

# 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！

## 問題描述

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 “!”

## 資料範例

輸入範例 1  
(無輸入值)

輸出範例 1  
`Hello World!`

## 範例解釋

輸入範例 1, 無輸入值，簡單而快樂的輸出 `Hello World!`

# 問題 1 – 熵用魔法 (Entropy Magic)

(5 分)

## 問題敘述

作為一個物理大師，小 Y 盡力想避免這個宇宙產生「熱寂」。他從小 P 那邊學到了一個可以降低熵的魔法，希望能夠拯救整個世界。但是這個魔法的使用存在某些限制，為了簡單起見，我們將整個宇宙視為一個長度為  $N$  的序列  $a_1, a_2, a_3, \dots, a_N$ ，每一個元素都是 0 或 1，這個魔法可以選擇一個長度為 3 且並非全部都是 0 或 1 的區間  $a_i, a_{i+1}, a_{i+2}$  ( $1 \leq i \leq N - 2$ )，之後將其換成  $a_{i+1} \oplus a_{i+2}, a_i \oplus a_{i+2}, a_i \oplus a_{i+1}$ ，其中  $\oplus$  代表的是 XOR，他的運算規則如下：

$a$	$b$	$a \oplus b$
0	0	0
0	1	1
1	0	1
1	1	0

小 Y 想問你是否可以使用這個魔法將整個序列變成 0，把熵降到最低。可以證明如果能用魔法將整個序列變成 0，那麼必定可以使用  $2N$  次以內的魔法就達成，因此假如可以用這個魔法將整個序列變成 0，請輸出一個長度不超過  $2N$  的操作序列，如果有很多個操作序列滿足條件，請輸出任意一個。

## 輸入格式

輸入的第一行包含一正整數  $N$ ，代表小 Y 將整個宇宙視為一個長度為  $N$  的序列。

下一行包含一個長度為  $N$  的序列  $a_1, a_2, a_3, \dots, a_N$ ，代表宇宙被轉換成的序列。

## 輸出格式

第一行輸出 “YES” 或 “NO” 代表是否可以使用這個魔法將整個序列變成 0。

如果是 “YES” 的話，第二行輸出一個整數  $M$ ，代表你可以使用恰好  $M$  次魔法將整個序列變成 0。第三行請輸出  $M$  個正整數  $b_i$ ，代表第  $i$  個操作會對  $a_{b_i}, a_{b_i+1}, a_{b_i+2}$  使用魔法。如果有多種使用魔法的方式可以滿足條件，請輸出任意一個。

如果  $M > 2N$  或是依序使用這些魔法並不會讓整個序列變成 0，你將會獲得 Wrong Answer。

## 資料範圍

- $3 \leq N \leq 10^6$
- $0 \leq a_i \leq 1$

### 輸入範例 1

5  
0 0 0 0 0

### 輸出範例 1

YES  
0

### 輸入範例 2

4  
1 1 1 1

### 輸出範例 2

NO

### 輸入範例 3

3  
0 1 0

### 輸出範例 3

NO

### 輸入範例 4

3  
0 0 1

### 輸出範例 4

NO

## 範例說明

在範例 1 中，不需要使用任何魔法就可以讓序列的每一項都是 0。

在範例 2 中，不存在任何一個長度為 3 且並非全部都是 0 或 1 的區間，因此完全無法使用魔法。

在範例 3 中，使用魔法將會使序列變成  $010 \rightarrow 101 \rightarrow 101 \rightarrow \dots$ ，永遠不會到達 000。

## 問題 2 - 找重複! (Find Duplicate!)

---

(10 分)

### 問題敘述

請在一個有  $N$  個數字的數字列裡面尋找重複的數字。每遇到一個重複，就輸出該數字。

### 輸入格式

一排以空白分隔的  $N$  個數字。

### 輸出格式

一排用空白分隔的數字，依照重複出現的順序列出，每重複一次輸出一個。

### 資料範圍

$2 < N < 10,000,000$

數列中的每個數字介於  $0 \sim N$  之間

執行時間限制 10s

### 資料範例

#### 輸入範例 1

1 3 5 2 7 3 4

#### 輸出範例 1

3

#### 輸入範例 2

3 10 15 2 5 30 5 10 7 8 9 100 5

#### 輸出範例 2

5 10 5

### 範例說明

範例 1 的數字列中，3 重複了 1 次，其它數字沒重複，所以輸出只有一個 3。

範例 2 裡面有 10 跟 5 有重複，其中 5 重複了兩次。依照重複的順序 5 先重複了，之後碰到重複的 10，最後 5 又重複一次，所以輸出是 5 10 5（以空白間隔）。

## 問題 3 – 海龜繪圖 (Turtle Draw)

(10 分)

### 問題敘述

海龜繪圖是電腦輔助教學早期就研製的系統，後來也在 scratch, SVG 等都有看到類似概念的東西，海龜繪圖是一個與數學座標系相同的作圖系統，往右代表 x 軸增加，往上代表 y 軸增加（電腦繪圖則是往下代表 y 軸增加），並利用路徑的方式去描述作圖的位置，由原圖 (0, 0)，水平朝右邊出發，請寫一個程式依照輸入的指令和值，輸出計算後最終的 x, y 值。

#	指令	說明	值的範圍
1	up(n)	Y 值增加 n (垂直往上)	$n=[0, 1, \dots, 100]$
2	down(n)	Y 值減少 n (垂直往下)	$n=[0, 1, \dots, 100]$
3	left(n)	X 值減少 n (水平往左)	$n=[0, 1, \dots, 100]$
4	right(n)	X 值增加 n (水平往右)	$n=[0, 1, \dots, 100]$
5	turnleft(n)	方向向左轉 n 度	$n=\{15, 30, 45, 60, 75, 90\}$
6	turnright(n)	方向向右轉 n 度	$n=\{15, 30, 45, 60, 75, 90\}$
7	forward(n)	依原本方向前進 n	$n=[0, 1, \dots, 100]$
8	backward(n)	保持原本方向後退 n	$n=[0, 1, \dots, 100]$
9	home	依原本方向，但目前位置移到 (0, 0)	
10	quit	程式結束，並印出最終 x, y 值	

例如：turnleft 30 度再 forward 10, x 和 y 計算如下：

$30 \text{ degree} = (30 \times \text{PI}) / 180 = \text{PI} / 6$  (度度量轉徑度量)

$X = X + 10 * \cos(\text{PI} / 6)$

$Y = Y + 10 * \sin(\text{PI} / 6)$

### 輸入格式

每行有一個指令及 n 值，並以逗號 “,” 間隔。(或僅有指令而無 n 值及逗號)

### 輸出格式

輸出最終的 x, y 值，四捨五入到小數第二位，並以逗號 “,” 隔開。

### 資料範圍

$0 \leq n \leq 10$

### 輸入範例 1

```

up, 3
down, 3
left, 3
quit
    
```

### 輸出範例 1

-3.00,0.00

### 範例 1 說明

up,3 => (0,3)

down,3 => (0,0)

left,3 => (-3,0)

### 輸入範例 2

turnleft,75

forward,5

turnleft,75

forward,50

turnleft,60

forward,20

down,10

left,10

quit

### 輸出範例 2

-69.33,9.83

### 輸入範例 3

turnright,30

backward,30

home

right,50

forward,20

quit

### 輸出範例 3

67.32,-10.00

## 問題 4 – 綠蠨龜地圖 (Turtle Maps)

(10 分)

