



## 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!
```

## 範例說明

---

輸入範例1，無輸入值，簡單而快樂的輸出Hello World!

# 1\_ $n$ 維空間的移動方法數 (Ways of travel in $n$ dimensions)

(5分)

## 題目敘述

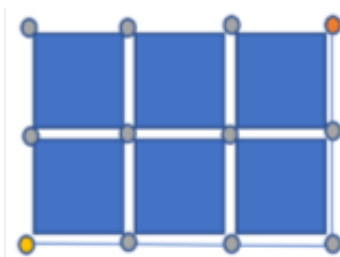
當由一個二維平面的原點  $(0, 0)$  移動到  $(x, y)$  時，每次移動一步，可以向右（ $x$  軸方向）或向上（ $y$  軸方向）正向移動直到到達  $(x, y)$  為止，可以知道由原點到  $(x, y)$  有  $\binom{x+y}{x}$  種移動方法。

組合公式：

$$\binom{n}{m} = \frac{n!}{m! \times (n-m)!}$$

例如：有 10 種方法到達  $(3, 2)$ ，即

$$\binom{3+2}{3} = \binom{5}{3} = \frac{5!}{3! \times 2!} = \frac{120}{6 \times 2} = 10$$



需要協助延伸這個公式，來計算出移動到  $n$  維空間的方法，例如： $(0, 0, 0, 0, \dots)$  到  $(a_1, a_2, a_3, a_4, \dots, a_n)$ 。

## 輸入格式

$a_1, a_2, a_3, a_4, \dots, a_n$  整數序列以逗號分開。

## 輸出格式

一個正整數  $N$  表示總共的移動方法

## 資料範圍

- 維度  $n$  滿足  $1 \leq n < 10$ 。
- 輸入值  $a_i$  滿足  $0 \leq a_i < 10$ 。
- 題目保證輸出結果  $N$  滿足  $0 < N < 2^{31} - 1$ 。

## 測試範例

## 輸入範例1

3,2

## 輸出範例1

10

## 輸入範例2

2,3,1

## 輸出範例2

60

## 輸入範例3

1,0,2,3,3,0,2,1

## 輸出範例3

3326400

## 範例說明

Case 1. 如題目敘述之計算說明。

Case 2. 如下圖所示， $(0,0,0)$  到  $(2,3,1)$  在 3 維空間中會有 60 種方法。



## 2\_涼子電腦 (Cool Computing)

(10分)

### 題目敘述

由  $100^{200} < 99^{202} < \dots n^{\lfloor 20000/n \rfloor} \dots < 4^{5000} < 3^{6666} > 2^{10000}$ , 可以知道三位元是最少狀態來表達最多的數字, 同時利用類比訊號的正電流 (+1)、負電流 (-1) 和近0電流(0), 也可以更有效率的 (更少的電子元件) 來儲存、運算或是傳輸資料, 再加上與二進位的差異會更加安全, 或是在二進位上提供錯誤檢查或是metadata (增加附屬屬性, 如時間), 以及適用於表達三軸  $(x, y, z)$  的空間象限, 或是 RGB 的色彩的情況上。

然而在人機介面上仍需要將 3 進位轉為 10 進位, 所以我們每 **11 個 3 進位** 00000000000 - 12002011200 來表示 **5 個 10 進位** (00000 - 99999), 因為在合理值範圍內最高位 (最左邊) 不會是 2, 所以我們用它做安全性的檢查碼, 將它設為 2 的規則, 依 11 個 3 進位的數字和取 3 的餘數(n值), 再看是否出現位置是對的筆數, 這樣我們除了可以檢查數字是否有錯外, 還可以避免中間資料被增加或篡改。所以設為 2 的情況有 3 種規則：

- 規則1：第 1, 4, 7, ... 筆時, 將 11 個 3 進位數字各別加起來取3 的餘數, 如果是 0是, 則將最高位設為 2。
- 規則2：第 2, 5, 8, ... 筆時, 將 11 個 3 進位數字各別加起來取3 的餘數, 如果是 1是, 則將最高位設為 2。
- 規則3：第 3, 6, 9, ... 筆時, 將 11 個 3 進位數字各別加起來取3 的餘數, 如果是 2是, 則將最高位設為 2。

以下是含檢查碼的11位3進位所代表的符合規則的10進位值: (00000 — 99999 任何值依出現位置都可能設為 2)

規則1:餘是0時,高位元設為2	規則2: 餘是1時,高位元設為2	規則3:餘是2時,高位元設為2
20000000000 -> 0	00000000000 -> 0	00000000000 -> 0
00000000001 -> 1	20000000001 -> 1	00000000001 -> 1
00000000002 -> 2	00000000002 -> 2	20000000002 -> 2
00000000010 -> 3	20000000010 -> 3	00000000010 -> 3
00000000011 -> 4	00000000011 -> 4	20000000011 -> 4
20000000012 -> 5	00000000012 -> 5	00000000012 -> 5
...	...	...
10000000000 -> 59049	20000000000 -> 59049	10000000000 -> 59049
10000000001 -> 59050	10000000001 -> 59050	20000000001 -> 59050
20000000002 -> 59051	10000000002 -> 59051	10000000002 -> 59051
10000000010 -> 59052	10000000010 -> 59052	20000000010 -> 59052
20000000011 -> 59053	10000000011 -> 59053	10000000011 -> 59053
10000000012 -> 59054	20000000012 -> 59054	10000000012 -> 59054
...	...	...
12002011111 -> 99994	22002011111 -> 99994	12002011111 -> 99994
12002011112 -> 99995	12002011112 -> 99995	22002011112 -> 99995
12002011120 -> 99996	22002011120 -> 99996	12002011120 -> 99996
12002011121 -> 99997	12002011121 -> 99997	22002011121 -> 99997
22002011122 -> 99998	12002011122 -> 99998	12002011122 -> 99998
22002011200 -> 99999	12002011200 -> 99999	12002011200 -> 99999

試寫一支程式排除錯誤資料後輸出總和。

### 輸入格式

第一行輸入為資料的筆數，剩下每一行都是一組 3 進位值

## 輸出格式

不含錯誤資料的10 進位總和值, 總和值小於等於 99999000

## 資料範圍

- 輸入資料範圍第一行(筆數) 值小於 1000
- 後面的每一行值介於 00000000000 – 2222222222

## 測試範例

### 輸入範例 1

