

# 送分題 — solution

---

(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？

檔案是否用 `STDIN` 輸入？

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！

## 問題敘述

試寫一程將輸入的兩個整數相加後輸出。

## 輸入格式

A B

兩個正整數，中間以空白區隔。

## 輸出格式

C

代表兩個正整數之和。

## 資料範圍

A, B, C 皆為小於 9999999 的正整數。

## 資料範例

### 輸入範例 1

2756 707

### 輸出範例 1

3463

### 輸入範例 2

23024 19109

### 輸出範例 2

42133

### 輸入範例 3

6253 14484

### 輸出範例 3

20737

### 範例說明：

兩個整數 2756 及 707 相加後輸出 3463。

### 補充說明：

以下為 STDIN /STDOUT 的方式輸入輸出的 Sample Code：

#### [Sample Code: C]

```
#include <stdio.h>

int main() {

    int a, b;
    scanf("%d %d", &a, &b);
    printf("%d\n", a+b);
    return 0;

}
```

### [Sample Code: C++]

```
#include <iostream>

int main() {

    int a, b;
    std::cin >> a >> b;
    std::cout << a + b << std::endl;
    return 0;

}
```

### [Sample Code: Java]

```
import java.util.Scanner;

public class solution {
    public static void main(String args[]) {
        Scanner in = new Scanner(System.in);
        int a, b;
        a = in.nextInt();
        b = in.nextInt();
        System.out.println(a + b);
    }
}
```

### [Sample Code: Python]

```
import sys

a, b = (int(x) for x in sys.stdin.readline().strip().split())
sys.stdout.write("%d\n" % (a + b))
```

# 問題 1 – 地雷掃描儀 (minesweeper)

---

(5 分)

## 問題敘述

想必大家都有玩過叫做「踩地雷」的電腦遊戲。現在給定地雷的位置資訊，請你實作一個程式，計算出每個格子的相鄰八格的地雷總數量。

所謂相鄰的定義為：格子的上、下、左、右以及左上、左下、右上、右下，不包含自己。

## 輸入格式

輸入第一行包含兩個正整數  $n, m$ ，分別代表地圖的長與寬。

接下來  $n$  行每行有  $m$  個整數，1 代表那格有地雷，0 代表沒有。

## 輸出格式

輸出  $n$  行，每行有  $m$  個整數，分別代表相鄰某格的格子們的地雷總數。

詳見範例輸出。

## 資料範圍

- $1 < n, m < 1024$ ,
- $a_{ij} \in \{0,1\}$ ，其中  $a_{ij}$  為地圖矩陣上的資訊。

## 資料範例

以下範例測資說明每筆各挑兩個點作為示範。

### 輸入範例 1

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

### 輸出範例 1

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

說明：以第一橫列第一格 (1, 1) 為例，在其附近有第一橫列第二格 (1, 2) 以及第二橫列第一格 (2, 1) 為地雷（即數字為 1），所以第一橫列第一格為輸出 2。再以第三橫列第三格 (3, 3) 為例，附近沒有數字為 1，所以輸出 0。

### 輸入範例 2

```
5 5
1 1 1 1 1
1 0 0 0 1
1 0 0 0 1
1 0 1 0 1
1 1 1 1 1
```

### 輸出範例 2

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

說明：(1, 1) 附近有兩個相鄰點 (1, 2), (2, 1) 為 1，所以輸出 2。(3, 3) 附近只有一個相鄰點 (4, 3) 為 1，所以輸出 1。

### 輸入範例 3

```
5 6
0 1 1 1 1 0
1 0 1 0 1 0
1 1 1 1 1 1
1 0 1 0 1 0
1 1 1 1 0 0
```

### 輸出範例 3

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

說明：(2, 2) 附近除了 (1, 1) 以外的七個相鄰點都為 1，所以輸出 7。(2, 4) 附近八個相鄰點全為 1，所以輸出 8。

## 問題 2 – 呆呆企鵝 (penguin)

(5 分)

### 問題敘述

呆呆企鵝是一款兼具趣味與環保意識的可愛桌遊。這款遊戲中每位玩家都會化身為一隻隻可愛的呆呆企鵝，在一塊因為全球暖化而日漸消融的長條型浮冰上苦中作樂地競速。從浮冰的一頭開始，最先到達浮冰的另一頭，且沒有落下就能成為眾多呆呆企鵝中的佼佼者——勝利企鵝！

呆呆企鵝的遊玩規則如下：遊戲的一開始會有一塊長條形的浮冰，長度為  $n$  ( $100 \leq n \leq 10000$ ) 單位，最初每位玩家的呆呆企鵝會位於單位 0。當還沒有勝利者誕生之前，每輪所有玩家都要按照玩家編號由小到大輪流擲骰子，決定當輪該玩家的呆呆企鵝要往前跳幾單位。然「鵝」，因為暖化而消融的冰水使得浮冰滑溜溜的，造成呆呆企鵝會因為著陸時站不住腳而「哎呀！滑倒！」，並依照每個單位的滑度  $k_i$  ( $0 \leq k_i \leq 5, 0 \leq i < n$ ) 向末端滑行  $k_i$  個單位。

特別需要注意的是：每輪所有呆呆企鵝移動後浮冰都會以  $m$  ( $1 \leq m \leq 5$ ) 單位的速度由起始端與末端消融！如果當時有呆呆企鵝站在消融的浮冰上，亦或是往前跳與滑動的過程中超出浮冰的範圍，則這些企鵝將落入海中，失去成為勝利企鵝的資格！若想從呆呆企鵝成為勝利企鵝、躍入「鵝生勝利組」，關鍵即在於某輪跳躍、滑行並恰好停止於浮冰的末端最後一單位或成為所有企鵝中的唯一倖存鵝！當勝利企鵝誕生之時則遊戲立刻結束，由該呆呆企鵝代表的玩家獲勝。

現在，由於桌遊「呆呆企鵝」實在過於流行，發行商決定將其改版成線上遊戲，因此你被請來撰寫「呆呆企鵝」的單機版模擬程式：給定一局的浮冰長度  $n$ 、消融速度  $m$ 、每單位浮冰的滑度  $k_i$  與該玩家於遊戲中每輪將骰出的數字  $d$  ( $1 \leq d \leq 6$ )，請在不考慮其他玩家的情況下模擬遊戲的進行（也就是說，你只需要模擬一個玩家的狀況），若玩家成功到達末端請輸出「Winner Penguin」，反之則輸出「Dumb Dumb Penguin」，並輸出直到遊戲模擬結束前共骰了幾次骰子。

### 輸入格式

```
n m t
k0 k1 k2 ... kn-1
d0 d1 d2 ... dt-1
```

輸入的第一行包含三個正整數  $n, m, t$ ，分別代表浮冰的長度，以及浮冰兩端每輪消融的速度與給定將骰出的數列長度。

第二行包含  $n$  個整數  $k_i$ ，代表由單位 0 開始每單位浮冰的滑度。

第三行包含  $t$  個正整數  $d_j$  ( $1 \leq d_j \leq 6$ )，保證在骰完數字前所有浮冰必已消融，遊戲必結束。

## 輸出格式

請輸出兩行，第一行輸出該玩家於遊戲模擬的結果，若勝利請輸出「Winner Penguin」，反之則輸出「Dumb Dumb Penguin」。另外無論勝利與否，第二行請輸出「Survived [n] rounds」，其中 [n] 為該玩家已骰出的骰子次數。