### 問題敘述

蕭總對於「烏龜會飛」這件事情非常著迷：他投身於綠蠨龜的研究，就是想要看到綠蠨龜在天空中翱翔的一天！他發現，綠蠨龜有三個特性：

- 吃飯很慢
- 講話很嗆
- 很喜歡地圖

發現綠蠨龜們對於地圖情有獨鍾的蕭總決定幫忙綠蠨龜們畫綠蠨龜聚落的地圖。這個地圖可以表示為以  $(0,0)$  和  $(1,1)$  兩個點為對角線的正方形。他們發現，綠蠨龜聚落也可以視為座標平面上的一個矩形，而地圖上的  $(0,0)$ 、 $(1,0)$ 、 $(1,1)$ 、 $(0,1)$  四隅正好依序對應到綠蠨龜聚落的  $(x_1, y_1)$ 、 $(x_2, y_2)$ 、 $(x_3, y_3)$ 、 $(x_4, y_4)$  的四個角落！

聰明的蕭總馬上發現到了：倘若地圖完全包含於綠蠨龜聚落以內的話，則存在一個點  $P$ ，地圖上標示  $P$  點的位子，恰好就是綠蠨龜聚落上的  $P$  點！

而因為蕭總為了  $T$  個綠蠨龜聚落畫了  $T$  張圖，蕭總又很忙很忙，所以他就拜託你幫忙寫一份程式，對於每一份地圖和綠蠨龜聚落，請幫忙找到這個  $P$  點的座標到底為何？

### 輸入格式

第一行有一個正整數  $T$ ，代表有幾個聚落。接下來的  $T$  行，每一行有 8 個以空白間隔的浮點數  $x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4$  依序代表這個綠蠨龜聚落的四個角落。

### 輸出格式

對於每一個聚落，請輸出兩個由空格分隔的浮點數  $X_i, Y_i$  代表  $P$  的座標（ $P$  的定義請見問題敘述）。此外，也保證答案存在且唯一。如果你的答案的兩個座標與正解的答案的兩個座標的相對誤差皆小於等於  $10^{-5}$  就會算正確。

相對誤差：如果答案是  $a$ ，而你輸出  $b$  的話，則你的答案會算是正確的若且唯若  $\min(|a - b|, \frac{|a - b|}{|a|}) \leq 10^{-5}$ 。

### 資料範圍

- $1 \leq T \leq 10^5$
- $|x_i|, |y_i| \leq 10^6$

### 輸入範例 1

```
3
0 0 2 0 2 2 0 2
-0.5 -0.5 1.5 -0.5 1.5 1.5 -0.5 1.5
1 0 1 1 0 1 0 0
```

### 輸出範例 1

```
0.0 0.0
0.5 0.5
0.5 0.5
```

### 輸入範例 2

```
5
35.45838 10.75771 20.30875 31.79885 -29.69125 -4.20115 -14.54162
-25.24229
-34.29322 -6.83750 -9.13374 -35.27865 42.86626 10.72135 17.70678
39.16250
1.76762 -11.97798 7.85087 -10.45716 -0.14913 21.54284 -6.23238
20.02202
-36.66981 -14.82302 -13.78842 -37.70441 40.21158 16.29559
17.33019 39.17698
-32.34050 -30.40612 12.37671 -42.74190 28.37671 15.25810 -
16.34050 27.59388
```

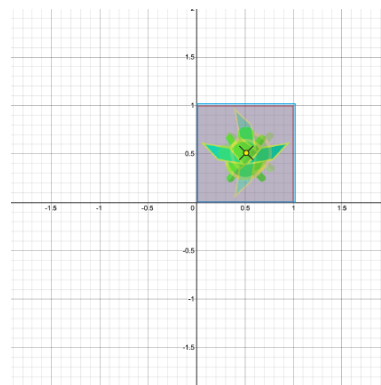
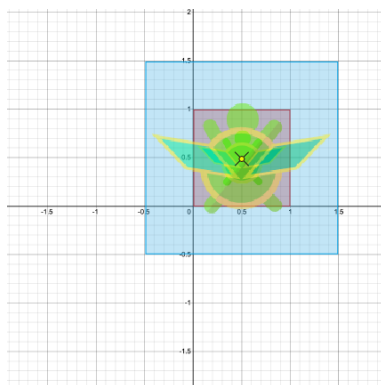
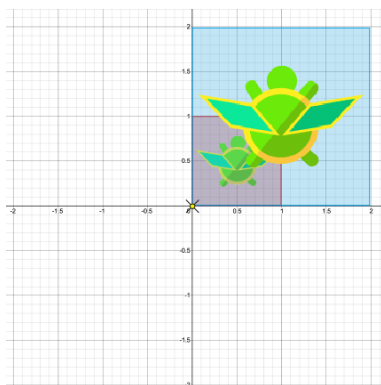
### 輸出範例 2

```
0.4692521323 0.5576028062
0.4628180343 0.4444572639
0.2417009103 0.3745290840
0.4772064810 0.4857012755
0.5045677930 0.6426378565
```

### 範例說明

提供第 1 筆範例測資的三張地圖以供參考：





此處，紅色方形代表的是地圖的範圍、藍色方形為綠蠓龜聚落範圍；為了標示方向，有放一張烏龜的圖在上面。黃色圈圈和 X 字則代表答案所在的位置。倘若紙本因為列印關係而看不清楚，請參考提供的電子版本。

## 問題 5 – 噪音防治 (Noise Prevention)

(15 分)

### 問題敘述

旅行家 Joy 喜歡到處旅行，這天，他來到了 YTP 國，一個對旅行者非常友善的國度，為了充份享受 YTP 國  $N$  個城鎮的樂趣，Joy 決定花費  $\frac{N \times (N-1)}{2}$  天把每對城鎮來回玩過一遍，也就是說，每天他會選擇一對還沒選擇過的相異城鎮對  $\{i, j\}$  並從城鎮  $i$  玩到城鎮  $j$ ，再從城鎮  $j$  玩回城鎮  $i$ 。

不過，Joy 身為旅人有一個棘手的問題，那就是他特別在乎自己的耳膜，每天晚上他總是要先調查好隔天的行程，並試圖找出隔天行程中會遇到的最大噪音值為何，然後準備好相對應的防護措施，才不會被這趟行程中的噪音搞壞耳膜導致他沒辦法看他喜歡的 Vtuber 們直播。

現在，Joy 已經蒐集好了 YTP 國  $M$  條雙向道路的噪音資訊了，已知每條道路的噪音值都是常駐的，也就是不會隨著時間改變，因此他可以事先幫每對城鎮對都找出「**最大噪音值最小的往返路線**」後，來回都沿著這條路線走，就可以最小化他旅程的傷害與花費。

但這些資料實在是太龐大了，由於噪音防護措施的花費與每天遭遇的最大噪音值總和成正比，所以 Joy 委託你幫他寫一支程式，請你幫他先找出所有城鎮對「**最大噪音值最小的往返路線**」的**最大噪音值總和**，好讓他可以先準備好旅遊資金。

### 輸入格式

首行輸入兩個以空格隔開的正整數  $N, M$ ，代表 YTP 國有  $N$  個城鎮、 $M$  條道路。  
接下來  $M$  行，第  $i$  行三個以空格隔開的正整數  $u_i, v_i, w_i$ ，代表城鎮  $u_i$  和城鎮  $v_i$  之間有一條道路，且這條道路的常駐噪音值為  $w_i$ 。

### 輸出格式

輸出一個整數，代表所有城鎮對「**最大噪音值最小的往返路線**」的**最大噪音值總和**。

### 資料範圍

- $1 \leq N \leq 3 \times 10^5$
- $N - 1 \leq M \leq 5 \times 10^5$
- $1 \leq u_i, v_i \leq N, u_i \neq v_i$
- $1 \leq w_i \leq 10^6$
- 保證任兩座 YTP 國的城鎮都可以透過一連串的道路互相往返