```
5
00000000000
00000000001
00000000002
22222222222
20000000001
```

### 輸出範例 1

```
1
```

### 輸入範例 2

```
9
20000000000
20000000001
20000000002
00000000010
00000000011
00000000012
00000000020
00000000021
00000000022
```

## 輸出範例 2

36

## 輸入範例 3

```
11
20000000120
20000000121
20000000122
22002011200
22002011200
22002011200
22002011201
22002011201
22002011201
12222222222
12222222222
```

## 輸出範例 3

181948

## 範例說明

### 範例 1

00000000000 => 因是第 1 筆,  $n=0$ , 所以 00000000000 應該是 20000000000 才正確

00000000001 => 因是第 2 筆,  $n=1$ , 所以 00000000001 應該是 20000000001 才正確

00000000002 => 因是第 3 筆,  $n=2$ , 所以 00000000002 應該是 20000000002 才正確

22222222222 => 因是第 4 筆,  $n=0$ , 所以 22222222222 可轉為 12222222222 但其值轉為 10 進位時超過 99999

20000000001 => 因是第 5 筆,  $n=1$  正確, 所以 20000000001 可轉為 00000000001, 十進位值為 1

所以 4 筆資料是錯的, 所有合理值的十進位總和為 1

### 範例 2

20000000000, 因是第 1 筆,  $n=0$ , 合理轉為 0,

20000000001, 因是第 2 筆,  $n=1$ , 合理轉為 1,

20000000002, 因是第 3 筆,  $n=2$ , 合理轉為 2,

00000000010, 因是第 4 筆,  $n=0$ , 合理轉為 3, 3 進位 10 = 十進位 3,  $(1 \times 3 + 0 = 3)$

00000000011, 因是第 5 筆,  $n=1$ , 合理轉為 4, 3 進位 11 = 十進位 4,  $(1 \times 3 + 1 = 4)$

00000000012, 因是第6筆,  $n=2$ , 合理轉為 5, 3進位12=十進位5,  $(1 \times 3 + 2 = 5)$

00000000020, 因是第7筆,  $n=0$ , 合理轉為 6, 3進位12=十進位6,  $(2 \times 3 + 0 = 6)$

00000000021, 因是第8筆,  $n=1$ , 合理轉為 7, 3進位12=十進位7,  $(2 \times 3 + 1 = 7)$

00000000022, 因是第9筆,  $n=2$ , 合理轉為 8, 3進位12=十進位8,  $(2 \times 3 + 2 = 8)$

所以和是  $0 + 1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 = 36$

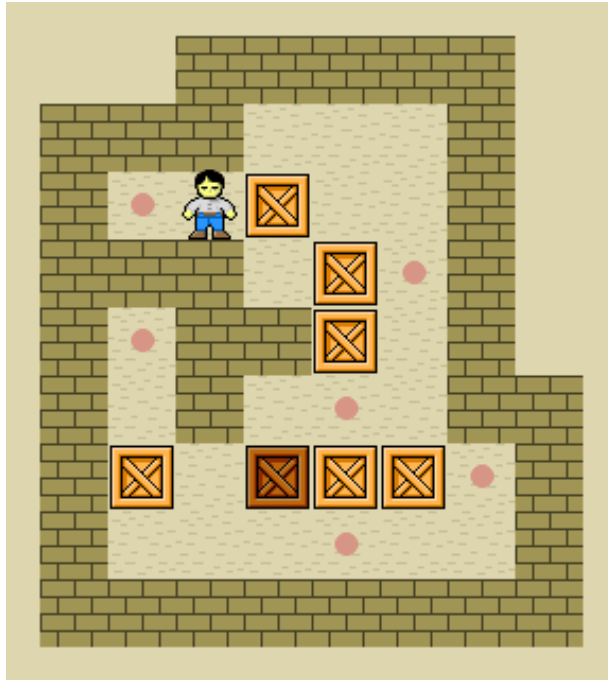


## 3\_簡單的倉庫番 (Simple Sokoban)

(10分)

### 題目敘述

倉庫番是一款經典的電子遊戲，玩家需要使用上下左右操控畫面中的人物去推箱子，將所有箱子移動到目標格子，即可通關。



喵喵遊俠覺得倉庫番太難了，他決定把地圖中的人物拿掉，並重新設計遊戲。新設計的規則是：給你一個地圖，地圖上有牆壁與數個箱子與數個目標格子，箱子與目標格子的數量一樣。每一次操作，可以直接移動任一個箱子到上下左右的位置，箱子不能移動到牆壁上也不能超出地圖範圍且任意時刻不能有兩個箱子在同一個位置上。玩家只要移動所有箱子使得所有箱子都在目標格子上即可通關。

現在喵喵遊俠根據這個規則設計了好幾個地圖，但不確定每個地圖是否都存在通關的方法。請幫喵喵遊俠判斷地圖是否能順利通關。

### 輸入格式

第一行輸入一個數字  $T$  代表接下來有  $T$  個地圖需要判斷。

對於每一個地圖，一開始輸入兩個數字  $N, M$  代表地圖大小是  $N \times M$ 。

接著輸入  $N$  行，每一行包含  $M$  個字元，第  $i$  行的第  $j$  個字元代表地圖中位置  $(i, j)$  是什麼，有四種符號：

- #：牆壁
- A：箱子
- B：目標格子
- .：空的格子

### 輸出格式

對於每個地圖，輸出一行，若存在通關的方法就輸出 YES，否則輸出 NO。

## 資料範圍

- $1 \leq T \leq 10$
- $1 \leq N, M \leq 1000$
- 保證地圖中至少有一個箱子
- 箱子的數量等於目標格子的數量

## 測試範例

### 輸入範例 1