## 資料範圍

- $100 \leq n \leq 10000$
- $1 \leq m \leq 5$
- $t \leq n$ ，且保證在骰完給定的骰子數列前必有贏家產生。

## 資料範例

### 輸入範例 1

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

### 輸出範例 1

```
Winner Penguin
Survived 1 rounds
```

### 解釋 1

玩家於第一輪骰出 3，企鵝跳躍三單位後抵達滑度為 1 的浮冰上又滑了一單位，共四單位到達終點。共骰了一次骰子故生存了一輪。

### 輸入範例 2

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

### 輸出範例 2

```
Dumb Dumb Penguin
Survived 1 rounds
```

### 解釋 2

玩家於第一輪中骰出 1，並因滑度為 2 而導致企鵝停止於單位 3 的浮冰，後因浮冰從兩端各融化 3 單位而落水失去資格。共骰了一次骰子故生存了一輪。

### 輸入範例 3

```
10 4 6  
2 1 3 4 1 2 0 4 3 5  
4 3 2 2 2 2
```

### 輸出範例 3

```
Dumb Dumb Penguin  
Survived 2 rounds
```

### 解釋 3

玩家於第一輪中骰出 4 後移動四單位來到滑度為 1 的浮冰上因而再度移動一單位，共 5 單位，浮冰此輪從兩端各融化 4 單位；第二輪中，玩家骰出 3，因而移動 3 單位而滑出浮冰之外，共骰了兩次骰子故生存了兩輪。



## 問題 3 – 金字塔 (pyramid)

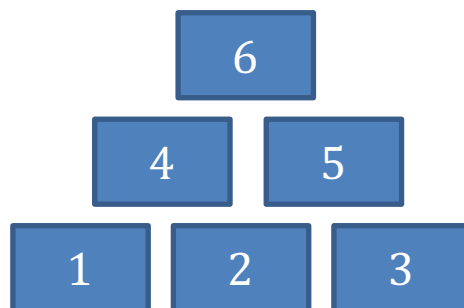
---

(5 分)

### 問題敘述

簡易金字塔看起來像個三角形，如果有 6 個磚塊可以像左邊一樣蓋成 3 層，假設不考慮建築結構，那麼越多的磚塊就表示可以蓋越多層，但如果不是剛好的話（如：5 個磚塊），那麼就只能蓋 2 層。

如果提供磚塊的個數，你能計算出最高能蓋多少層嗎？（下面層必須比上面層多一個）



### 輸入格式

輸入的第一行包含一個正整數  $n$ ，代表磚塊數。

### 輸出格式

輸出一個整數於一行，代表最高可以蓋的層數。

### 資料範圍

$0 < n < 2^{31}$ ，且  $n$  為正整數。

### 資料範例

#### 輸入範例 1

6

#### 輸出範例 1

3

#### 解釋 1

如右上圖，6個磚塊可以堆的層數為3層。

### 輸入範例 2

5

### 輸出範例 2

2

### 解釋 2

如右圖，5個磚塊可以堆的層數為2層。



### 輸入範例 3

2147483647

### 輸出範例 3

65535

## 問題 4 – 庭院深深深幾許 (deEEP)

---

(10 分)

### 問題敘述

- 「『庭院深深深幾許』同字不相鄰有幾種排法？」——全台灣的高中數學老師
- 「『山外青山樓外樓』同字不相鄰有幾種排法？」——雜牌的數學測驗卷

你受夠這種算出來毫無用處的數學習題了，決定寫一個程式來解決這類問題。為了處理方便，輸入句子中的每個字會被對應到大寫字母。

### 輸入格式

$N$   
 $S$

輸入的第一行包含一個正整數  $N$ ，代表字串的長度。  
第二行有一個字串  $S$ 。

### 輸出格式

輸出一個整數於一行，代表同字不相鄰的排列數。

### 資料範圍

- $1 \leq N \leq 8$
- $S_i \in \{A, B, C, \dots, Y, Z\}$
- 對於所有  $S_i$ ，字串  $S$  中必有  $A, B, C \dots S_i$ 。

### 資料範例

#### 輸入範例 1

5  
ABCDE

#### 輸出範例 1

120

#### 範例說明 1

五個相異字母 ABCDE 排列，共有  $5! = 5 \times 4 \times 3 \times 2 \times 1 = 120$  種排法。

因為第一個字母可以是 ABCDE 5 個中任一個；第二個字母不能和第一個相同，所以剩 4 種可以選；第三個字母不能和前兩個相同，所以剩 3 種可以選；第四個字母剩 2 種；第五個字母剩 1 種。逐項相乘便是全部可能的排列數。

## 輸入範例 2

4

AAAB

## 輸出範例 2

0

## 範例說明 2

四個相字母 AAAB 排列，無論是 AAAB、AABA、ABAA、BAAA 都會出現同字相鄰。沒有同字不相鄰的排法，故排列數為 0。

## 輸入範例 3

7

ABCCDE

## 輸出範例 3

240

## 問題 5 – Dependencies (dependencies)

---

(10 分)

### 問題敘述

`include` 指令在每個電腦語言大多都有，基本的一些函式別人都有了，就可以利用 `include` 的方式，拿來使用參考，而後續實作的物件自然就會因為都繼承於這些底層函式，而有交互參考的現象，也因此如果能將 `include` 的項目按照參考的關係，依序做排列，那麼這些參考的順序因為影響記憶體載入 / 釋放 / 參考，有時也能提升程式效能～

為了簡化問題的複雜情況（參考檔名稱、路徑...），我們假設檔案分別以 A-Z 代表，以 A-Z 的順序列出每個檔案的所有參考，並以逗號隔開，例如：`B,,A,A` 代表 A 檔案參考了 B 檔案，而 C 和 D 檔案參考了 A 檔案，所以因此也參考了 B 檔案，所以如果我們問 B 檔案，則可查出 ABCD 都使用到 B 檔案～也就是 ABCD 檔案都屬於 B 檔案的 dependencies。

由於 `include` 都有防止重覆載入避免重覆定義的情況，因此一般都支援交互參考的情況，也就是 `B,A` 是可以接受的。同理，如果 `B,A,A,A` 的話，一樣由 B 可查出 ABCD 都使用到 B 檔案～

### 輸入格式

輸入的第一行為一個長度不超過 700 的字串，包含 A-Z 各個檔案的所有參考。

接下來第二行為一個長度不超過 26 的英文大寫字串，每個字元代表要查 dependencies 的檔案。

### 輸出格式

依照第二行每個字元的順序，個別輸出每一個檔案參考到該字元檔案的檔案於一行（請按字典序輸出）。

### 資料範例

#### 輸入範例 1

```
B,A,BA,A,,Z,K,F,,,,,,,,,,,,,C,,,,,,Y,A,X  
ACKFP
```

#### 輸出範例 1

```
ABCDFHRXYZ  
CR  
GK  
FH  
P
```

## 解釋 1

A	B	C	D	E	F	G	H	I	J	K	L	M
B	A	BA	A		Z	K	F					
N	O	P	Q	R	S	T	U	V	W	X	Y	Z
				C						Y	A	X

第一格是 B，即 A include B 檔案；

第二格是 A，即 B include A 檔案；

第三格是 BA，即 C include B, A 兩個檔案，依此類推。

所以，

問所有參考到 A 檔案的檔案有 ABCDFHRXYZ

問所有參考到 C 檔案的檔案有 CR

問所有參考到 K 檔案的檔案有 GK

問所有參考到 F 檔案的檔案有 FH

問所有參考到 P 檔案的檔案有 P

## 輸入範例 2

IF, YOU, DO, NOT, KNOW, HOW, TO, SOLVE, THIS, QUESTION, PLEASE, TRY, TO, USE, D  
EPH, FIRST, SEARCH, WISH, YOU, HAVE, FUN, IN, THE, YTP, EXAM, OKAY  
WINYTPGAME

