒有來找小孤獨出去玩，這讓她難過到開始反向演化。當喜多一行人發現小孤獨時，她已經退化成一條蛞蝓了！為了讓小孤獨能夠順利進化回人類，結束樂團的一行人決定一起前往江之島，享受最後一天的暑假！

來到江之島後，結束樂團的成員們想要盡快前往各種景點遊玩。不過，退化成蛞蝓的小孤獨只剩下了南北移動和東西移動的能力。因此，如果結束樂團想從 (x_1, y_1) 移動到位於 (x_2, y_2) 的景點的話，他們得移動 $|x_1 - x_2| + |y_1 - y_2|$ 單位的距離。現在地圖上有 N 個景點，他們希望你能隨時告訴他們，距離他們所在位置最近的景點有多遠。

除此之外，喜多在SNS上的粉絲還會時不時推薦其他江之島的景點他們，這會讓他們有機會移動更短的距離就抵達某個江之島的景點。

現在，告訴你 N 個景點的位置，接下來有 Q 個事件要你處理，一個事件可能包含 1 $x\ y$ ，代表結束樂團現在在 (x, y) ，然後他們想知道最少要移動多少距離才能抵達一個景點。或是包含 2 $x\ y$ 代表喜多的粉絲告訴他們有一個新的景點，保證這個景點是新的景點。

輸入格式

第一行包含兩個正整數 N, Q 。

接下來會有 N 行，每行會有兩個整數 x_i, y_i 代表第 i 個景點的位置。

接下來會有 Q 行，每行代表一個事件 $1\ x_j\ y_j, 2\ x_j\ y_j$ ，如題目所敘。

輸出格式

對於每個 $1\ x_j\ y_j$ 輸出一個正整數，代表當下從 (x_j, y_j) 移動到某個景點所需的最短距離。

資料範圍

- $1 \leq N, Q \leq 3 \times 10^5$
- $-5 \times 10^4 \leq x_i, y_i, x_j, y_j \leq 5 \times 10^4$
- 保證 $N + 2$ $x_j\ y_j$ 事件數量 $\leq 3 \times 10^5$

子任務

- 子任務 1 滿足只會有 $1\ x_j, y_j$ 事件 (10分)
- 子任務 2 沒有其他限制 (15分)

測試範例

輸入範例 1

```
5 5
0 4
-2 1
-3 -1
2 1
3 -1
1 -3 3
1 3 2
1 0 0
2 0 1
1 0 0
```

輸出範例 1

```
3
2
3
1
```

輸入範例 2

```
5 7
-3 3
0 0
3 -3
3 3
-3 -3
1 1 2
1 -2 2
1 0 -2
2 0 -3
2 0 3
1 1 2
1 0 -2
```