```
1
5 5
##.B#
..AA#
B####
#A..B
...##
```

### 輸出範例 1

```
YES
```

### 輸入範例 2

```
2
3 5
A.#.B
..#..
.....
3 5
A.#.B
..#..
..#..
```

### 輸出範例 2

```
YES
NO
```

## 輸入範例 3

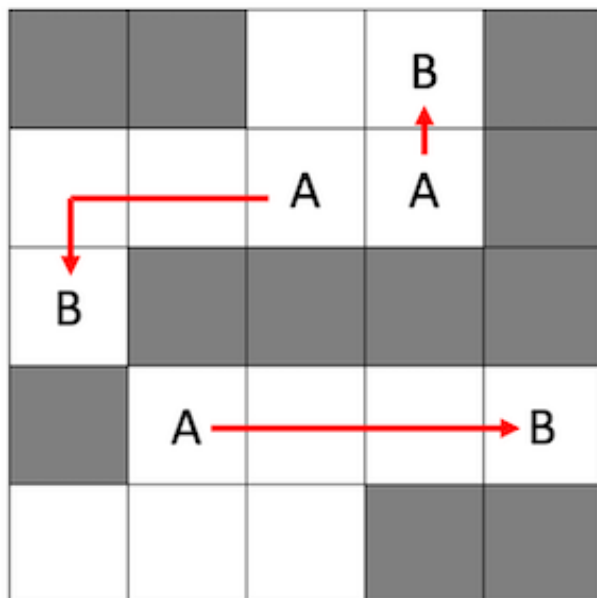
```
1
5 4
#.B.
.#..
A.#.
.#B.
.##A
```

## 輸出範例 3

NO

## 範例說明

- 以下是第一筆範例中一組可行的通關方法



- 第二筆範例中的第一個地圖可以輕易找出移動方法，至於第二個地圖，因為加上一個牆壁後箱子就無法移動到目標格，因此無通關方法。
- 第三筆範例中，左邊的箱子因為牆壁的阻擋無法移動到目標格子上，因此無通關方法。

## 4\_小綿羊想讓人告白 (Little lamb: Love Is War)

(15分)

### 題目敘述

從前，有一隻住在南極可愛的小企鵝暗戀一隻住在北極美麗的小綿羊，但由於相隔甚遠，小企鵝只敢將這樣濃烈的愛意深藏於心中，不敢表達。而知道此事的小綿羊，其實也深愛著小企鵝，於是她想了一個辦法讓小企鵝親自來告白。

小綿羊知道小企鵝很喜歡數字形狀的高麗菜，於是她決定種  $D$  顆且每顆數字形狀皆不同的高麗菜，想說可以寄給小企鵝當定情禮物，希望他可以過來找自己告白。

然而，高麗菜太重了，最後只能寄其中的四顆過去，而小綿羊特別喜歡數字  $Y$ ，因此她希望這四顆高麗菜的數字，加起來總和是  $Y$ 。現在請你幫她算出，共有幾種可能的組合，會正好四個數字加起來是  $Y$ ，好讓選擇困難症的她，可以做一個這麼多面的骰子來決定要用哪一組。

也就是說，存在多少種四元組  $(a, b, c, d)$  使得  $a < b < c < d$  且  $X_a + X_b + X_c + X_d = Y$ 。

### 輸入格式

第一行輸入  $D, Y$ ，分別代表高麗菜數量和特別數字。

第二行輸入  $D$  個整數  $X_1, X_2, \dots, X_D$ ，代表第  $i$  個高麗菜的數字是  $X_i$ 。

### 輸出格式

輸出從  $D$  個數字中取四個數字總和為  $Y$  的可能性有幾種。

### 資料範圍

- $4 \leq D \leq 2000$
- $0 \leq Y \leq 10^8$
- $0 \leq X_i \leq 10^8$

### 測試範例

#### 輸入範例 1

```
4 10
1 2 3 4
```

#### 輸出範例 1

```
1
```

## 輸入範例 2

```
5 10
1 2 3 5 4
```

## 輸出範例 2

```
1
```

## 輸入範例 3

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

## 輸出範例 3

```
2
```

## 範例說明

---

範例一：(0, 1, 2, 3) => 一種

範例二：(0, 1, 2, 4) => 一種

範例三：(0, 1, 2, 4), (1, 2, 3, 4) => 兩種

# 5\_辰辰的啞鈴 (Tommy's Dumbbell)

(10分/5分)

## 問題敘述

辰辰有一組特別強的艾洋牌啞鈴，想要他多重就可以多重。我們都知道，啞鈴的結構是需要一根槓鈴桿，兩頭插槓片。而艾洋牌啞鈴的結構比較特殊，除了需要兩邊平衡以外，兩邊所插的槓片也需要一模一樣。比方說左邊是  $5g+15g$ ，右邊也要一樣  $5g+15g$ ，不能夠  $10g+10g$ 。

現在辰辰有  $d$  槓片，他希望你幫他算算看，究竟有幾種相異的重量可以被組裝出來呢？

## 輸入格式

第一行輸入一個數字  $d$ ，代表有多少個槓片。

第二行輸入  $d$  個數字  $a_i$ ，代表每個槓片的重量。

## 輸出格式

輸出數字  $y$ ，代表有多少種相異重量可以被組裝出來。

## 資料範圍

- $1 \leq d \leq 30000$
- $1 \leq a_i \leq 100$

## 子任務

- 對於 10 分的測資滿足  $1 \leq d \leq 100$
- 其餘 5 分則是沒有額外限制

## 輸入範例 1