### 輸入範例 1

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

### 輸出範例 1

```
45
```

### 輸入範例 2

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

### 輸出範例 2

```
10
```

### 輸入範例 3

```
8 13
8 2 7
4 8 11
6 1 14
4 5 14
1 7 9
3 1 6
7 1 12
1 3 9
2 1 8
2 4 13
4 3 3
8 6 8
2 3 13
```

### 輸出範例 3

```
262
```

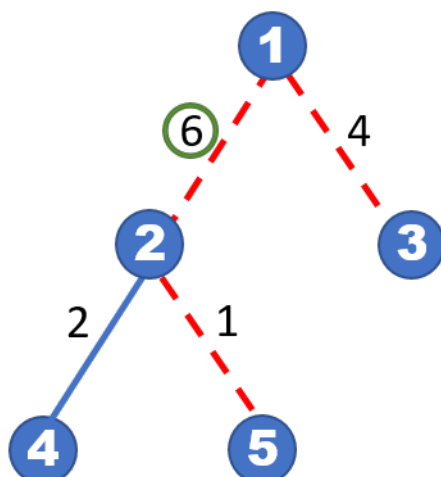
## 範例說明

在輸入範例 1，Joy 會花上  $\frac{5 \times 4}{2} = 10$  天遊玩每一對城鎮，這 10 對分別是：

{1,2}，最大噪音值 6；{1,3}，最大噪音值 4；{1,4}，最大噪音值 6；{1,5}，最大噪音值 6；{2,3}，最大噪音值 6；{2,4}，最大噪音值 2；{2,5}，最大噪音值 1；{3,4}，最大噪音值 6；{3,5}，最大噪音值 6；{4,5}，最大噪音值 2。

因此所有最大噪音值的總和為  $6 + 4 + 6 + 6 + 6 + 2 + 1 + 6 + 6 + 2 = 45$ 。

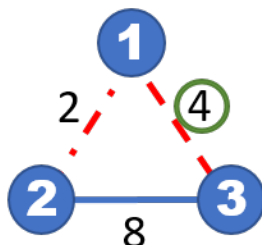
最大噪音值的計算方式以 {3,5} 這對城鎮為例，如下圖：



注意到城鎮 3 到城鎮 5 之間只有一種簡單走法，如圖中的虛線路線，而中間會經過最大噪音值的道路就是從城鎮 1 到城鎮 2、噪音值為 6 的那條，因此城鎮對 {3,5} 的最大噪音值為 6。

在輸入範例 2，Joy 會花上  $\frac{3 \times 2}{2} = 3$  天遊玩每一對城鎮，這 3 對分別是：  
 {1,2}，最大噪音值 2；{1,3}，最大噪音值 4；{2,3}，最大噪音值 4。  
 因此所有最大噪音值的總和為  $2 + 4 + 4 = 10$ 。

最大噪音值的計算方式以 {2,3} 這對城鎮為例，如下圖：



注意到城鎮 2 到城鎮 3 之間有兩種簡單走法，儘管城鎮 2 到城鎮 3 之間有一條直接的道路，但這條道路的噪音值為 8，Joy 如果選擇繞路先走到城鎮 1 再到城鎮 3 的話，就可以和圖中的虛線路線一樣，經過噪音值不超過 4 的道路，進而讓這天旅程的最大噪音值只有 4。

在輸入範例 3，Joy 會花上  $\frac{8 \times 7}{2} = 28$  天遊玩每一對城鎮，由於資料龐大，這裡不一一條列，但請注意到輸入的第 5 和第 7 條道路，這兩條道路連接的城鎮對是一樣的，城鎮的順序並不影響道路的可用性，這表示城鎮 1 到城鎮 7 有兩條直接的道路可以選擇，因此這還是一種合法的輸入，至於要選擇哪一條可以被策略所決定。

## 問題 6 – 跳吧！電熊(Hop Skip and Jump)

(10 分/12 分/20 分) 此題有部份給分

### 問題敘述

**電熊**大法師具有可以將時間跳過一小時的能力：被他跳過的這一小時的事情不會消失，但是**電熊**自己會完全感受不到中間的過程。例如**電熊**接下來的一個小時後要掃地，但是掃地的過程太無聊了，**電熊**就會發動能力讓這掃地的一小時跳過，這樣**電熊**就不會感覺到無聊了。這個能力聽起來很強，卻有一些限制要遵守：

第一、這能力只能在整點發動，例如下午 3 點發動，不能在不是整點的時候，例如 3 點 10 分發動。

第二、這能力發動次數有限，並不能發動無限次。目前**電熊**還可以發動  $k$  次能力。

第三、這能力是不能連續發動的：如果發動了跳躍時間的能力，為了要讓腦袋適應被跳過的時間，發動完後的一小時內不能再次發動的。

某天，**電熊**有了一個預知夢，他知道了接下來從整點開始算的  $N$  個小時內會發生什麼樣的事情，並將這每個小時所經歷的事情量化，做成一個開心數值；如果這個小時**電熊**在看 Vtuber，那麼這個小時量化出來的開心數值就會很高，**電熊**就會想要直接親身感受這段時間；如果這個小時**電熊**都在做一些無聊的工作，那麼這個小時量化出來的開心數值就會很低，此時**電熊**就會傾向要發動能力跳過這段時間。現在問題來了：如果**電熊**可以發動  $k$  次跳過一小時的能力，並且符合上述關於能力的限制，那麼**電熊**所經歷的開心程度總值最大為何呢？

### 輸入格式

輸入的第一行包含兩個正整數  $N, k$ ，代表**電熊**夢到了從整點算下來的  $N$  小時內會發生的事情，以及**電熊**可以發動  $k$  次跳躍一個小時的能力。

接下來的一行有  $N$  個整數  $x_i$ ，表示**電熊**接下來經歷  $N$  小時中，每個小時依序的開心數值。

### 輸出格式

請輸出**電熊**最多可以發動  $k$  次能力的條件下，**電熊**所經歷過的開心程度總值最大為何？

## 資料範圍

對於所有的輸入資料，皆滿足  $1 \leq k \leq N \leq 1000000$ ， $-1000 \leq x_i \leq 1000$ 。

此外，該題有部份給分：

- 第一部份為範例測資不計分
- 第二部份測資通過給 10 分，測試資料滿足  $1 \leq N \leq 20$ 。
- 第三部份測資通過給 12 分，測試資料滿足  $1 \leq N \leq 3000$ 。
- 第四部份測資通過給 20 分，測試資料滿足  $1 \leq N \leq 200000$ 。

### 輸入範例 1

```
3 3
-4 -2 -4
```

### 輸出範例 1

```
-2
```

### 輸入範例 2

```
3 1
-3 -5 -3
```

### 輸出範例 2

```
-6
```

### 輸入範例 3

```
3 2
-3 -5 -3
```

### 輸出範例 3

```
-5
```

## 範例說明

在範例 1 中，**電熊**可以跳過第一個小時，經歷第二個小時，再跳過第三個小時。開心總值為 -2，僅有這種走法可以達到最佳解。

在範例 2 中，**電熊**可以經歷第一個小時，跳過第二個小時，經歷第三個小時。開心總值為  $(-3) + (-3) = (-6)$ ，僅有這種走法可以達到最佳解。

在範例 3 中，**電熊**可以跳過第一個小時，經歷第二個小時，再跳過第三個小時。開心總值為 -5，僅有這種走法可以達到最佳解。

## 問題 7 – 壘球大師 (Softball)

(20 分)

### 問題敘述

東東很喜歡打壘球。今年，他所在的「YTP 壘球隊」會參加  $N$  場比賽，每場比賽會從第  $l_i$  天開始，打到第  $r_i$  天結束，並且會帶給東東  $w_i$  的快樂度。作為一個有責任感的球員，東東如果決定參加每場比賽就會從第一天參加到最後一天，因此他無法參加兩個日期有重疊的比賽。因為「YTP 壘球隊」的規模十分龐大，東東有非常多的比賽可以選擇，他想要知道，如果做出了最好的選擇，他最多能獲得多少的快樂度呢？