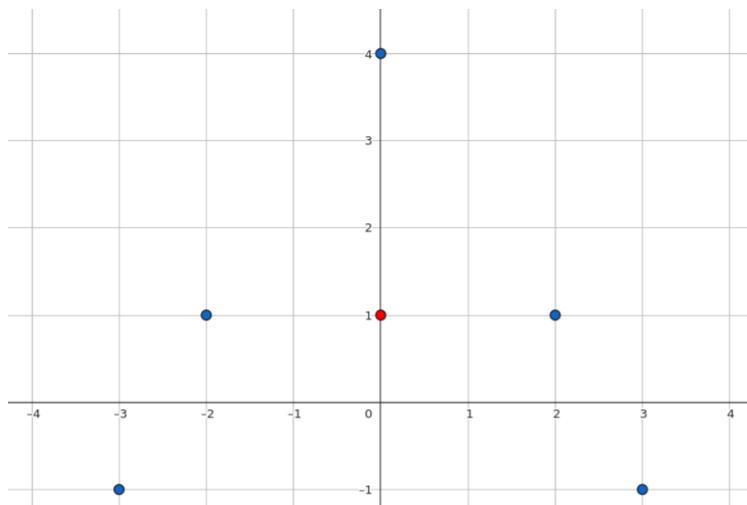
輸出範例 2

```
3
2
2
2
1
```

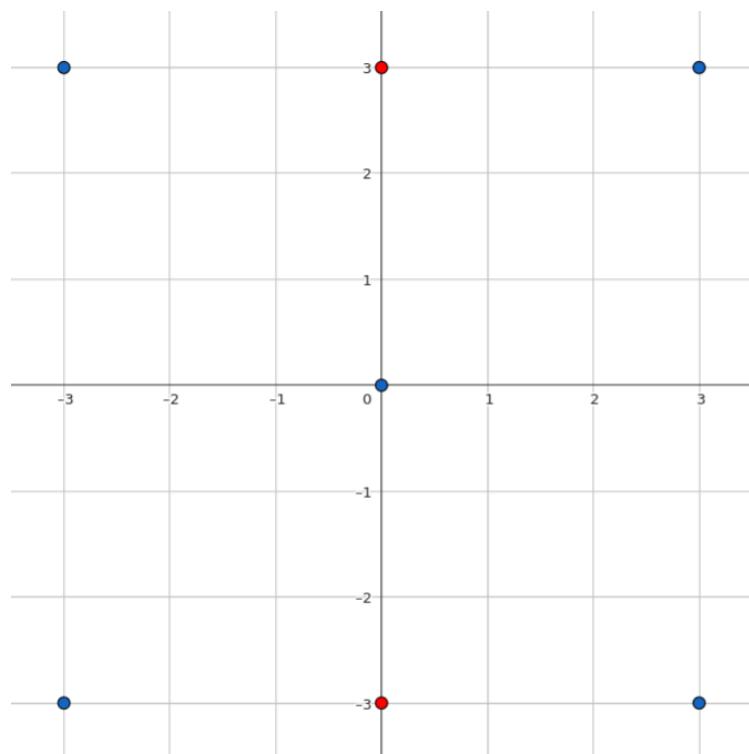
範例說明

以下是測試範例的示意圖，藍點為一開始的 N 個景點，紅點為喜多粉絲提供的新景點。

- 測試範例 1：



- 測試範例 2 :



14_IQ 999

(2分/1分/7分/4分/11分)

時間限制: 2 seconds

記憶體限制: 512 MB

(10 points/15 points)