```
4
5 5 10 10
```

## 輸出範例 1

```
4
```

## 輸入範例 2

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

## 輸出範例 2

1

## 輸入範例 3

---

6  
3 5 8 3 9 5

## 輸出範例 3

---

4

## 範例說明

---

範例一：兩邊分別可為0g、5g、10g、15g

範例二：兩邊分別可為0g

範例三：兩邊分別可為0g、3g、5g、8g四種

## 6\_檢查機器人 (Robot)

(20分)

### 題目敘述

一個  $N$ -檢查機器人的機制如下：

- 它有  $N$  個「檢查門」，每一個「檢查門」裡頭可以放一顆彈珠或者不放彈珠。每一個「檢查門」均設有壓力裝置，可以檢測這個檢查門是否有被放彈珠進去。
- 機器人有內建的設定，代表它認為一些「檢查門」應該要有彈珠，一些檢查門不應該要有彈珠。
- 當選擇完哪些「檢查門」要放彈珠，並放入彈珠了之後，可以按「檢查」按鈕
- 按「檢查」按鈕之後，倘若有奇數個「檢查門」符合「此檢查門內建設定應該有彈珠且此檢查門實際上真的有彈珠」，則它會輸出 `1`；否則它會輸出 `0`。

你發現了已經有人做了  $M$  次測試同一個  $N$ -檢查機器人的資料！請寫一支程式判斷這個  $N$ -檢查機器人的內建設定，或者判斷這個設定不可能存在吧！

### 輸入格式

輸入的第一行有兩個數字  $N, M$ ，分別代表機器人有幾個「檢查門」與進行了幾次測試。

接下來的  $M$  行，其中第  $i$  ( $1 \leq i \leq M$ ) 行將有一個字串  $S_i$  和一個數字  $c_i$ ，滿足  $S_i$  為一個長度  $N$  的 `0-1` 字串，而  $S_i$  的第  $j$  個字元代表第  $i$  次測試的第  $j$  個「檢查門」裡有沒有放入彈珠。 $c_i$  為 `0` 或 `1`，代表第  $i$  次測試機器人的輸出。

### 輸出格式

若給定的資料沒有解答，請輸出 `INCONSISTENT`；否則，請輸出 `CONSISTENT`，並在第二行輸出  $N$  個空白分隔的數字  $x_i$ ，滿足  $x_i$  為 `0` 或 `1`，且  $x_i$  代表機器人內部覺得第  $i$  個「檢查門」該不該放彈珠。若答案為 `CONSISTENT`，則輸出任何一組解即可。

### 資料範圍

- $1 \leq N, M \leq 500$
- $|S_i| = N$

### 資料範例

### 測試範例



## 輸入範例 1

```
3 3
100 1
010 0
001 1
```

## 輸出範例 1

```
CONSISTENT
1 0 1
```

## 輸入範例 2

```
4 2
1010 0
0101 1
```

## 輸出範例 2

```
CONSISTENT
0 1 0 0
```

## 輸入範例 3

```
3 3
010 0
100 1
110 0
```

## 輸出範例 3

```
INCONSISTENT
```

## 範例說明

gww

對於第一筆測資，因為每一行只有一個 **1**，所以那個 **1** 所在的彈珠就代表了答案的對應格是不是 **1**；對於第二筆的第一行，答案是 **0100** 的時候沒有任何彈珠符合答案，為偶數；對於第二行，放入 **0101** 的時候第二個檢查門符合，為 **1** 是奇數。

可以被證明第三筆測資並無符合的答案存在。

## 7\_糖果商店 (Candy Shop)

(5分/15分)

### 題目敘述

喵喵遊俠很喜歡吃糖果，因此他決定自己開一家糖果店來賺錢。他的店有販賣  $N$  種糖果，編號 1 到  $N$ ，初始每一種糖果的數量都是 0。糖果店到目前為止已經營業了  $P$  天，每一天喵喵遊俠都會選定一段連續編號的糖果進行補貨或者是有一段連續編號的糖果售出，也就是說，每一天都會有一段編號區間  $l$  到  $r$  的糖果的數量一起加  $y$ 。若  $y$  大於等於 0 代表補貨，若  $y$  是負的代表售出，注意到糖果的數量可以是負數，代表顧客先預訂了該種類的糖果。

喵喵遊俠好奇某些種類的糖果的數量是在哪一天第一次大於等於  $k$ 。請你寫一支程式幫助喵喵遊俠。

### 輸入格式

第一行輸入三個整數  $N, P, Q$  代表糖果有  $N$  種、糖果店已經營業了  $P$  天以及有  $Q$  筆詢問。

接著輸入  $P$  行，每一行有三個整數  $l_i, r_i, y_i$ ，其中的第  $i$  行代表在第  $i$  天編號  $l_i$  到  $r_i$  的糖果數量都加  $y_i$ 。

接著輸入  $Q$  行，每一行有兩個整數  $x_i, k_i$ ，代表詢問第  $x_i$  種糖果的數量是在哪一天第一次大於等於  $k_i$ 。

### 輸出格式

對於每一筆詢問，若第  $x_i$  種糖果的數量從來沒有大於等於  $k_i$  過則輸出  $-1$ ，反之則輸出答案。

### 資料範圍

- $1 \leq N, P, Q \leq 5 \times 10^5$
- $1 \leq l_i \leq r_i \leq N$
- $-10^9 \leq y_i \leq 10^9$
- $1 \leq x_i \leq N$
- $1 \leq k_i \leq 10^9$

### 子任務

- 子任務 1 滿足  $1 \leq P, Q \leq 2000$

### 測試範例

#### 輸入範例 1