## 輸出範例 2

ABCDEFGHIJKLMNPOQRSTUVWXYZ  
ABCDEFGHIJKLMNPOQRSTUVWXYZ  
ABCDEFGHIJKLMNPOQRSTUVWXYZ  
ABCDEFGHIJKLMNPOQRSTUVWXYZ  
ABCDEFGHIJKLMNPOQRSTUVWXYZ  
ABCDEFGHIJKLMNPOQRSTUVWXYZ  
G  
ABCDEFGHIJKLMNPOQRSTUVWXYZ  
ABCDEFGHIJKLMNPOQRSTUVWXYZ  
ABCDEFGHIJKLMNPOQRSTUVWXYZ

### 輸入範例 3

DO, YOU, ENJOY, WRITE, PYTHON, C, CPLUS, OR, JAV, CODE,, IT, IS, A, EASY, WAY, TO, CHANGE, THE,  
WORLD, AND, MAKE, THE, WORLD, WONDERFUL, PLACE  
BOXKZ

### 輸出範例 3

B  
ABCDEFGHIJKLMNOPQRSTUVWXYZ  
X  
ABCDEFGHIJKLMNOPQRSTUVWXYZ  
Z

## 問題 6 – 翻煎餅 (pancake)

---

(10 分)

### 問題敘述

你即將舉辦一場派對，為了迎接客人，你決定拿出你的拿手好菜——煎餅！

為了準備，你拿出了一個很特殊的鐵盤，在這個鐵盤上做的所有的煎餅都會是排成一直線的。

就在你煎的很快樂時，你的常用鍋鏟突然不見了，眼見還有很多煎餅需要翻面（不然會焦掉），你迅速找到了一把造型奇特的鍋鏟，它的長度是  $k_i$ ，代表它只能一次為恰好連續的  $k_i$  個煎餅翻面！

焦急的你面臨這樣的問題，假設我們簡化煎餅的配置成一個字串  $s_i$ ，代表哪些煎餅是需要翻面的，那些是已經翻過面的，你可以迅速找到「使得所有煎餅都翻面」所需的最小次數嗎？

### 輸入格式

```
T
s1 k1
s2 k2
... ..
sT kT
```

輸入的第一行只有一個正整數  $T$ ，代表問題的次數。

接下來的  $T$  行，每行包含一個字串  $s_i$  以及一個正整數  $k_i$ ，分別代表該次問題煎餅的配置還有鍋鏟的長度，其中煎餅配置會用兩種字元「o」和「x」表示，「o」代表這個煎餅已經被翻過面了；「x」代表這個煎餅需要被翻面。

### 輸出格式

對於每個問題，輸出一個整數，代表『使得所有煎餅都翻面』所需的最小次數。若不可能達成，請輸出  $-1$ 。

### 資料範圍

- $1 \leq T \leq 1000000$
- $1 \leq k_i \leq 1000000$
- $1 \leq \sum |s_i| \leq 1000000$



## 資料範例

### 輸入範例 1

```
2  
OOO 2  
XOX 2
```

### 輸出範例 1

```
0  
2
```

### 解釋 1

範例的第一個問題，由於每個煎餅都翻過面了，所以你不需要再做任何動作；  
範例的第二個問題，我們可以先翻左邊兩塊煎餅，再翻右邊兩塊煎餅，便可以讓煎餅全部翻面。

### 輸入範例 2

```
1  
OXXX 4
```

### 輸出範例 2

```
-1
```

### 解釋 2

由於怎麼翻都只能讓全部的煎餅一起翻面，所以不可能讓所有的煎餅都翻面。

### 輸入範例 3

```
1  
OXOXOXOO 3
```

### 輸出範例 3

```
3
```

## 問題 7 – 第 K 名的逆襲 (kthrevenge)

---

(10 分)

### 問題敘述

你的老師有天對著全班說：「沒有人會記住第二名，大家只知道第一名」。你很不服氣，於是向老師發起了挑戰，你認為你可以記住第一名以外的人。

為了達成這個挑戰，你需要依序處理  $Q$  筆詢問：

第一類：將數值  $v$  加入到  $S$ 。

第二類：輸出  $S$  中第  $K$  小的值，並將它從  $S$  中刪除。特別地，如果  $S$  的大小未滿  $K$ ，輸出  $-1$  即可。

注意， $S$  中可以出現重複的元素。

### 輸入格式

$Q$   $K$   
 $query_1$   
 $query_2$   
... ..  
 $query_Q$

輸入的第一行包含兩個正整數，分別為  $Q$  和  $K$ 。

接下來有  $Q$  行輸入， $query_i$  代表第  $i$  筆詢問，每筆都屬於以下的一種：

第一種：

1  $v_i$

第二種：

2

其中  $v_i$  為一正整數，代表關於第  $i$  筆詢問所需要加入的數值（必為第一類詢問）。保證第二類詢問有至少一筆。

### 輸出格式

對於每一筆第二類詢問，輸出一行整數。

### 資料範圍

- $1 \leq Q \leq 10^5$
- $1 \leq K \leq 10^5$
- $1 \leq v_i \leq 10^{12}$

## 資料範例

### 輸入範例 1

```
3 2
1 7122
1 7111
2
```

### 輸出範例 1

```
7122
```

### 範例說明 1

經過兩次插入， $S = \{7111, 7122\}$ 。需輸出之第二小的值為 7122。

### 輸入範例 2

```
3 1
1 7122
2
2
```

### 輸出範例 2

```
7122
-1
```

### 範例說明 2

經過一次插入， $S = \{7122\}$ 。接著的詢問輸出第一小的值 7122 並從  $S$  刪除之。最後一個詢問時， $S$  裡不存在第一小的值，故輸出 -1。

### 輸入範例 3

```
5 2
1 7122
1 7122
1 71227122
2
2
```

### 輸出範例 3

```
7122
71227122
```

### 範例說明 3

經過三次插入， $S = \{7122, 7122, 71227122\}$ 。接著的詢問輸出第二小的值 7122 並從  $S$  刪除之。最後一次詢問時， $S = \{7122, 71227122\}$ ，此時第二小的值為 71227122。

## 問題 8 – 塔防遊戲 1 (towerdefense1)

---

(10 分)

### 問題敘述

塔防遊戲是遊戲的一種類型，進行方式通常是在一個場面上擺設各式各樣的塔，這些塔會自動去攻擊範圍內的敵人。當敵人全部被消滅，或者維持一段時間而沒有被敵人攻陷，就算成功。