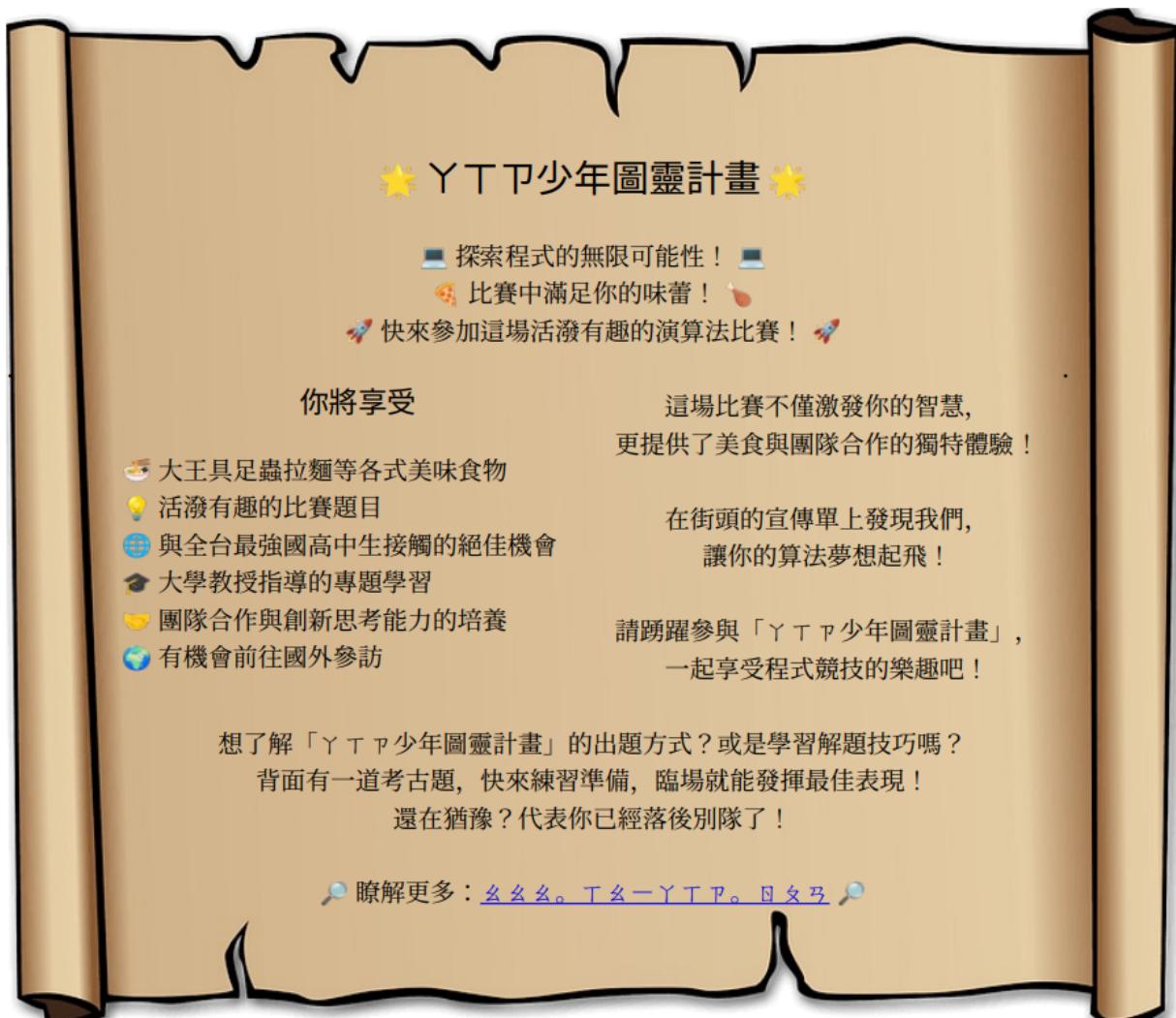
Time Limit: 2 second

Memory Limit: 1024 MB

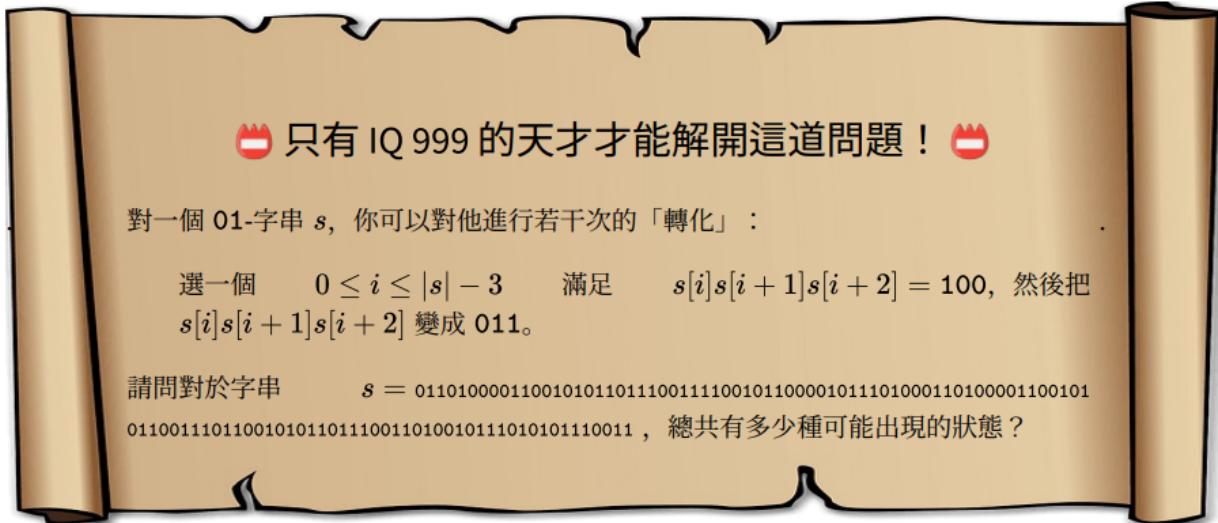
問題敘述

2024 年 5 月 13 日，月曜日，天氣晴，宜燒水。

へにゃ的煮水壺壞掉了，於是她出了遠門去買煮水壺。在回家的路上，她在電線桿上發現了一張來自 YTP 的宣傳單：



へにゃ把傳單翻到背面，果然發現了一道問題：



「這還不簡單，當然是 172 800 dayo！」ヘニヤ心道。

ヘニヤ決定拿這道問題來考考你。她覺得題目不能就這麼簡單，於是讓字串長度可以到 2 000 000 個字元這麼長。

「好像還是太簡單了，那來加上修改操作好了！」她決定。

所以接下來還會有 q 次單點修改 $s[p[i]]$ 字元的操作（1 變 0、0 變 1），請在所有操作前以及每次操作完後輸出該字串透過「轉化」能出現的狀態數量，並模 998 244 353 後輸出。