```
5 3 3
3 4 2
2 5 10
4 4 7
4 1
1 2
4 19
```

## 輸出範例 1

```
1
-1
3
```

## 輸入範例 2

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

## 輸出範例 2

```
4
3
-1
4
```

## 輸入範例 3

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

## 輸出範例 3

```
-1
2
```

## 範例說明

- 第一筆範例中，編號四的糖果數量在這三天的變化是 2, 12, 19，第一天糖果數量就大於等於 1 因此第一筆詢問要輸出 1。第三天糖果數量第一次大於等於 19 因此第三筆詢問要輸出 3。而因為編號 1 的糖果的數量經過三天後都沒有變動，因此第二筆詢問輸出 -1。

- 第二筆範例中，編號八的糖果數量在這四天的變化是 3, 1, 5, 10，因此數量第一次大於等於 10 是第四天的時候因此第一筆詢問輸出 4。編號十的糖果數量在這四天的變化是 0, 0, 4, 4，因此數量第一次大於等於 1 是第三天的時候因此第二筆詢問輸出 3。編號四的糖果數量在這四天的變化是 3, 1, 1, 6，因為不存在數量大於等於 7 的日子因此第三筆詢問輸出 -1。編號二的糖果數量在這四天的變化是 3, 3, 3, 8，因此數量第一次大於等於 8 是第四天的時候因此第四筆詢問輸出 4。
- 第三筆範例中，編號一的糖果數量在這三天的變化是 -10, 0, 0，因此第一筆詢問輸出 -1。編號二的糖果數量在這三天的變化是 0, 10, 0，因此數量第一次大於等於 10 是第二天的時候，因此第二筆詢問輸出 2。

## 8\_圓形和三角形 (Circle and triangle)

(20 分)

### 題目敘述

約翰有  $A$  個相異的白色圓形積木，編號為  $1 \sim A$ ，還有  $B$  個相異的三角形白色積木，編號為  $A + 1 \sim A + B$ 。

但是約翰討厭白色，因此他買了  $M$  罐染劑將這些積木著色，這些染劑的編號分別為  $1 \sim M$ ，而且都不會是白色、並且兩兩相異。一罐染劑可以使用無限多次。

對於每個圓形，約翰會隨機使用  $1 \sim K$  的染劑來進行染色。








對於每個三角形，約翰會隨機使用  $K + 1 \sim M$  的染劑來進行染色。

然而你並不確定  $M$  與  $K$  的確切數值，你只知道  $2 \leq M \leq N$  且  $1 \leq K \leq M - 1$ ，其中  $N$  是  $M$  的上界。任兩種染色結果不同，若且唯若  $M$  不同、或者  $K$  不同、或者存在一個積木在這兩個結果中被染上不同的顏色。令  $X(M)$  為使用  $M$  種染劑能夠產生的不同結果數。

實際上可以發現，約翰使用了  $M$  種染劑的機率為  $\frac{X(M)}{X(2)+X(3)+\dots+X(N)}$ 。

已知每一罐染劑的價格皆為  $T$  元，請輸出約翰所花費的金額的期望值。

由於答案可能並非整數，令其為  $\frac{Y}{Z}$ ，則請輸出  $YZ^{-1}$  模  $P$  後的數值即可。

$M=2, K=1$		
$M=3, K=1$		
$M=3, K=2$		
		

紅色的編號為一、橘色的編號為二、黃色的編號為三。

在以上圖片中，展示了  $N = 3, A = 2, B = 1$  的所有不同情況，也就是範測一。

請注意到，雖然在  $M = 3, K = 1$  還有  $M = 3, K = 2$  的情況中有一種染色方法是相同的，但是因為兩者的  $K$  的值並不相同，因此算是兩種不同的結果。

另外，同個形狀的積木之間是相異的，因此  $M = 3, K = 2$  中左下角與右上角為不同的結果。

### 輸入格式

輸入的第一行會包含一個整數  $C$ ，代表有幾筆測資。

接下來的  $T$  行，每行會包含五個正整數  $N$ 、 $A$ 、 $B$ 、 $T$ 、 $P$ ，分別代表約翰最多會買幾瓶染劑、圓形的數量、三角形的數量、染劑的價格、還有取模的數字  $P$ 。

## 輸出格式

對於每筆測資請輸出一個非負整數，代表約翰所花費的金額的期望值模  $P$  後的數值。

## 資料範圍

$$1 \leq C \leq 10^3$$

$$2 \leq N \leq 10^6, \text{ 保證所有測資中的 } N \text{ 加總不超過 } 10^6$$

$$1 \leq A, B, T \leq 10^6$$

$$10^8 \leq P \leq 2 \times 10^9, \text{ 保證 } P \text{ 會是質數}$$

## 測試範例

### 輸入範例 1

```
1
3 2 1 1 1000000007
```

### 輸出範例 1

```
857142866
```

### 輸入範例 2

```
9
2 2 2 1 1000000007
2 1 3 2 1000000009
3 1 1 3 1000000007
3 2 1 1 1000000009
3 2 2 2 998244353
3 3 1 3 998244353
3 3 2 1 1000000009
3 3 3 2 998244353
4 2 2 3 1000000009
```

## 輸出範例 2

```
2
4
800000014
142857147
110916045
181498982
384615391
117440518
813953507
```

## 輸入範例 3