現在小莫正在玩一個塔防遊戲，遊戲場景是一個 $6 \times 6$ 的正方形矩陣，裡面有一些障礙物。怪物有可能在任何不是障礙物的地方出現，玩家能做的事情就是在四面邊界設置弓箭塔，這些弓箭塔能夠攻擊所有範圍內出現的怪物，從塔前直到遇到障礙物為止的直線內都是塔的攻擊範圍。

小莫想要讓場面上所有的空地都能夠被至少一個弓箭塔兼顧，請問小莫至少需要在邊界設置幾座弓箭塔呢？

註：弓箭塔防衛的方向，只考慮水平或垂直方向，不考慮斜射（如範例 2 圖示）

### 輸入格式

共 6 行，每行 6 個字元，「.」代表空地，「#」代表障礙物。

### 輸出格式

請輸出一個整數於一行，代表小莫至少需設置幾座弓箭塔。  
如果不可能兼顧所有空地，請輸出「-1」。

### 資料範例

#### 輸入範例 1

```
.....  
.....  
.....  
.....  
.....  
.....
```

#### 輸出範例 1

6

#### 範例 1 說明

場面上沒有任何障礙物，四面邊界任選一個擺滿 6 個弓箭塔即可監視所有空地。

## 輸入範例 2

```

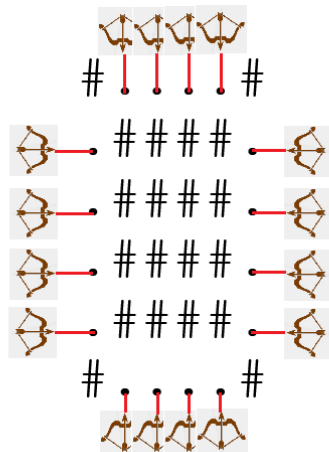
# . . . . #
. #####.
. #####.
. #####.
. #####.
. #####.
# . . . . #
  
```

## 輸出範例 2

16

### 範例 2 說明

場面上共有 16 個空地，上下左右各 4 個：  
 看守左邊的空地需要在左邊界設置 4 個弓箭塔，  
 看守上面的空地需要在上邊界設置 4 個弓箭塔，  
 右邊和下面以此類推，  
 總共需要 16 個弓箭塔。



## 輸入範例 3

```

. . . . .
. . . . .
. . # . .
. # . # .
. . # . .
. . . . .
  
```

## 輸出範例 3

-1

### 範例 3 說明

第 4 行第 3 列的空地被障礙物圍住，不管弓箭塔擺哪裡都無法攻擊到，因此無解。

## 問題 9 – 塔防遊戲 2 (towerdefense2)

---

(15 分)

### 問題敘述

塔防遊戲是遊戲的一種類型，進行方式通常是在一個場面上擺設各式各樣的塔，這些塔會自動去攻擊範圍內的敵人。當敵人全部被消滅，或者維持一段時間而沒有被敵人攻陷，就算成功。

現在小莫正在玩一個塔防遊戲，遊戲場景是一個  $r \times c$  的長方形矩陣，裡面有一些障礙物。怪物有可能在任何不是障礙物的地方出現，玩家能做的事情就是在四面邊界設置弓箭塔，這些弓箭塔能夠攻擊所有範圍內出現的怪物，從塔前直到遇到障礙物為止的直線內都是塔的攻擊範圍。

小莫想要讓場面上所有的空地都能夠被至少一個弓箭塔兼顧，請問有沒有可能辦到呢？

### 輸入格式

第 1 行有 2 個正整數  $r$  和  $c$ ，代表遊戲場面的大小。

第 2 行有 1 個正整數  $n$ ，代表障礙物的數量。

接下來有  $n$  行，第  $i + 2$  行有 2 個正整數  $r_i$  和  $c_i$ ，代表第  $i$  個障礙物的位置是第  $r_i$  行第  $c_i$  列。

### 輸出格式

如果存在一種弓箭塔的擺法使所有空地都被兼顧，請輸出一行「yes」。

否則，請輸出一行「no」。

### 資料範圍

$$1 \leq r, c \leq 1000000$$

$$0 \leq n \leq 5000$$

$$1 \leq r_i \leq r$$

$$1 \leq c_i \leq c$$

### 資料範例

#### 輸入範例 1

```
6 6
0
```

#### 輸出範例 1

```
yes
```

### 範例 1 說明

$6 \times 6$  的場地，沒有障礙物，在左邊界擺滿 6 個弓箭塔即可。

### 輸入範例 2

```

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

### 輸出範例 2

```
no
```

### 範例 2 說明

6 × 6的場地，(4,3)的空地被障碍物擋住，無法由上下左右任何一個方向攻擊，因此無解。

### 輸入範例 3

```

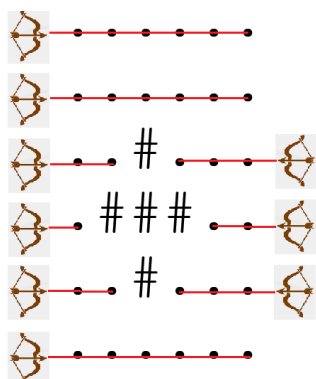
6 6
5
3 3
4 2
4 3
4 4
5 3
    
```

### 輸出範例 3

```
yes
```

### 範例 3 說明

6 × 6的場地，5 個障碍物在(4,3)處圍成實心的「+」字形，因此左邊界擺滿 6 個弓箭塔、右邊界擺 3 個弓箭塔即可監控所有空地。





## 問題 10 – 愛吃拉麵的小暘教授 (ramen)

---

(15 分)

### 問題敘述

小暘教授非常愛吃拉麵，大家都叫他拉麵小暘，並且他有一些神奇的體質，會使得他體內累積的拉麵量越多 IQ 越高。為了維持他無人能比的強度，小暘教授需要有足夠的錢負擔他的拉麵費用（其實是他私心不想錯過任何限定款的拉麵）。國立拉麵大學（簡稱 NRU，National Ramen University）得知了小暘教授的現況，也同時需要小暘教授的天才神腦，因此 NRU 主動提供了小暘教授一堆工作機會。

小暘教授雖然是教授，不過他不是超人，不能一次同時做兩個工作。但是小暘教授是強人，能夠在一份工作結束後馬上做另一份工作。嚴謹的說，將每一個工作的工作時間視為一個開區間，一份工作行事曆若是合法的，則工作表上任兩個工作的工作時間不得重疊。

每個工作有三個重要資訊，分別是開始時間、結束時間以及收益。現在你是小暘教授的學徒、秘書兼碼農，作為一個稱職的秘書，你需要幫他排一份合法的行事曆。

請幫幫小暘教授求出：在合法的工作行事曆下，所能賺取金錢的最大值。

### 輸入格式

第一行有一個正整數  $N$ ，代表小暘教授現在有多少份 NRU 工作機會。

接下來  $N$  行，每行有三個整數  $a_i, b_i, m_i$ ，分別代表第  $i$  份工作的開始時間、結束時間和總收益。

其中對所有的  $i$ ，保證  $a_i < b_i$ 。

### 輸出格式

只有一行，包含一個正整數  $M$ ，代表合法工作下所能賺到的金錢的最大值。

### 資料範圍

- $0 \leq N \leq 5 \cdot 10^4$
- 對所有  $i = 1, \dots, N, 0 \leq a_i, b_i \leq 5 \cdot 10^5$
- $0 < m_i < 2^{31}$

## 資料範例

### 輸入範例 1