### 輸入格式

輸入的第一行包含一個正整數  $N$ ，表示「YTP 壘球隊」今年一共會參加幾場比賽。

接下來的  $N$  行每行包括 3 個正整數  $l_i, r_i, w_i$ ，代表每場比賽的開始日期、結束日期以及能帶給東東的快樂度。

注意到這題的輸入量十分龐大，使用 C++ 作答的同學請在程式碼開頭加上 `#include<cstdio>`，並使用 `scanf` 和 `printf` 作答。

### 輸出格式

輸出一個非負的整數代表東東最高能獲得的快樂度。

### 資料範圍

- $1 \leq N \leq 10^6$
- $1 \leq l_i \leq r_i \leq 10^9$
- $1 \leq w_i \leq 10^9$

### 輸入範例 1

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

### 輸出範例 1

```
6
```

### 輸入範例 2

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

### 輸出範例 2

7

### 輸入範例 3

```
3
1 3 5
3 5 7
5 8 11
```

### 輸出範例 3

```
16
```

### 範例說明

在範例 1，東東的最佳策略是三場比賽都參加，總共的快樂度是  $1 + 2 + 3 = 6$ 。

在範例 2，東東的最佳策略是參加第二場比賽，總共的快樂度是 7。

在範例 3，東東的最佳策略是參加第一場和第三場比賽，總共的快樂度是  $5 + 11 = 16$ 。

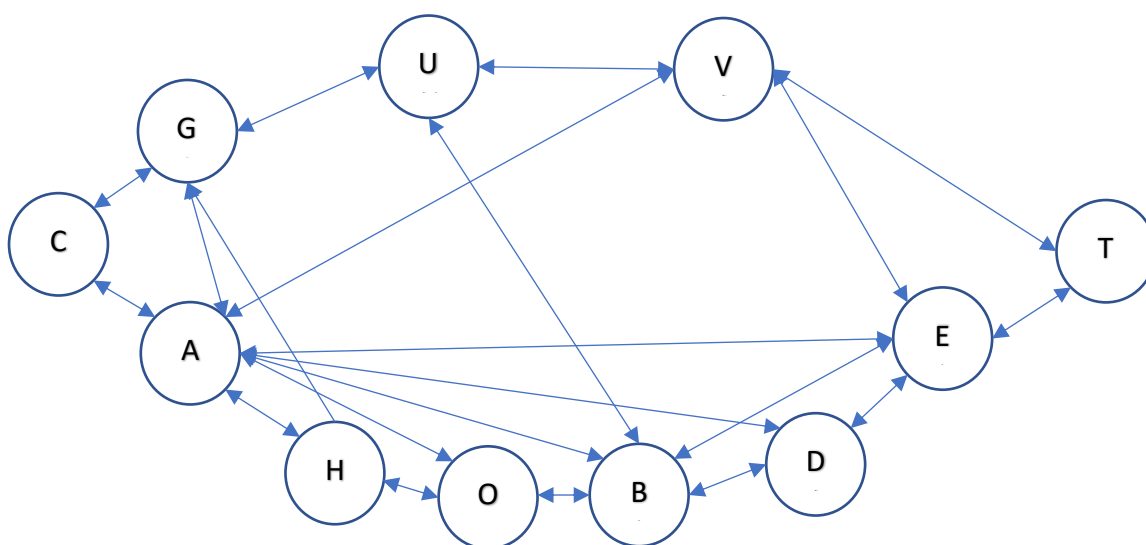


## 問題 8 – 最安全回家的路 (SafeWayHome)

(20 分)

### 問題敘述

今年發生人為疏失的重大交通意外，造成許多人失去了生命，因為可以選擇的交通方式有限，所以需要大家在求快、求便宜和求安全下，依照心中的想法尋求最小的數字（時間、成本、意外數），找到平安回家的方法。



### 輸入格式

第一行兩個字母，以逗號分開，是開始地點的字母和結束地點的字母，接下來有數行，每行三個值以逗號分開，表示路徑的開始點的英文字母、結束點的英文字母，以及一個數值（如：成本），而最後一行“-,-,0”表示結束。(註: 沒有兩條路徑有共同的起點和結束點)

### 輸出格式

第一行為最小的最終累加值。

第二行開始為每一個符合最小值的從開始地點到結束地點的路徑。

依照經過的節點少優先排列，如果經過節點相同，則以字母順序排列。

如果沒有方法到達, 輸出 “NOTFOUND”

### 資料範圍

字母 A-Z，數字 0-999 (0 僅用為結束使用), 一張地圖所有地點不超過 15 個。

符合最小值的路徑不超過 100,000 條。

### 輸入範例 1

A, V  
A, C, 3  
C, G, 8  
G, U, 10  
U, V, 10  
A, G, 7  
A, V, 27  
-, -, 0

### 輸出範例 1

27  
AV  
AGUV

### 範例 1 說明

AV 和 AGUV 都同樣是合計最小值為 27。

### 輸入範例 2

V, A  
C, A, 3  
A, C, 5  
C, G, 8  
G, C, 5  
G, U, 10  
U, G, 8  
V, U, 10  
A, G, 10  
G, A, 8  
A, V, 10  
V, A, 26  
-, -, 0

### 輸出範例 2

26  
VA  
VUGA  
VUGCA

### 輸入範例 3

A, X  
A, C, 3  
C, G, 8  
G, A, 11  
Y, X, 10  
-, -, 0

### 輸出範例 3

NOTFOUND

## 問題 9 – 網頁超圓結 (URLoop)

(20 分)

### 問題敘述

小莫正在研究網頁爬蟲！

網頁爬蟲，是讓電腦自動瀏覽網站，讓程式自動點選網頁中的每一個超連結，連到不同網頁，對這些不同網頁又可以點它們各自的超連結，連到更多網頁.....以此類推  
如果過程中把每個網頁都存起來，甚至能把整個網站複製下來呢 OwO

正當小莫開心地看著自己寫的爬蟲程式一步一步悠哉地瀏覽網頁，突然發現好像哪裡怪怪的——這個網站怎麼爬都爬不完 OAO

經過仔細追查後，發現原來有 90%都是同一個網頁，都是自己連到自己！只是網址中的某個小片段隨著點擊超連結會重複越來越多次，網址越長越長，永遠沒有結束的一天！



如圖，三個網址中，無論%2F3%2Fpage 重複多少次，都是同一個網頁

[https://www.systexsoftware.com.tw/category/news/?post\\_type=anypage%2F168%2Fpage%2F3%2Fpage%2F3%2Fpage%2F168%2Fpage/2/](https://www.systexsoftware.com.tw/category/news/?post_type=anypage%2F168%2Fpage%2F3%2Fpage%2F3%2Fpage%2F168%2Fpage/2/)  
[https://www.systexsoftware.com.tw/category/news/?post\\_type=anypage%2F168%2Fpage%2F3%2Fpage%2F3%2Fpage%2F3%2Fpage%2F168%2Fpage/2/](https://www.systexsoftware.com.tw/category/news/?post_type=anypage%2F168%2Fpage%2F3%2Fpage%2F3%2Fpage%2F3%2Fpage%2F168%2Fpage/2/)  
[https://www.systexsoftware.com.tw/category/news/?post\\_type=anypage%2F168%2Fpage%2F3%2Fpage%2F3%2Fpage%2F3%2Fpage%2F3%2Fpage%2F168%2Fpage/2/](https://www.systexsoftware.com.tw/category/news/?post_type=anypage%2F168%2Fpage%2F3%2Fpage%2F3%2Fpage%2F3%2Fpage%2F3%2Fpage%2F168%2Fpage/2/)