```
1
1000000 1000000 999999 999998 998244353
```

## 輸出範例 3

```
101688907
```

## 範例說明

在範例一中，約翰可能會買兩罐染劑或是三罐染劑。

若約翰買了兩罐染劑，則約翰必須將一號染劑用在圓形上、二號染劑用在三角形上，因此  $X(2) = 1$ 。

若約翰買了三罐染劑，則有以下兩種情況：約翰將前兩個染劑用在圓形上、第三個染劑用在三角形上；還有約翰將第一個染劑用在圓形上、後兩個染劑用在三角形上。

前者共有  $2^2 \times 1^1 = 4$  種不同結果、後者共有  $2^1 \times 1^2 = 2$  種結果，因此  $X(3) = 4 + 2 = 6$ 。

那麼，約翰花費金額的期望值為：
$$\frac{2 \times X(2) + 3 \times X(3)}{X(2) + X(3)} = \frac{20}{7}。$$

$20 \times 7^{-1}$  在模 1000000007 之下是 857142866，因此輸出 857142866。

## 9\_專業樹語 (Tree)

(20分)

### 題目敘述

這是一個互動題

小 T 是一個專業的間諜，有一天他嘗試潛入敵國王城竊聽機密情報。沒想到大意的小 T，竟然把自己六歲大的女兒小 Y 獨自留在家中。當他好不容易潛入王城時，竟然從電報得知，小 Y 被敵國間諜綁架壓入城堡當作人質。雖然情勢對於小 T 極為不利，但小 T 還有一個最後手段，也就是使用小 Y 的超能力「讀心術」。小 T 想透過小 Y 的遠距讀心術，將自己調查到的王城地圖，傳達給小 Y 讓她能夠順利脫困。但因為小 Y 年紀還小，所以只能讀懂二進位，聰明的你，可以幫小 T 和小 Y 設計一個方法來傳輸這個王城地圖嗎？

解決這題的流程如下：

- 你將要設計兩個函數 `send`, `recv`
- `send` 代表小 T 的策略，這個函數將會收到一顆有根樹，你需要將樹透過盡量短的二進位字串傳達給小 Y
- `recv` 將會收到剛剛 `send` 所生成的二進位字串，小 Y 需要正確的輸出有根樹
- 注意：你的兩個函數將會被分開執行，你將無法透過除了 `send` 所回傳的字串以外的資料進行溝通 (ex: 全域變數)

### 輸入與輸出

函數的界面包含：

```
std::string send(int N, std::vector<std::pair<int, int>> edges);
```

- `N` 代表王城大小，也就是樹的節點數量
- `edges` 是 `N-1` 個 `pair<int,int>`，`edges[i]` 代表跟第  $i$  條邊相鄰的節點  $u_i, v_i$
- 請回傳一個由 `'0'` 與 `'1'` 組成的字串
- 節點 0 是根節點

```
std::vector<std::pair<int, int>> recv(std::string code);
```

- `code` 一個 `'0'` 與 `'1'` 組成的字串代表 `send` 所傳來的機密資料
- 請回傳一個長度是 `N-1` 的 `vector<pair<int,int>>`，第  $i$  個 `pair` 代表跟第  $i$  條邊相鄰的節點  $u_i, v_i$

### 分數規則

- 首先要拿到分數，`recv` 回傳的有根樹，形狀必須和 `send` 獲得的樹完全相同
- `send` 所回傳的密碼必須是由 `'0'` 或 `'1'` 組成
- 兩個函數所使用的時間總和，必須小於時間限制

$A$  樹和  $B$  樹形狀相同即代表對於樹  $A$  存在一種點編號的調整方法(除了根固定以外可以隨意交換其他點編號)，讓任何  $B$  樹上的邊  $(u, v)$  都存在於編號調整過後的  $A$  樹上。

每筆測資如果節點數為  $N$ ，輸出長度為  $L$ ，則對應分數如下



- $\max(\min(100, \frac{46N-L}{44N} \times 70 + 30), \min(30, \frac{1.5N^2-L}{N^2} \times 30), 0)/5$

給分的方法是每筆測資分數的平均值

## 資料範圍

- $1 \leq N \leq 200,000$
- $0 \leq u_i, v_i < N$

## 測試範例

透過 cms 上提供的 `compile_cpp.sh` 編譯執行

```
chmod +x compile_cpp.sh
./compile_cpp.sh
./Tree
```

注意：這個測試程式跟 cms 上的不同，不會將兩個函數分開執行

### 輸入範例 1

```
3
0 1
0 2
```

### 輸出範例 1

```
3
0 2
0 1
```

### 輸入範例 2

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

## 輸出範例 2

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

## 輸入範例 3