```
5
0 5 1
5 7 2
7 9 3
0 6 4
7 10 10
```

### 輸出範例 1

```
14
```

### 範例解釋 1

選擇第 4 個和第 5 個工作，可以得到總金錢  $4+10=14$ 。

### 輸入範例 2

```
7
0 3 10
1 4 9
0 1 2
10 18 9
3 5 2
19 50 1
19 51 2
```

### 輸出範例 2

```
23
```

### 範例解釋 2

選擇第 1, 4, 5, 7 個工作，可以得到總金錢  $10+9+2+2=23$ 。

### 輸入範例 3

```
7
0 3 10
2 4 9
0 1 2
10 18 9
3 5 2
19 50 1
19 2000 3
```

### 輸出範例 3

```
24
```

### 範例解釋 3

選擇第 1, 4, 5, 7 個工作，可以得到總金錢  $10+9+2+3=24$ 。

## 問題 11 – 伏特衝擊 (lightning)

---

(15 分)

### 問題敘述

話說到當時，葛倫·雷達斯下定決心要在阿爾扎諾帝國魔術學院好好開始教課的那天。

「那今天就教教你們這個『伏特衝擊』好了。」

「基本的詠唱是，『雷精啊，以紫電的衝擊，擊倒敵人』」

只見葛倫的手中展開了簡易的紫色魔法陣，並發出了一道學生們感到都平凡無奇的紫色閃電筆直地衝向前方。

「伏特衝擊而已，我早就爐火純青了。」一名女學生不屑地說道。

「那問題來了，如果將詠唱改成『雷精啊，以紫電的，衝擊，擊倒敵人』，會發生什麼事？」

「那種咒文是不會正常啟動的，一定會以某種形式失敗。」另一名男學生在一旁諷刺。

「這我當然知道啊，我問的就是這失敗的形式到底是什麼。」

看著學生們個個一臉茫然，葛倫決定公布正確解答。

「怎麼了？都不知道？那好吧，正確答案是一一往右彎。」

葛倫複誦了一遍先前改造過的咒語，而顯現出來的結果完全與他說的吻合，瞬間台下便是一陣騷動。

「順帶一提，省略一部分的話呢，力道會大幅下降。」

「這些都做不到，就別說自己是爐火純青了。」

此時，在教室角落的小 Y 對於這個結果感到相當的興奮，他在回去後便針對「伏特衝擊」進行了研究，發現「伏特衝擊」的咒語組成會是由  $N$  個單字所組成序列的子序列，隨著調查，小 Y 發現了單字們的「權重函數」，根據「權重函數」做出相對應的子序列，咒文就可以在適當的斷句下詠唱成功！

於是他首先便將每個單字都計算出了他們的權重  $w_1, w_2, \dots, w_N$ ，並下了這樣的結論：

對於一個合法且長度是  $K$  的咒文，其使用的單字順序是  $a_1, a_2, \dots, a_K$ ，將滿足：

- $K \geq 2$  且  $K$  為偶數。
- $1 \leq a_1 < a_2 < \dots < a_K \leq N$ 。
- $w_{a_1} < w_{a_2}, w_{a_{K-1}} < w_{a_K}$ 。
- 如果  $i > 1$  且  $i$  是奇數，則  $w_{a_i} > w_{a_{i-2}}$  且  $w_{a_i} < w_{a_{i-1}}$ 。
- 如果  $i > 2$  且  $i$  是偶數，則  $w_{a_i} > w_{a_{i-2}}$  且  $w_{a_i} > w_{a_{i-1}}$ 。

簡單來說的話，就是咒文單字所形成的權重序列，連線後會形成閃電的樣子！

由於小 Y 沒有太多的時間窮舉所有可能的咒文序列，所以他希望找到最長合法的一個咒文之後再開始鑽研，你可以告訴他最長的咒文長度，好讓他確認他找到的咒文長度是最長的嗎？

## 輸入格式

$N$

$w_1 w \dots w_N$

輸入的第一行只有一個正整數  $N$ ，代表單字序列的長度。

接下來的一行有  $N$  個正整數  $w_1, w_2, \dots, w_N$ ，代表每個單字的權重大小。

## 輸出格式

輸出只有一行一個整數，代表小 Y 可以找到的最長咒文長度。如果不存在任何的咒文子序列，則輸出 0。

## 資料範圍

- $1 \leq N \leq 2000$
- $1 \leq w_i \leq 1000000000$

## 資料範例

### 輸入範例 1

5

1 3 5 2 4

### 輸出範例 1

4

### 解釋 1

1 3 2 4 是一個合法的最長咒文子序列，注意 1 5 2 4 並不合法，因為根據第五條規則，第四個數字要大於第二個數字。

### 輸入範例 2

5  
5 4 3 2 1

### 輸出範例 2

0

### 解釋 2

由於咒文的長度至少要是 2，所以不存在任何咒文子序列，輸出 0。

### 輸入範例 3

10  
1 1 3 3 5 5 2 2 4 4

### 輸出範例 3

4

## 問題 12 – 大字串 (bigstring)

---

(20 分)

### 問題敘述

現在你有一個長度為  $N$  的大字串  $S$ ， $S$  由小寫字母組成。

你的任務是要找到一個最小的數字  $M$ ，使得存在回文字串  $s_1, s_2, \dots, s_M$ ，滿足  $S = s_1 s_2 \dots s_M$ 。

一個字串  $s$  是回文若且唯若  $s_i = s_{|s|-i+1}$  對所有  $i = 1, 2, \dots, |s|$  都成立。

### 輸入格式

$N$   
 $S$

輸入的第一行包含一個正整數  $N$ ，代表大字串的長度。

第二行包含一個字串  $S$ ， $|S| = N$ 。

### 輸出格式

輸出一個整數於一行，代表題目敘述中的  $M$

### 資料範圍

- $1 \leq N \leq 2000000$
- 保證  $S$  由小寫字母組成。

### 資料範例

#### 輸入範例 1

5  
sssas

#### 輸出範例 1

2

#### 解釋 1

sssas = ss + sas

#### 輸入範例 2

10  
iqiiqqzzii

## 輸出範例 2

5

## 解釋 2

$iqi i q z z i i = i q i + i + q q + z z + i i$

## 輸入範例 3

15

ppffffgfdpdgffg

## 輸出範例 3

5

## 解釋 3

$ppffffgfdpdgffg = pp + fff + fgf + dpd + gffg$



## 問題 13 – 番薯運輸計畫 (yamtransporting)

(20 分)

### 問題敘述

在一個島上，有  $N$  個城市 (編號  $1, 2, \dots, N$ ) 和  $M$  條雙向道路，第  $i$  條道路的長度為  $W_i$ 。1 號城市的番薯生產過剩，身為島主的你決定發起一個「番薯運輸計畫 (Yam Transporting Project)」將番薯銷往  $N$  號城市。你想興建更多道路來縮短 1 號城市和  $N$  號城市的交通距離，並且和團隊研擬了  $K$  條可興建的雙向道路。但是礙於經費考量，你最多只能從中選一條興建。

請問在至多可以興建一條道路的情況下，從 1 號城市到  $N$  號城市的最短距離為何？

### 輸入格式

$N M K$

$x_1 y_1 w_1$

$x_2 y_2 w_2$

...

$x_M y_M w_M$

$x_{M+1} y_{M+1} w_{M+1}$

$x_{M+2} y_{M+2} w_{M+2}$

...

$x_{M+K} y_{M+K} w_{M+K}$

第一行包含三個整數  $N M K$ ，分別代表城市數量、道路數量、可興建道路數量。

接下來的  $M$  行，每行包含三個整數  $x_i y_i w_i$ 。  $x_i y_i$  表示道路兩端的城市，  $w_i$  表示道路長度。

接下來的  $K$  行，每行包含三個整數  $x_i y_i w_i$ 。  $x_i y_i$  表示可興建道路兩端的城市，  $w_i$  表示可興建道路長度。

### 輸出格式

請輸出一個整數於一行，代表 1 號城市到  $N$  號城市在最多興建一條道路下的最短距離。如果無論挑哪一條可興建道路都無法將番薯送到  $N$  號城市，請輸出  $-1$ 。

### 資料範圍

- $2 \leq N \leq 5 \times 10^5$
- $1 \leq M, K \leq 5 \times 10^5$
- $1 \leq w_i \leq 10^9$
- $1 \leq x_i, y_i \leq N$
- 保證任兩條道路或可興建道路不會連接同一對城市 (即對任意兩條道路或可興建道路  $i, j$ ，不發生  $x_i = x_j$  且  $y_i = y_j$  的狀況)
- 保證道路或可興建道路兩端城市不同 (即  $x_i \neq y_i$ )

## 資料範例

### 輸入範例 1