這可不行><

小莫希望讓他的爬蟲程式在網址中能找出最長的重複片段，才能終結無限輪迴，你可以幫他嗎？

## 輸入格式

輸入只有兩行

第一行是個有可能很長很長的字串  $s$ ，代表小莫的爬蟲程式目前找到的網址

第二行有個正整數  $c$ ，代表你找到的重複片段至少要連續重複出現  $c$  次以上

## 輸出格式

請輸出一行三個數字  $i, L, n$  以空白間隔，代表你找到最長的重複片段的資訊

$i$  代表這個重複片段在字串中的位置，例如字串開頭的位置時  $i=0$

$L$  代表重複片段的長度

$n$  代表連續重複幾次， $n$  應該要整除  $L$

既然至少要連續重複出現  $c$  次以上，代表你找出的  $n$  應該要  $\geq c$

重複片段要最長，所以  $L$  應該儘量大

如果有多解，輸出重複次數最多的（ $n$  最大）

如果還是有多解，輸出位置最接近字串開頭的（ $i$  最小）

如果無解，請輸出一行三個數字  $-1 -1 -1$ （以空白間隔）

## 資料範圍

$2 \leq C \leq \text{字串 } s \text{ 的長度} \leq 200000$

$s$  中出現的字元僅限於 `ascii` 編碼 33 到 126 的字元，其中包含英文大小寫字母（`a-z`、`A-Z`）、數字（`0-9`）、各種括號、特殊符號等等，不包含空白

以下程式碼可印出這些字元：

```
for(int i=33;i<=126;i++)printf("%c",(char)i);
```

輸出：

```
!"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNopQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz{|}~
```

## 資料範例

### 輸入範例 1

aaaaaaaaaa

3

### 輸出範例 1

0 10 10

### 輸入範例 2

baaaaabaaaab

3

## 輸出範例 2

1 5 5

## 輸入範例 3

baaaaabaaaaab  
3

## 輸出範例 3

7 6 6

## 輸入範例 4

baaaaabaaaaab  
2

## 輸出範例 4

0 12 2

## 輸入範例 5

caaaaabaaaaab  
2

## 輸出範例 5

1 12 2

## 輸入範例 6

caaaaabaaaaab  
6

## 輸出範例 6

-1 -1 -1

## 範例解釋

範例 1、2、3 的重複單元都是 a

範例 4 的重複單元是 baaaaa

範例 5 的重複單元是 aaaaab

範例 6 找不到任何重複 6 次以上的重複片段

## 問題 10 – 旅行 (Trip)

(20 分)

### 問題敘述

今年暑假，你搭乘私人飛機來到一個名為 YTP 的國家旅遊。

YTP 國共有  $N$  個城市，第  $i$  座城市的座標為  $(X_i, Y_i)$ 。

任兩座城市的座標皆相異。

一開始，你可以任意選一座城市降落。

如果你正在城市  $i$ ，對於任意的  $j$ ，你可以花費  $|X_i - X_j| + |Y_i - Y_j|$  元來搭火車到城市  $j$ 。

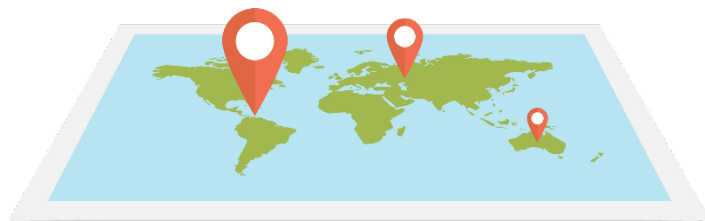
如果你和你的私人飛機都在城市  $i$ ，你可以花費  $C$  元來開飛機到任意一座其他的城市。

所以，如果你想要使用飛機來移動，但是你和你的飛機不在同個城市的話，就必須先靠火車回到飛機所在的城市。

在你的暑假計畫中，你想要拜訪過每座城市至少一次，並且在最後回到私人飛機所在的城市後離開 YTP 國。

到達和飛離 YTP 國都不需要花費任何費用。

請問你至少要花費多少錢才能完成你的暑假計畫呢？



### 輸入格式

第一行包含兩個整數  $N$   $C$ ，代表 YTP 城市有  $N$  個城市、你可以花  $C$  元開飛機從一座城市到另外一座城市。

接下來  $N$  行中，每行包含兩個整數  $X_i$   $Y_i$ ，代表第  $i$  座城市的座標。

### 輸出格式

請輸出一個整數，代表要完成你的暑假計畫所需要的最少花費。

### 資料範圍

- $1 \leq N \leq 18$
- $0 \leq C \leq 999$
- $0 \leq X_i, Y_i \leq 100$



時間限制：2 秒  
空間限制：256MB

### 輸入範例 1

```
3 1
10 10
20 20
10 0
```

### 輸出範例 1

```
2
```

### 輸入範例 2

```
3 10
0 0
1 1
1 0
```

### 輸出範例 2

```
4
```

### 輸入範例 3

```
5 35
2 2
0 6
2 0
4 4
100 100
```

### 輸出範例 3

```
55
```

## 範例說明

範例 1：最佳方案是降落在第一座城市，接著開飛機到第二座，最後開飛機到第三座。  
總費用為  $1 + 1 = 2$  元。

範例 2：最佳方案是降落在第一座城市，接下來沿著“1 2 3 1”的順序搭火車拜訪城市。並在回到第一座城市後，開飛機離開 YTP。  
總費用為  $(|0 - 1| + |0 - 1|) + (|1 - 1| + |1 - 0|) + (|1 - 0| + |0 - 0|) = 4$  元。

範例 3：最佳的方案如下，首先降落到第二座城市，接著沿著“2 1 3 4 2”的順序搭火車拜訪城市，回到第二座城市後，開私人飛機到第五座城市。最後，因為你已經拜訪過所有城市了，因此離開 YTP 國。

總費用為 $(|0 - 2| + |6 - 2|) + (|2 - 2| + |2 - 0|) + (|2 - 4| + |0 - 4|) + (|4 - 0| + |4 - 6|) + 35 = 55$ 元。

# 問題 11 – 任務終結者 (Missions Completely Completed)

---

(25 分)

## 前言

小莫買新遊戲了！

趕快打開來玩一下。

「霞道列車首」敘述一個經營「霞道」這條路線的鐵路公司，在這條途經各種湖光山色的鐵道中，你是一位列車長，運送各種不同資源到需要的地方是你的任務之一。

故事中有許多任務，許多存在著依賴關係，例如你需要先運送工人和斧頭到森林中，之後才有工人砍出來的木頭能從森林運回木製家具工廠，工廠有了木頭之後才能開始製造木製家具讓你運送去各大城市販售等等，你沒辦法一開始就幫工廠運輸家具；此外也有一些任務是排斥關係，例如運送工地建材將農莊改建成大樓之後，你就從此無法再幫忙農莊運送牛奶產品，這個運送牛奶的任務將從此消失，如果你想要運送牛奶，那就要趁農莊還在的時候趕快做；又例如你可以選擇在一個車站附近蓋工廠或發展觀光產業，但沒辦法兩者都做等等，這時我們會說兩個任務互相排斥。

故事一開始你有一些任務可以做，過程中每當完成一個任務，就有可能解鎖其他任務，或者導致別的任務消失。然而，已經完成過的任務，仍然可以再玩一次！（如果沒有因為完成其它任務而消失的話）