```
1
```

## 輸出範例 3

```
1
```

## 範例說明

- 輸入 1 與輸入 2 的邊雖然編號不同，但兩顆樹以 0 為根的形狀相同，所以是正確解答。
- 輸出的解答不是唯一

# 10\_閃亮☆閃亮☆星空祭 (Pika ☆ Pika ☆ Starry Sky Festival)

(1 分 / 2 分 / 3 分 / 4 分 / 5 分 / 10 分)

## 題目敘述

星空國小一年一度的文化季——閃亮☆閃亮☆星空祭終於要到了！星川、空澄與夏色三位烘焙社的學生，在 Pinky 老師的帶領之下，準備拿出最棒的蛋糕在文化祭上販賣。

他們預計要烤一個高度為 1 的長方體的蛋糕，而他們準備的材料恰好可以做出體積為  $N$  的蛋糕。在烘焙教室中，一共有兩把尺。一把有  $A$  個刻度，並以 1 為刻度單位；另一把有  $B$  個刻度，並以  $\sqrt{86}$  為刻度單位（或許是 Pinky 老師剛看完《86－不存在的戰區－》？）。而同時使用這兩把尺時，他們可以量出所有形如  $a + b\sqrt{d}$ ,  $|a| < A$ ,  $|b| < B$  的長度。他們想要使用這兩把尺，量出蛋糕的長跟寬以製作模具。

請你幫幫他們，算出他們可以製作出哪些尺寸的蛋糕？

簡而言之，給定  $N, A, B$ ，請告訴他們有哪些整數組  $(a, b, x, y)$  滿足

1.  $(a + b\sqrt{86})(x + y\sqrt{86}) = N$
2.  $0 \leq a + b\sqrt{86}, x + y\sqrt{86}$
3.  $|a|, |x| < A$
4.  $|b|, |y| < B$

輸出請按照  $(a, b, x, y)$  的字典序排序。

## 輸入格式

輸入有一行，包含三個以空白分隔的整數  $N, A, B$ ，意思如題敘所述。

## 輸出格式

第一行請輸出一個整數  $t$ ，表示有幾種不同尺寸的蛋糕。

接著請輸出  $t$  行，每行包含四個以空白分隔的整數  $a, b, x, y$ ，表示  $a + b\sqrt{86} \times x + y\sqrt{86}$  是一個可行的尺寸。

輸出請按照  $(a, b, x, y)$  的字典序排序。

## 資料範圍

- $1 \leq N \leq 10^9$
- $1 \leq A, B \leq 10^{18}$

## 子任務

- $B = 1$  (2%)
- $A, B \leq 20$  (3%)
- $A, B \leq 2000$  (15%)
- $N = 1$  (20%)
- $N \leq 1000$  (20%)
- 無限制 (40%)

## 測試範例

### 輸入範例 1

```
10 20 20
```

### 輸出範例 1

```
8
-18 2 9 1
-9 1 18 2
1 0 10 0
2 0 5 0
5 0 2 0
9 1 -18 2
10 0 1 0
18 2 -9 1
```

### 輸入範例 2

```
86 43 43
```

### 輸出範例 2

```
1
0 1 0 1
```

### 輸入範例 3

```
84 200 10
```

### 輸出範例 3

```
20
1 0 84 0
2 0 42 0
3 0 28 0
4 0 21 0
6 0 14 0
7 0 12 0
10 -1 60 6
10 1 60 -6
```

```

12 0 7 0
14 0 6 0
20 -2 30 3
20 2 30 -3
21 0 4 0
28 0 3 0
30 -3 20 2
30 3 20 -2
42 0 2 0
60 -6 10 1
60 6 10 -1
84 0 1 0

```

## 不一定有用的提示

在  $\mathbb{Z}[\sqrt{86}] = \{a + b\sqrt{86} | a, b \in \mathbb{Z}\}$  中，任何的數字仍然擁有「唯一」的「質因數分解」。

也就是，你不會出現像  $(3 + \sqrt{5})(3 - \sqrt{5}) = 4 = 2 \cdot 2$  的情況。

注意：兩個質因數分解  $n = p_1 \dots p_k = q_1 \dots q_l$  是相同的若且唯若  $k = l$  而且存在一個排列  $a_i$  使得  $p_{a_i} | q_i$  且  $q_i | p_{a_i}$ 。

例如  $6 = -2 \cdot -3 = 3 \cdot 2$  是同一個質因數分解，而  $9 = 3 \cdot 3 = (9 + 6\sqrt{2})(9 - 6\sqrt{2})$  也是同一個質因數分解： $3/(9 + 6\sqrt{2}) = 3 - 2\sqrt{2}$  是整數； $(9 + 6\sqrt{2})/3 = 3 + 2\sqrt{2}$  也是整數。

簡而言之： $\mathbb{Z}[\sqrt{86}]$  是一個 UFD。

# 11\_人力資源分配 (Human Resource Allocation)

(4分/6分/7分/8分)

## 題目敘述

YTP (Yummy TeaPot) 公司最近事務繁忙，為了在最低的花費下解決眾多雜務，YTP 公司請了幾位專業的預測專家為他們分析了接下來他們  $N$  天將會遇到的  $N$  個任務，每個任務都各自包含一個能力區間  $[l, r]$  和懲罰金  $t$ ，且必須要找一位「幫手」完成，假設接受這項任務的幫手一開始的能力值為  $p$ ，則：

- 若  $p < l$  或  $p > r$ ，則會需要花費  $t$  塊錢。
- 否則若  $l \leq p \leq r$ ，則不需要花費任何錢，但取而代之的是，這位幫手的能力值會在執行完該項任務後轉變為  $l + r - p$ 。

注意到，無論幫手能力值為多少，他總是能完成任務，只會有花費上的差距而已。

同時，預測專家們還預測出了接下來  $N$  天當中，每天都會出現一位新的「幫手」，這  $N$  位幫手都會有各自不同的初始能力值  $p_i$ 。這些幫手的價格昂貴，請了第  $i$  位進公司，YTP 公司就必須付  $c_i$  塊錢出去。而一位幫手一旦只要被公司解僱之後，就再也不會回來了；反之只要有幫手在公司內，該天的任務就一定得交給他處理，由於這些幫手互相仇視，因此如果同一天有超過一位幫手在公司內，他們就會大打出手，這將導致公司損失慘重。

白話一點的講，對於第  $i$  天出現的幫手，YTP 公司必須決定是否要花  $C$  塊錢請這位幫手進來公司，若決定要，則 YTP 公司必須解僱當前在公司內的幫手，並讓新的幫手開始處理第  $i$  天的任務；否則，原本在公司內的幫手會繼續處理該天的任務，並且不用再多付請該位幫手進公司的錢。

在需要完成所有任務的前提下，你可以幫助 YTP 公司先行估計出他們最小的花費嗎？

注意到你一定得請第一天的幫手，不然會沒有幫手處理第一天的任務。

## 輸入格式

輸入首行有一個正整數  $N$ ，代表接下來  $N$  天都有一個任務。

接下來一行  $N$  個非負整數  $p_1, \dots, p_N$ ，代表第  $i$  天出現的幫手能力值為  $p_i$ 。

接下來一行  $N$  個非負整數  $c_1, \dots, c_N$ ，代表第  $i$  天出現的幫手花費為  $c_i$ 。

最後接著  $N$  行，第  $i$  行三個非負整數  $l_i, r_i, t_i$ ，代表第  $i$  個任務的能力區間為  $[l_i, r_i]$ 、懲罰金為  $t_i$ 。

## 輸出格式

輸出一個非負整數於一行，代表在完成所有任務的前提下，YTP 公司最小的花費。

## 資料範圍

- $1 \leq N \leq 2 \times 10^5$
- $0 \leq t_i, p_i, c_i \leq 10^9$
- $0 \leq l_i \leq r_i \leq 10^9$

## 子任務

- 子任務 1 滿足  $N \leq 18$
- 子任務 2 滿足  $N \leq 3000$
- 子任務 3 滿足  $l_i = r_i$
- 子任務 4 沒有特別限制

## 測試範例

### 輸入範例 1

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

### 輸出範例 1