```
5 4 2
1 3 9
4 2 1
1 4 8
3 5 10
3 4 15
2 3 20
```

### 輸出範例 1

```
19
```

### 解釋 1

不論興建哪條道路，最短距離都是 19 (經過城市依序為 1 → 3 → 5)。

### 輸入範例 2

```
5 3 3
2 1 8
4 1 5
4 3 1000
5 3 1
4 2 20
1 3 2
```

### 輸出範例 2

```
1006
```

### 解釋 2

應選擇第一條可興建道路，使最短距離為 1006 (經過城市依序為 1 → 4 → 3 → 5)。

由於最多只能挑一條可興建道路，選擇第二條或第三條都會使 1 號城市無法將番薯送到  $N$  號城市。

### 輸入範例 3

```
5 3 2
1 2 10
4 2 20
1 4 30
3 4 40
5 3 50
```

## 輸出範例 3

-1

## 解釋 3

無論興建哪一條道路都無法將番薯送到  $N$  號城市，應輸出 -1。

## 問題 14 – 字串回復 (stringrecover)

(25 分)

### 問題敘述

有一個字串，字串中都沒有重複使用的字元。在一次意外中，它的字元順序被弄亂了。我們想利用子字串交換的方式，把這字元順序亂掉的字串回復到原始字串，最少要交換幾次才能完成？

所謂子字串交換，指的是可以從原字串中選擇任意兩段不重疊的子字串，互相將內容置換。例如：

**ABCDEF****GH** → **A****EF****G****DC****H** (ABCDEF $\overline{GH}$  將 BC 與 EFG 交換，成為 A**EF****G****DC****H**)

**3452187** → **1875234** (3452187 將 34 與 187 交換，成為 1875234)

**oday T** → **T oday** (oday**T** 將 oday 與 T 交換，成為 Today)

這些都是一次子字串交換的例子。

很明顯的 oday**T** 只需一次交換 (oday 和 **T**) 就可以回復成 Today。如果是要把 7654321 回復成 1234567 需多少次呢？

這有非常多種方法，但是最少都需要 3 次交換。這裡列幾個以 3 次交換完成回復的過程，作為參考(還有很多，不易列完)：

**7654321** → **3214765** → 1234**765** → **1234567** (765 與 321 交換成為 3214765，再將 3 與 1 交換成為 1234765，最後將 7 與 5 交換成為 1234567)

**7654321** → **5432167** → **3214567** → **1234567** (以下例子說明省略)

**7654321** → **1654327** → **1254367** → **1234567**

給你原始字串以及字元順序亂掉的字串，請寫支程式告訴我們最少要幾次子字串交換，才能回復成原來的字串。

### 輸入格式

1. 第一行：原始字串。
2. 第二行：一個正整數  $N$ ，表示後續有多少個字元順序亂掉的字串。
3. 第三行起有  $N$  行：每行一個字串，都是字元順序亂掉的字串。

## 輸出格式

1. 輸出共  $N$  行，每行包含一個整數，代表順序亂掉的字串各最少需幾次交換，才能回復成原始字串

## 資料範圍

1. 字串由英數字元 ('a'~'z', 'A'~'Z', '0'~'9') 所構成，大小寫視為不同。  
**字串中任何字元都不會重複出現**，字串長度不超過 60。
2.  $N$  不超過 10。

## 資料範例

### 輸入範例 1

```
1234567
2
7654321
5641237
```

### 輸出範例 1

```
3
1
```

### 範例 1 說明

本例有兩個字串 7654321 與 5641237 要回復成 1234567。  
7654321 至少需要 3 次交換（如問題敘述中所舉例）才能回復成 1234567，  
5641237 則只需要 1 次交換（56 與 123 交換）就能回復成 1234567，  
所以輸出兩行分別為 3 與 1。

### 輸入範例 2

```
B6C
3
6CB
B6C
C6B
```

### 輸出範例 2

```
1
0
1
```

## 範例 2 說明

本例有 3 個字串要回復成 B6C。  
6CB 需 1 次(將 6C 與 B 交換)，  
B6C 需 0 次(不需任何交換)，  
C6B 需 1 次(將 C 與 B 交換)，  
所以輸出 3 行，分別為 1、0、1。

## 輸入範例 3

```
abcdefghijkl  
1  
begjifdchka
```

## 輸出範例 3

```
4
```

## 範例 3 說明

本例只有一個字串 begjifdchka 需要回復成 abcdefghijk，這字串能以 4 次子字串交換完成回復(有許多方式，下方有幾個舉例)，但無法以 3 次交換完成回復，最少需要 4 次。因此輸出一行 4。

```
b[egj]ifd[ch]ka ->bc[hi]f[deg]jka ->bcde[g][f]hijka ->[bcdefghijk][a] ->abcdefghijkl  
[begjifdchk][a] ->abeg[j]i[fdch]k ->abe[g]f[dc]hijk ->ab[e]d[c]fghijk ->abcdefghijkl  
be[gji][f]dchka ->[befgjidchk][a] ->ab[efgji]d[c]hk ->abcdefg[j]i[h]k ->abcdefghijkl
```

## 輸入範例 4