請注意修改操作是永久的，**並非**互相獨立。

輸入格式

- line 1: s
- line 2: q
- line 3: $p[0] \ p[1] \ \dots \ p[q - 1]$
- 如果 $q = 0$ 則第三行輸入為空。

輸出格式

- line 1: 所有修改操作前的答案。
- line $2 + i$ ($0 \leq i \leq q - 1$) : 第 $(i + 1)$ 次修改操作之後的答案。

資料範圍

- $1 \leq |s| \leq 2\,000\,000$ 。
- $s[i] \in \{0, 1\}$ ($0 \leq i \leq |s| - 1$)。
- $0 \leq q \leq 200\,000$ 。
- $0 \leq p[i] \leq |s| - 1$ ($0 \leq i \leq q - 1$)。

子任務

- 子任務 1 : $|s| \leq 15$ 、 $q \leq 100$ 。
- 子任務 2 : $s[0] = 1$ 、 $s[i] = 0$ ($1 \leq i \leq |s| - 1$) 、 $q = 0$ 。
- 子任務 3 : $|s| \leq 3000$ 、 $q \leq 3000$ 。
- 子任務 4 : 所有修改操作都是將 1 變成 0。
- 子任務 5 : 無額外限制。

範例

輸入範例 1

```
010100101010001  
8  
1 11 4 9 3 7 0 11
```

該範例輸入符合子任務 1, 3, 5 的限制。

輸出範例 1

```
12  
8  
4  
2  
2  
2  
2  
4  
4
```

輸入範例 2

```
1111111  
7  
6 5 3 1 4 2 0
```

該範例輸入符合子任務 1, 3, 4, 5 的限制。

輸出範例 2

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

輸入範例 3

```
00011111000010101010001111010
12
26 6 29 11 11 28 11 26 5 13 17 26
```

該範例輸入符合子任務 3, 5 的限制。

輸出範例 3

```
15
30
30
30
24
30
30
24
24
48
72
104
104
```

輸入範例 4

```
01101000011001010110110011110010110000101110100011010000110010101100111011001010110111
0011010010111010101110011
0
```

該範例輸入符合子任務 3, 5 的限制。

請注意除了最後代表 q 的 0 之外，其他字元實際上位於同一行（可以參考附件中的 `04.in.txt`）。

輸出範例 4

172800

範例說明

在範例 1 中，在所有操作都結束之後，字串會變成 100010111110001。所有可以透過若干次轉化到達的狀態有下列 4 種：

- 100010111110001
- 100010111110001 → 011010111110001
- 100010111110001 → 011010111110001 → 011010111101101
- 100010111110001 → 100010111101101

在範例 2 中，所有操作之前字串的樣子是 1111111，已經無法再進行轉化，故答案為 1 種。而在第 5 次操作結束之後，字串會變成 1010000。所有可以透過若干次轉化到達的狀態有下列 5 種：

- 1010000
- 1010000 → 1001100
- 1010000 → 1001100 → 0111100
- 1010000 → 1001100 → 0111100 → 0111011
- 1010000 → 1001100 → 1001100 → 1001011

輸入範例 4 與宣傳單背後的字串相同，而へにゃ也展現了她的 IQ 999。

15_Nachoneko 算數學 (Nachoneko_and_the_Math_Homework)

(3分/ 7分/ 13分/ 2分)

時間限制: 1.5 seconds

記憶體限制: 512 MB

問題敘述

Nachoneko 媽媽畫圖畫累了，於是她撿起了被女兒 Gura 丟在旁邊的亞特蘭提斯國度的數學作業來寫寫看。

Nachoneko 知道自己和女兒都不擅長數學，但她想藉著這個機會來挑戰自己，同時也希望能夠和女兒一起解決這些問題。她坐在桌前，打開數學書本，開始閱讀題目。

然而很快地，Nachoneko 就卡住了。亞特蘭提斯國度的問題讓她覺得非常陌生，她不知道該從何下手。就在這時，她立刻想到了你，一位善於數學的朋友，於是她決定來請求你的幫助。她向你解釋了她和女兒在數學上的困惑，並希望你能指導她們。