## 問題敘述

身為完美主義者，小莫當然不希望有任務還沒完成過就消失！小莫想要把每一個任務都玩過一遍！

但這件事有可能嗎？

現在給你每個任務間的依賴關係和排斥關係，請你寫個程式幫小莫解決疑惑吧！

## 輸入格式

第一行有個正整數  $N$ ，代表任務的數量，你的目標是把這  $N$  個任務全部玩過一遍，這些任務分別編號  $1$  到  $N$

接下來有  $N$  行，依序是任務  $1$  到任務  $N$  的解鎖條件

每行首先是一個整數  $n$ ，代表解鎖這個任務之前要先完成哪些任務，也就是同一行接下來的  $n$  個整數  $i_1, i_2, \dots, i_n$

如果  $n$  是  $0$ ，代表這個任務從一開始就解鎖了

接著一行只有一個整數  $M$ ，代表任務間的排斥關係的數量

接著  $M$  行，每一行由兩個整數  $a$  和  $b$ ，代表一個排斥關係 - 完成任務  $a$  會導致任務  $b$  消失（但不代表完成任務  $b$  會導致任務  $a$  消失）

## 輸出格式

$N$  個介於  $1$  到  $N$  的不重複整數  $j_1, j_2, \dots, j_N$ ，代表小莫應該依序進行這  $N$  個任務，如果有多解，以編號最小的任務先做為優先

如果不可能每個任務都剛好玩過一遍，請輸出一行 `no`

## 資料範圍

$$1 \leq N \leq 1000000$$

$$0 \leq n$$

$$\text{所有 } n \text{ 的總和} \leq 1000000$$

$$0 \leq M \leq 1000000$$

$$1 \leq i_1 < i_2 < \dots < i_n \leq N, \text{ 任務的解鎖條件不會包含自己}$$

$$1 \leq a, b \leq N$$

每個數字間以空白區隔

## 資料範例

### 輸入範例 1

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

### 輸出範例 1

```
2 3 1
```

### 輸入範例 2

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

### 輸出範例 2

```
2 1
```

### 輸入範例 3

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

### 輸出範例 3

```
no
```

### 範例說明:

範例 3：無論先做哪個任務，都會導致另一個任務消失，所以把兩個任務都玩過是不可能的（不重開遊戲的前提下）

## 問題 12 – 孿生印記 (Twin Mark)

(11 分/25 分) 此題有部分給分

### 問題敘述

你得到了一份神秘的古文，裡面提到了“孿生印記分數”的計算方式，並給出了兩個字串  $A$ 、 $B$ 。

給定一個字串  $s$ ，孿生印記分數為字串  $ss$  在字串  $B$  裡面出現的次數，譬如  $s = \text{"abc"}$ ， $B = \text{"aaabcbcabcc"}$  來說， $ss = \text{"abcabc"}$  一共在  $B$  裡面出現了兩次，因此字串  $s$  的孿生印記分數就是 2 分。

現在你很好奇，對於字串  $A$  每一個前綴的孿生印記分數加總是多少呢？一個字串的前綴是所有從頭開始的子字串，舉例來說字串  $\text{"abcdef"}$  的前綴一共有 6 個，分別是  $\text{"a"}$ ， $\text{"ab"}$ ， $\text{"abc"}$ ， $\text{"abcd"}$ ， $\text{"abcde"}$ ， $\text{"abcdef"}$ 。

### 輸入格式

輸入一共有兩行，第一行為字串  $A$ ，第二行為字串  $B$ 。

- $|A|, |B| \leq 2 \times 10^5$
- 字串  $A$ 、 $B$  都只包含英文小寫字母。

你如果答對所有滿足  $|A|, |B| \leq 5000$  的測試資料，你將可以獲得其中的 **11** 分。

### 輸出格式

請一共輸出一行包含一個數字，代表每一個前綴的孿生印記分數加總。

### 輸入範例 1

```
abc  
aaababc
```

### 輸出範例 1

```
2
```

## 輸入範例 2

aaa  
aaa

## 輸出範例 2

2

## 輸入範例 3

abc  
abc

## 輸出範例 3

0

## 範例說明

在輸入範例 1 中：

"aa" 在 "aababc" 裡面出現了 1 次

"abab" 在 "aababc" 裡面出現了 1 次

"abcabc" 在 "aababc" 裡面出現了 0 次

因此答案為  $1 + 1 + 0 = 2$ 。

在輸入範例 2 中：

"aa" 在 "aaa" 裡面出現了 2 次

"aaaa" 在 "aaa" 裡面出現了 0 次

"aaaaaa" 在 "aaa" 裡面出現了 0 次

因此答案為  $2 + 0 + 0 = 2$ 。

在輸入範例 3 中：

"aa" 在 "abc" 裡面出現了 0 次

"abab" 在 "abc" 裡面出現了 0 次

"abcabc" 在 "abc" 裡面出現了 0 次

因此答案為  $0 + 0 + 0 = 0$ 。

## 問題 13 – 往花那家的路上 (Crossick)

(12 分/25 分) 此題有部份給分

### 問題敘述

這天是巴期待已久的日子，她終於能去公司的心儀前輩花那的家裡玩了。一大早坐著新幹線來到在京都的巴卻在錯綜複雜的京都迷路了。好在京都有許多計程車可以載巴抵達花那家裡。

京都總共有  $N$  個路口、 $M$  條雙向道路、 $K$  台計程車。為了讓繁忙的京都不要天天塞車，京都的路都有流量的限制：第  $i$  條路只能讓編號滿足  $l_i \leq x \leq r_i$  的計程車通過。已知花那住在編號為  $1$  的路口，請幫幫巴，告訴她有多少台計程車能夠成功載她到花那家，讓她能快點和朝思暮想的花那見面吧！

喔對了，巴真的迷路了，所以她自己也不知道她在哪一個路口，所以請你告訴巴每一個路口的答案。

### 輸入格式

輸入第一行有三個以空白隔開的整數  $N, M, K$ 。

接著共有  $M$  行，第  $i$  行有四個以空白隔開的整數  $x_i, y_i, l_i, r_i$ ，告訴你第  $i$  條路連接  $x_i, y_i$  兩個路口，而這條路只能通過編號位於  $[l_i, r_i]$  的計程車。

### 輸出格式

請輸出  $N - 1$  行，第  $i$  行有一個整數，表示如果巴現在在第  $i + 1$  個路口，則有多少台計程車可以載她到花那家。

### 資料範圍

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

$$0 \leq M \leq 10^5$$

$$1 \leq x_i, y_i \leq N$$

$$1 \leq l_i \leq r_i \leq K \leq 10^9$$

### 子任務

第一部份為範例測試資料，不計分

第二部份 (12 分) 測試資料滿足  $N, M, K \leq 5000$

第三部份 (25 分) 測試資料滿足 無特殊限制

### 輸入範例 1

```
2 2 10
1 2 1 2
1 2 8 9
```



### 輸出範例 1

4

### 輸入範例 2

```
3 5 1000000000
1 2 1000000001 300000000
2 3 2000000001 800000000
1 3 4000000001 600000000
2 1 7000000001 1000000000
3 3 123456789 987654321
```

### 輸出範例 2

```
7000000000 4000000000
```

### 輸入範例 3

```
6 10 30
6 5 22 29
5 6 14 23
6 2 21 27
6 1 5 12
5 1 21 28
2 4 3 9
6 5 26 30
4 5 1 30
6 1 3 12
4 1 4 15
```