```
3
```

### 輸入範例 2

```
3
1 3 7
4 2 10
1 100 40
2 4 10
100 110 20
```

### 輸出範例 2

```
14
```

### 輸入範例 3

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

## 輸出範例 3

8

## 範例說明

在第一筆範例中，最佳的方法如下：

- 在第一天請第一位幫手並花費 1 塊錢，由於第一位幫手的能力值為 1 落在  $[1, 3]$  區間內，因此不需要額外的花費就可以完成第一天的任務，但第一位幫手的能力值會變為  $1 + 3 - 1 = 3$ 。
- 在第二天不請第二位幫手，而是直接讓第一位幫手繼續處理任務，注意由於第一位幫手的能力值現在為 3，這樣的花費為 1。即使第二位幫手不需要任務上的花費，但聘請他的花費為 4，較不划算。
- 在第三天解僱第一位幫手、並花費 1 塊前聘請第三位幫手，由於第三位幫手的能力值為 5 落在  $[5, 8]$ ，因此不需要額外的花費就可以完成第三天的任務，但第三位幫手的能力值會變為  $5 + 8 - 5 = 8$ 。總共的花費為  $1 + 1 + 1 = 3$  塊錢，也是最小的花費。

在第二筆範例中，最佳的方法是直接請第一位幫手處理全部的任務，總花費為 14，注意到即使在第二天的當下，第一位幫手的花費顯著的比第二位幫手高，但若在當下聘請第二位幫手，會導致第三天的花費變高，使得總花費變得更高。

在第三筆範例中，最佳的方法是依序聘請第一、三、五位幫手，最小花費為 8。注意到該筆範例符合第三個子任務的限制。



# 12\_Meow (How many different breeds of cats)

(本題沒有中文題本)

(25 points)

## Statement

Today, you came to a cat village. There are  $N$  cats in this village.  
Each cat is very cute, and all the  $N$  cats are very cleverly lined up in a row.

As we all know, there are  $B$  breeds of cats in the world.

Breed of each cat from left to right in this village is  $a_1, a_2, \dots, a_N$ .

Because you are a coder and cat lover who is very interested in cat breeds,  
you may be curious about the following type of question:

given  $K$  intervals,  $[l_1, r_1], [l_2, r_2], \dots, [l_K, r_K]$ ,

how many different breeds of cats which appear a positive even number times in the union (that is, if a cat appears in multiple intervals, it should be counted only once) of the  $K$  intervals?

Now, give you  $M$  such questions, can you answer all correctly?

## Input Format

The first line contains three positive integers

$N, B$  and  $M$  --- the number of cats in this village, the number of breeds of cats in the world and the number of questions you need to answer.

The second line contains  $N$  positive integers  $a_1, a_2, \dots, a_N$ .

The following  $M$  lines contain questions, one per line.

The first integer in the line is  $K$ , followed by  $2 \times K$  integers

$l_1, r_1, l_2, r_2, \dots, l_K, r_K$  in the same line.

## Output Format

Print  $M$  lines. The  $i$ -th line contains one integer ---

the number of breeds of cats which appear a positive even number times in the given intervals.

## Constraints

- $1 \leq N, B, M \leq 10^5$
- $1 \leq a_i \leq B$
- $1 \leq K \leq 10^5$
- $1 \leq l_i, r_i \leq N$
- It is guaranteed that the sum of all  $K$ 's is not greater than  $10^5$ .

## Test Cases

## Input 1

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

## Output 1

```
0
0
1
1
1
```

## Illustrations

There are 5 cats of three breeds (1, 3, 2, 1 and 2) in this sample. An explanation of each question is shown below:

1. The first question contains 1 interval -  $[1, 2]$ . All the breeds of cats in the union of all intervals from left to right is 1 and 3. Since no breeds of cats appear even times, the answer is 0.
2. The second question contains one interval -  $[1, 3]$ . All the breeds of cats in the union of all intervals from left to right are 1, 3, and 2. Since no breeds of cats appear even times, the answer is 0.
3. The third question contains one interval -  $[1, 4]$ . All the breeds of cats in the union of all intervals from left to right are 1, 3, 2, and 1. Since only one breed (1) appears even times, the answer is 1.
4. The forth question contains two intervals -  $[1, 2]$  and  $[4, 5]$ . All the breeds of cats in the union of all intervals from left to right are 1, 3, 1, and 2. Since only one breed (1) appears even times, the answer is 1.
5. The fifth question contains three intervals -  $[1, 1]$ ,  $[3, 3]$  and  $[5, 5]$ . All the breeds of cats in the union of all intervals from left to right are 1, 2, and 2. Since only one breed (1) appears even times, the answer is 1.