```
vVYg8I00f73GqjNouR4FwD9C26mA5sKMezblThcpyri1kXn]StP  
7  
NThcp1kXqj4F26mCvtPMe3Gn]SwDYsI08yrblgOfuRiKzVo7A59  
Of73gvth1kR0yrncp6mA5siCz8IKMuqjN29G4wF]SDbVIYXePoT  
tKpkbziGq89FCo6mAyNg1vns]DS7004lThcjMRr5Vle3fPXYw2u  
3k6ysfhrIN4qc2TFP8nuwb5tARzi0JKvV1jY9SmDlpg7oXeGMCO  
v]5wjYDi3lhFfg8TkC26qXy1A4Iou9SKzmePVO7rnRpN0GsbtMc  
n]StPf73GqjNouR4Fw08ID9C26mA5sKMezblThcpyri1kXvVYgO  
r7PK0lTXR13vV4F6mNOqzc8sAtjhD9yeSpkGwgb2MoCYij5lufn
```

#### 輸出範例 4

12  
17  
21  
24  
21  
2  
22

#### 範例 4 說明

本例中 7 個字串各需要 12 次、17 次、21 次、24 次、21 次、2 次、22 次子字串交換才能回復成 `vVYg8I00f73GqjNouR4FwD9C26mA5sKMezblThcpyri1kXnJStP`，所以輸出 7 行，分別為 12、17、21、24、21、2、22。

## 問題 15 – 俠盜獵車手 (missioncomplete)

---

(25 分)

### 問題敘述

俠盜獵車手 (Grand Theft Auto) 是一款很有名的遊戲，它的「開放世界」是最大特色，舉凡開車、游泳、跑步、打網球、丟飛鏢、開船、開飛機、搶銀行、殺人放火，在俠盜獵車手的世界中通通辦得到。

此外，遊戲中也是有許多的任務等著玩家們挑戰。只要操控主角碰到位於指定地點的黃色圈圈，就可以接受並執行任務。

在任何時刻，你最多可以同時執行 4 個任務，每當一個任務完成後可以立即執行下一個任務。

今天，小莫正在玩俠盜獵車手，面對地圖中滿滿的任務地點，小莫不禁好奇，究竟至少要花多少時間才能將這些任務全部破關呢？

### 輸入格式

第一行有一個正整數  $n$ ，代表任務的數量。

第二行有  $n$  個整數  $s_1, s_2, \dots, s_n$ ，代表完成該任務所需要的時間。

### 輸出格式

請輸出一行，一個數字，代表完成全部任務最少需要的時間。

### 資料範圍

$$1 \leq n \leq 17$$

$$s_i \geq 1$$

$$\sum s_i \leq 2000$$

### 資料範例

#### 輸入範例 1

```
1  
2000
```

#### 輸出範例 1

```
2000
```

#### 範例 1 說明

只有 1 個任務，完成它需要 2000 的時間。



### 輸入範例 2

6  
1 3 2 6 5 4

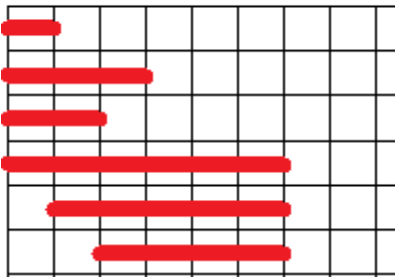
### 輸出範例 2

6

### 範例 2 說明

最佳解如下圖所示：

時間→



### 輸入範例 3

6  
4 3 2 4 4 4

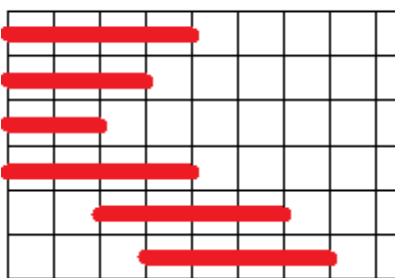
### 輸出範例 3

7

### 範例 3 說明

最佳解如下圖所示：

時間→



### 輸入範例 4

9  
1 2 3 4 5 6 7 8 9

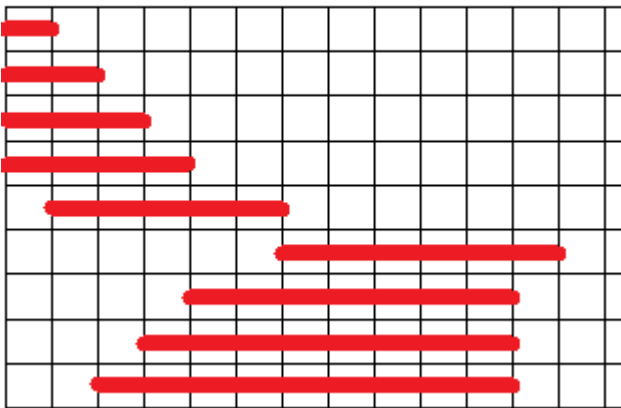
### 輸出範例 4

12

### 範例 4 說明

最佳解如下圖所示：

時間→



## 問題 16 – 洗街車路線問題 (streetcar)

(25 分)

### 問題敘述

在發財夢開始前，我們先認識一下故事的地理環境：小  $Y$  國裡面總共有  $N$  個加水站，而加水站之間則是由恰好  $N - 1$  個街道連接著。而且特別的是，不論你從哪一個加水站出發，都一定可以經過一些街道抵達任意一個加水站。

這一天，有個土豪在環遊全國的過程中口袋破洞了，讓每條街道都有一大堆錢幣。為此國內掀起了一股淘金熱，開洗街車這個職業一時間蔚為風潮。而你也搶到了一個開洗街車的職位，現在正在排班，你決定選擇一個能最大獲利的班次。因此需要計算出每個班次的最大收益，對於一趟班次，收益的定義如下：洗街車會由加水站  $S$  出發在不能經過重複街道的情況下抵達加水站  $T$ ，在路途中當洗街車位於任意的加水站時，你可以把洗街車上的吸塵器打開或關閉，當吸塵器的開關被打開時，洗街車可以獲得所有此時經過的街道上的價值。需要特別注意的是，在一趟班次裡面，你只能開啟吸塵器最多一次，而且由於有些路上的樹葉太多，因此街道的價值可能是正數、零或是負數。

現在已經知道了每條街道上的價值、了解每個班次的路線，請問對於每個班次，你可以獲得多少最大的獲利呢？

### 輸入格式

```
N Q
a1 b1 c1
a2 b2 c2
... ..
an-1 bn-1 cn-1
s1 t1
s2 t2
... ..
sQ tQ
```

測試資料第一行包含兩個正整數  $N, Q$ ，分別表示小  $Y$  國裡面的加水站數量以及可以排班的班次數量。

接下來的  $N - 1$  行，每一行有三個整數  $a_i, b_i, c_i$ ，代表加水站  $a_i$  和加水站  $b_i$  有一條價值為  $c_i$  的街道連接著。

接下來的  $Q$  行，每一行有兩個整數  $s_i, t_i$ ，分別代表著每趟班次的起點和終點的加水站編號。

## 輸出格式

輸出  $Q$  行，每行輸出一個整數。對於每個排班，請輸出一個數字代表可以得到的最大獲利。

## 資料範圍

- $1 \leq N, Q \leq 3 \times 10^5$
- $1 \leq a_i, b_i, s_i, t_i \leq N$
- $0 \leq |c_i| \leq 10^9$

## 資料範例

### 輸入範例 1

```
4 1
1 2 20
2 3 -10
3 4 20
1 4
```

### 輸出範例 1

```
30
```

### 範例說明 1

在第一個排班中，洗街車會依序經過加水站 1 -> 2 -> 3 -> 4，在 1 號加水站把吸塵器打開、在 4 號加水站把吸塵器關掉，總共可以得到總價值  $20 + (-10) + 20 = 30$ 。

### 輸入範例 2

```
1 1
1 1
```

### 輸出範例 2

```
0
```

### 範例說明 2

在第一個排班中，洗街車會依序經過加水站 1，在 1 號加水站把吸塵器打開、在 1 號加水站把吸塵器關掉，總共可以得到總價值 0。

### 輸入範例 3