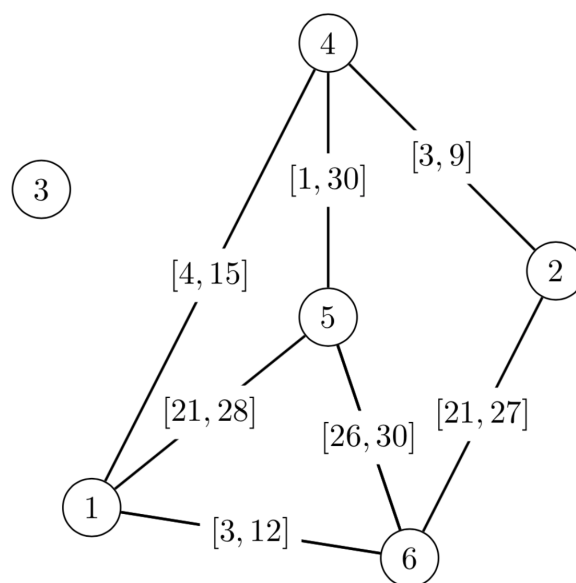
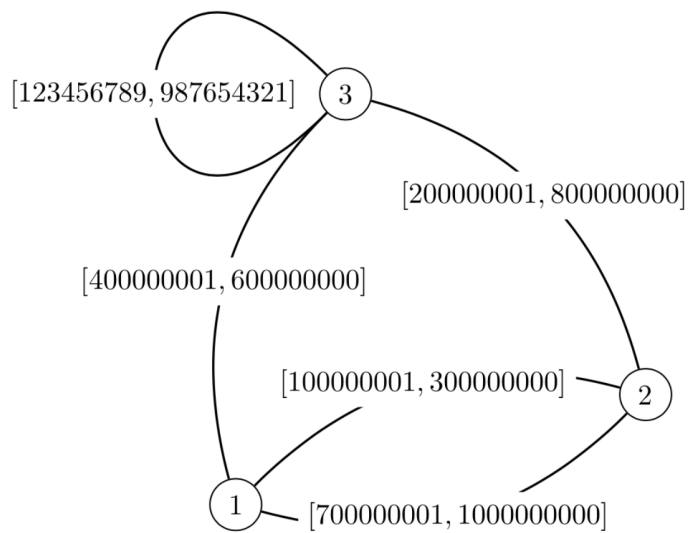
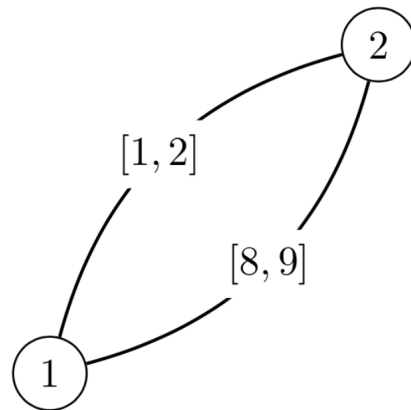
### 輸出範例 3

```
13 0 20 20 20
```

## 範例說明

在範例 1 中，僅有第 1,2,8,9 號的計程車能抵達花那的家，輸出為 4。

以下是 3 筆範例輸入的圖示：



## 問題 14 – 赤壁之戰 (Chibi)

(6 分/16 分/25 分) 此題有部份給分

### 問題敘述

曹操看到自己的船被火燒個精光之後，迅速下達撤退的指令。

他們撤退到了烏林之後，他們發現烏林是一個由很多樹組成的森林，看到樹之後，曹操不禁想到了一個困惑他許久的問題。

曹操把當時的地圖局勢畫成一個  $N$  個點的有根樹，樹的根節點是 1，點以 1 到  $N$  編號。每個點都代表一個陣地，而對於第  $i$  個陣地，他的兵力值為  $w_i$ ，代表有  $w_i$  個士兵正在防守那個陣地。

現在定義  $(i, j)$  是一個“好的二元組”，代表底下的條件全部都成立：

1. 點  $i$  是點  $j$  的祖先節點。
2.  $w_i > w_j$

曹操很好奇，在目前的地圖中，總共有多少個好的二元組。

除此之外，曹操還會做出  $Q$  個操作，第  $i$  個操作是把節點  $pos_i$  的兵力值改成  $val_i$ 。曹操同時也很好奇，對於每個介於  $[1, Q]$  之間的  $i$ ，在他依序做完第 1 個操作到第  $i$  個操作時，地圖上好的二元組的個數。

### 輸入格式

輸入的第一行包含兩個整數  $N, Q$ ，代表樹的節點數量，以及曹操操作的數量。

接下來的一行，包含  $N - 1$  個正整數  $p_2, p_3, \dots, p_N$ ， $p_i$  代表節點  $i$  的父節點是  $p_i$ 。

接下來的一行，包含  $N$  個正整數  $w_1, w_2, \dots, w_N$ ， $w_i$  代表第  $i$  個節點一開始的兵力值。

接下來的  $Q$  行，第  $i$  行包含兩個正整數  $pos_i, val_i$ ，代表在第  $i$  次操作中，曹操要把點  $pos_i$  的兵力值改成  $val_i$ 。

### 輸出格式

輸出  $Q + 1$  個數字，每個數字一行。

第一行輸出的數字代表在還沒有經過任何操作前，有多少個好的二元組。

接下來的  $Q$  行，第  $i$  行代表曹操依序做了操作 1 到操作  $i$  後，地圖上的好的二元組的數量。

## 資料範圍

- $2 \leq N \leq 100000$
- $0 \leq Q \leq 100000$
- $1 \leq p_i \leq i - 1$
- $1 \leq pos_i \leq N$
- $1 \leq w_i, val_i \leq N + Q$
- 任兩個  $w_i$  皆不相等，任兩個  $val_i$  皆不相等，任一  $w_i$  與任一  $val_i$  皆不相等

### 輸入範例 1

```
4 0
1 2 3
4 2 3 1
```

### 輸出範例 1

```
5
```

### 輸入範例 2

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

### 輸出範例 2

```
5
2
3
```

### 輸入範例 3

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

### 輸出範例 3

```
4
3
4
3
```

## 子任務

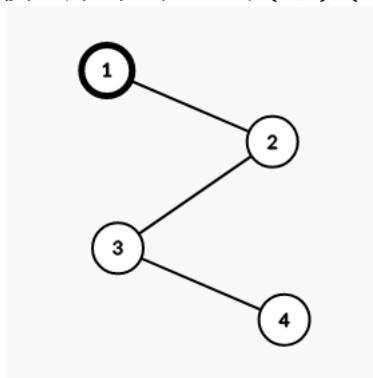
- 子任務一(不計分)，對應範例測試資料
- 子任務二(6 分)，對應的測試資料，滿足  $Q = 0, p_i = i - 1$
- 子任務三(16 分)，對應的測試資料，滿足  $p_i = i - 1$
- 子任務四(25 分)，對應的測試資料，沒有任何特殊的限制

## 範例說明

範例 1、2、3 的資料範圍分別對應了子任務一、二、三。

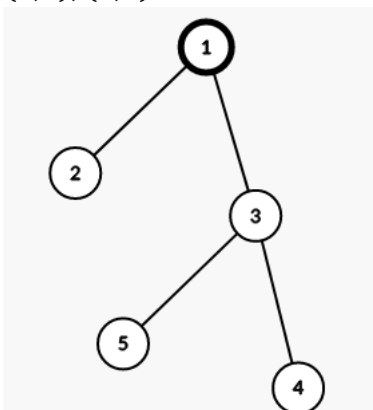
在範例 1 中，好的二元組為  $(1,2), (1,3), (1,4), (2,4), (3,4)$ 。

在範例 2 中，一開始好的二元組跟範例一一樣。在經過第一個操作後，好的二元組為  $(1,2), (1,3)$ 。在經過前兩個操作後，好的二元組為  $(1,2), (1,3), (2,3)$ 。



範例 1，範例 2 對應之圖示（圈圈裡面的數字是節點的編號）

在範例 3 中，一開始好的二元組為  $(1,2), (1,3), (1,4), (3,4)$ 。經過一個操作後，好的二元組為  $(1,2), (1,3), (1,4)$ 。經過前二個操作後，好的二元組為  $(1,2), (1,4), (3,4), (3,5)$ 。經過前三個操作後，好的二元組為  $(1,2), (3,5), (3,5)$ 。