定義一個將陣列映射到一個整數的函數 f 如下：

$$f(B) = \sum_{0 \leq i \leq j \leq |B|-1} \frac{b[i] + b[i+1] + \dots + b[j]}{\gcd\{b[i], b[i+1], \dots, b[j]\}} \quad (5)$$

其中， \gcd 代表的是最大公因數 (greatest common divisor) 。

給你一個長度為 n 的陣列 A ，並定義 $A \setminus a[i]$ 為 $[a[0], a[1], \dots, a[i-1], a[i+1], \dots, a[n-1]]$ 。請你對於每個 $i = 0, 1, \dots, n-1$ 都找出 $f(A \setminus a[i])$ 的值。因為答案可能很大，所以請將答案模 998 244 353 後輸出。

輸入格式

- line 1: n
- line 2: $a[0] \ a[1] \ \dots \ a[n-1]$

輸出格式

- line $1+i$ ($0 \leq i \leq n-1$) : $f(A \setminus a[i]) \bmod 998\,244\,353$

資料範圍

- $2 \leq n \leq 80\,000$ 。
- $1 \leq a[i] \leq 998\,244\,352$ ($0 \leq i \leq n-1$) 。
- 輸入的數字都是整數。

子任務

- 子任務 1 : $n \leq 300$ 。
- 子任務 2 : $n \leq 2000$ 。

- 子任務 3： $n \leq 16\,000$ 。
- 子任務 4：無額外限制。

範例

輸入範例 1

```
4
2 4 6 3
```

該範例輸入符合全部子任務的限制。

輸出範例 1

```
24
21
22
17
```

輸入範例 2

```
10
36 32 27 16 64 30 20 35 42 24
```

該範例輸入符合全部子任務的限制。

輸出範例 2

```
4587
4563
3979
5058
4002
4774
5032
4153
4521
4599
```

輸入範例 3

```
6
249561088 536870912 719323136 998244352 285212672 606076928
```

該範例輸入符合全部子任務的限制。

輸出範例 3

```
6793
5045
5138
6725
8083
7588
```

範例說明

在範例 1 中：

第一個數字是 $f(A \setminus a[0])$ 的值，也就是 $f([4, 6, 3])$ 的值。計算過程如下所示：

$$\begin{aligned} f([4, 6, 3]) &= \frac{4}{\gcd\{4\}} + \frac{4+6}{\gcd\{4, 6\}} + \frac{4+6+3}{\gcd\{4, 6, 3\}} + \frac{6}{\gcd\{6\}} + \frac{6+3}{\gcd\{6, 3\}} + \frac{3}{\gcd\{3\}} \\ &= 1 + 5 + 13 + 1 + 3 + 1 \\ &= 24 \end{aligned}$$

第三個數字是 $f(A \setminus a[2])$ 的值，也就是 $f([2, 4, 3])$ 的值。計算過程如下所示：

$$\begin{aligned} f([2, 4, 3]) &= \frac{2}{\gcd\{2\}} + \frac{2+4}{\gcd\{2, 4\}} + \frac{2+4+3}{\gcd\{2, 4, 3\}} + \frac{4}{\gcd\{4\}} + \frac{4+3}{\gcd\{4, 3\}} + \frac{3}{\gcd\{3\}} \\ &= 1 + 3 + 9 + 1 + 7 + 1 \\ &= 22 \end{aligned}$$

在範例 3 中，請記得將輸出對 998 244 353 取模。

後記

在你的引導下，Nachoneko 和 Gura 一起解決了一個又一個亞特蘭提斯國度的數學問題。他們互相鼓勵，彼此分享解題心得，並且在你的幫助下克服了他們的困難。

最終，他們完成了整份數學作業。Nachoneko 看著自己手中的作業感到非常自豪，而 Gura 也對自己的進步感到十分驕傲。他們知道，只要有耐心和正確的引導，他們就可以克服任何數學難題。