範例 3 對應的圖例（圈圈裡面的數字是節點的編號）

## 問題 15 – Brainf\*ck

(25 分)

### 問題敘述

Brainf\*ck 是一個極簡程式語言，他用一個初始值為 0 的陣列來模擬記憶體，陣列每個元素都是一個 0~255 的整數，如同常見的程式語言，255 加上 1 會變成 0，而 0 減掉 1 會變成 255，另外還有一個指標一開始指著陣列的第一個元素。

Brainf\*ck 只有八種字元，程式執行的時候，會先執行程式碼第一個字元代表的指令，接著執行第二個字元代表的指令，一直到整個程式碼被執行完，但若遇到代表迴圈的 [ ] 可能會重複執行指令。各個字元代表的意義如下：

<	將指標往左移一格
>	將指標往右移一格
+	將指標指向的元素加上 1
-	將指標指向的元素減掉 1
[	若當前指標指向的元素為 0，則跳到對應的右括號繼續執行程式
]	跳至對應的左括號繼續執行程式

實際上還有代表輸入輸出的 “.” 和 “,”，但本題並不會使用。若把 brainf\*ck 的六個字元轉換成 C 語言可以對應到下面的表格（取自維基百科）

<	++ ptr;
>	--ptr;
+	++(*ptr);
-	--(*ptr);
[	while(*ptr){
]	}

其中，ptr 是一個指向陣列某個元素的指標。例如 [+] 會把當成指標指向的元素不斷加一，直到 overflow 成 0，相當於 C 語言中的

```
while(*ptr){
    ++(*ptr);
}
```

而 `[->+<]` 則相當於

```
while(*ptr){
    --(*ptr);
    ++(* (ptr + 1));
}
```

注意到最後的 `<` 是為了確保每次迴圈開始的時候指標是在 `ptr` 而不是 `ptr + 1`。

本題規定記憶體大小只有 1024，也就是 `brainf*ck` 會在大小 1024 的陣列上執行，若指標超出陣列範圍則會產生 `error`。給定正整數  $n$  和長度為  $n$  的陣列  $a_1, a_2, \dots, a_n$ ，請設計一個長度不超過 18000 的 `brainf*ck` 程式使得程式結束時記憶體的前  $n$  個數字恰為  $a_1, a_2, \dots, a_n$  且記憶體其他位置皆為 0。另外為了避免程式執行過久，這個程式最多只能執行  $10^7$  個指令（字元）。

## 輸入格式

輸入共兩行。第一行有一個正整數  $n$ ，第二行有  $n$  個以空白分隔的整數  $a_1, a_2, \dots, a_n$ 。

## 輸出格式

輸出一個字串代表一個 `brainf*ck` 程式碼，這個字串只能有 `+-><[]` 這六種字元，若出現非法字元（包含最後的換行字元）則會判定為 `Wrong Answer`。合法字元的數量最多只能有 18000 個。所有產生正確記憶體數值的程式碼皆可得到 `Accepted`。

## 資料範圍

$$1 \leq n \leq 1000$$

$$0 \leq a_i \leq 255$$

### 輸入範例 1

```
2
0 16
```

### 輸出範例 1

```
++++[->++++<]
```

### 輸入範例 2

```
1
255
```

### 輸出範例 2

```
-
```

## 範例說明

範例 1 中的 brainf\*ck 程式碼 ++++[->++++<] 相當於以下 c 語言程式碼，所以 brainf\*ck 程式執行結果，陣列第一格會是 0，第二格會是 16

```
// initialize
int memory[1024] = {0};
int ptr = 0;

memory[ptr] += 4;          //++++
while(memory[ptr]){        //[
    --memory[ptr];         //-
    ++ptr;                 //>
    memort[ptr] += 4;       //++++
    --ptr;                 //<
}                           //]
```

範例 2 則是把陣列第一格減一，因為初始值為 0，所以結果會是 255。



# 問題 16 – ひなの與カミト的婚禮 (Oreapo)

(25 分)

## 問題敘述

ひなの與カミト終於要結婚了。在規劃婚禮的途中，他們卻開始猶豫要如何分配會場的座位。他們的婚禮預計將使用至少  $L$  個，至多  $R$  個宴會廳。由於一個桌子能容納  $K$  人，婚禮公司要求所有宴會廳的位子個數的最大公因數必須恰好是  $K$ 。由於他們預計寄出  $N$  封喜帖，但不一定每個人都有空參加，請對每個  $1 \leq n \leq N$  回答有多少將  $n$  個座位分配至若干個宴會廳的方式可以滿足以上需求。

一種分配方法指的是一個正整數  $k$ ，表示使用的宴會廳個數，與一個正整數數列  $a_1, \dots, a_k$ ，表示每個宴會廳的位子個數。

兩種分配方法  $(k, a)$  與  $(l, b)$  視為同一種，若且唯若  $k = l$  且可經由重排將  $a$  變成  $b$ 。

## 輸入格式

輸入共有一行，包含四個整數  $N, L, R, K$ 。

預計寄出  $N$  封喜帖，

宴會廳，至少  $L$  個，至多  $R$  個，

所有宴會廳的位子個數的最大公因數必須恰好是  $K$ 。

## 輸出格式

請輸出一行，包含  $N$  個以空白隔開的整數，第  $i$  個整數代表若有  $i$  個賓客會參加，則有多少種分配座位的方法可以滿足以上需求。

因為答案可能很大，請輸出答案除以 998244353 的餘數。

## 資料範圍

- $1 \leq N \leq 10^5$
- $1 \leq L \leq R \leq N$
- $1 \leq K \leq N$

## 輸入範例 1

6 2 5 1

### 輸出範例 1

0 1 2 3 6 6

### 輸入範例 2

9 3 6 2

### 輸出範例 2

0 0 0 0 0 1 0 2 0

### 輸入範例 3

20 1 20 1

### 輸出範例 3

1 1 2 3 6 7 14 17 27 34 55 63 100 119 167 209 296 347 489 582

## 範例說明

範例 1 中，依輸入條件  $1 \leq n \leq 6$ ，考慮宴會廳有 2~5 間，且最大公因數 1。可能的座位的分配方法枚舉如下：

$n=1$  個賓客, 不符宴客廳 2~5 的要求，分配座位的方法 = 0。

$n=2$  個賓客, 宴客廳 2 間且人數(1,1)，分配座位的方法 = 1。

$n=3$  個賓客, 宴客廳 2 間且人數(2,1)；宴客廳 3 間且人數(1,1,1)，分配座位的方法 = 2。

$n=4$  個賓客, 宴客廳 2 間且人數(3,1)；宴客廳 3 間且人數(2,1,1)；宴客廳 4 間且人數(1,1,1,1)。分配座位的方法 = 3。

$n=5$  個賓客, 宴客廳 2 間且人數(2,3)或(4,1)兩種；宴客廳 3 間且人數(2,2,1)或(3,1,1)兩種；宴客廳 4 間且人數(2,1,1,1)；宴客廳 5 間且人數(1,1,1,1,1)，分配座位的方法總數為 6。

$n=6$  個賓客,

若使用 2 間宴會廳，且人數(5,1) 一種。

若使用 3 間宴會廳，且人數(4,1,1) 或 (3,2,1) 兩種。

若使用 4 間宴會廳，且人數(3,1,1,1) 或 (2,2,1,1) 兩種。

若使用 5 間宴會廳，且人數(2,1,1,1,1) 一種。

分配座位的方法 =  $1+2+2+1 = 6$  種。

因此，輸出數列為 (第  $i$  個整數有  $i$  個賓客會參加的分配方法)：

0 1 2 3 6 6