```
6 2
1 2 -10
2 6 3
1 3 13
3 4 2
3 5 -7
4 6
5 6
```

### 輸出範例 3

```
15
13
```

### 範例說明 3

在第一個排班中，洗街車會依序經過加水站 4 -> 3 -> 1 -> 2 -> 6，在 4 號加水站把吸塵器打開、在 1 號加水站把吸塵器關掉，總共可以得到總價值  $2 + 13 = 15$ 。  
在第二個排班中，洗街車會依序經過加水站 5 -> 3 -> 1 -> 2 -> 6，在 3 號加水站把吸塵器打開、在 1 號加水站把吸塵器關掉，總共可以得到總價值 13。

## 問題 17 – 電熊的 $a+b$ 問題 (bear)

(25 分)

### 問題敘述

這世界上有一隻凌駕於人類智商的熊，牠就是電熊，平時會電電別人當有趣。如果不想被電，那就只能餵他吃蜂蜜，或是解出一道電熊出的  $a+b$  問題！

現在，你在路上遭遇了電熊，手邊沒有蜂蜜，但又不想被電的你，自然只剩解出題目這一條路了！電熊會給你若干個資訊，只要你能夠回答出  $a+b$ ，你就可以免於攻擊。這些資訊及變數的定義如下：

1.  $p, q$  為兩個質數，且  $p, q$  不相等。
2.  $n = pq$ ,  $g = n + 1$ ,  $\lambda = (p - 1)(q - 1)$ 。
3.  $a, b, r_a, r_b$  為四個整數。
4.  $c_a \equiv g^a r_a^n \pmod{n^2}$ ,  $c_b \equiv g^b r_b^n \pmod{n^2}$ 。

現在給你  $c_a c_b \pmod{n^2}$ ,  $\lambda, n$  這三個數字，你能夠告訴電熊  $a+b$  的最小可能值是多少嗎？

### 輸入格式

每個輸入只有一行，包含三個整數，分別是  $c_a c_b \pmod{n^2}$ ,  $\lambda, n$ 。  
這三個數字中間以空白隔開。

### 輸出格式

請輸出一行整數，也就是  $a+b$  的最小可能值為多少。

### 資料範圍

- $1 \leq p, q \leq 256$ ，其中  $p, q$  是質數。
- $1 \leq a + b < n$
- $1 \leq r_a, r_b < n$
- $1 \leq c_a c_b < n^2$

### 資料範例

#### 輸入範例 1

503160642 32760 33127

#### 輸出範例 1

3498

### 範例解釋 1

在這個範例中，電熊心中所想的 $a = 348$ ,  $b = 3150$ ,  $r_a = 12745$ ,  $r_b = 3536$ 。

所以答案是 $348 + 3150 = 3498$ 。

而 $c_a c_b$ 是透過下面式子得出來的結果：

$$(33127 + 1)^{348} (12745)^{33127} (33127 + 1)^{3150} (3536)^{33127} \equiv 503160642 \pmod{33127^2}$$

### 輸入範例 2

838515010 29548 29893

### 輸出範例 2

7414

### 範例解釋 2

在這個範例中，電熊心中所想的 $a = 7171$ ,  $b = 246$ ,  $r_a = 20971$ ,  $r_b = 8718$ 。

所以答案是 $7171 + 246 = 7414$ 。

而 $c_a c_b$ 是透過下面式子得出來的結果：

$$(29893 + 1)^{7171} (20971)^{29893} (29893 + 1)^{246} (8718)^{29893} \equiv 838515010 \pmod{29893^2}$$

### 輸入範例 3

144104393 39508 39913

### 輸出範例 3

32277

### 範例解釋 3

在這個範例中，電熊心中所想的 $a = 14836$ ,  $b = 17441$ ,  $r_a = 8125$ ,  $r_b = 38352$ 。

所以答案是 $14836 + 17441 = 32277$ 。

而 $c_a c_b$ 是透過下面式子得出來的結果：

$$(39913 + 1)^{14836} (8125)^{39913} (39913 + 1)^{17441} (38352)^{39913} \equiv 144104393 \pmod{39913^2}$$

## 符號意思以及可能有用的提示

- $a \equiv b \pmod{c}$  的意思是  $a \% c = b \% c$ ，其中 $\%$ 是取餘數的意思。例如  $5 \% 2 = 1$ 。  
歐拉定理和二項式定理可能會對你有幫助。

## 問題 18 – 數字來排隊 (sequence)

(25 分)

### 問題敘述

給定  $N$  個數字  $a_1, a_2, \dots, a_N$  及一個非負整數  $k$ ，我們將使用這  $N$  個數字來建立  $N$  個數列。一開始  $N$  個數列都是空的，我們用以下的方式依序建立每一個數列：

- 首先將  $a_i$  加入第  $i$  個數列。
- 依序考慮後面的所有數字  $a_j$  ( $j > i$ )，令當前這個數列（也就是第  $i$  個數列）的最後一個數字為  $y$ ，若  $|a_j - y| \leq k$ ，則我們將  $a_j$  放到這個數列的尾端。
- 重複上一個步驟直到考慮完所有  $a_j$  ( $j > i$ )，完成第  $i$  個數列的建立。

當建立完所有數列後，會有  $Q$  個詢問，每個詢問包含一個整數  $P$ 。若一個數列的最後一個數字不小於  $P$ ，我們稱這個數列為「 $P$  數列」。請問所有  $P$  數列內的所有數字和為多少？

### 輸入格式

$N$   $k$

$a_1$   $a_2$  ...  $a_N$

$Q$

$P_1$   $P_2$  ...  $P_Q$

輸入的第一行包含正整數  $N$ ，代表數字的數量；以及非負整數  $k$ ，意義見題目敘述。

輸入的第二行包含  $N$  個非負整數。

輸入的第三行為正整數  $Q$ ，代表詢問的數量。

輸入第四行為  $Q$  個非負整數  $P$ ，每個  $P$  代表一個詢問的內容。

### 輸出格式

輸出  $Q$  行整數，每一行依序對應到  $Q$  個詢問的答案。

### 資料範圍

- $1 \leq N \leq 2 \cdot 10^5$
- $0 \leq a_i \leq 2 \cdot 10^5$
- $0 \leq k \leq 10^9$
- $1 \leq Q \leq 2 \cdot 10^5$
- $0 \leq P_i \leq 10^9$

### 資料範例

#### 輸入範例 1

3 0

1 2 3

3

1 2 3



## 輸出範例 1

6  
5  
3

### 解釋 1

三個數列分別為：

- 1
- 2
- 3

在建立第一個數列時，我們先將 $a_1 = 1$ 放進數列裡。接著依序考慮 $a_1$ 後面的數字，因為 $|a_2 - a_1| = 1 > 0$ ，所以我們不將 $a_2$ 加入數列。接著 $|a_3 - a_1| = 2 > 0$ ，因此 $a_3$ 也不會被加入數列。第一個數列完成，只有 $a_1$ 這個數字。第二個與第三個數列使用同樣的方式建立。

## 輸入範例 2

5 1  
2 0 3 1 4  
3  
0 4 5

## 輸出範例 2

22  
20  
0

### 解釋 2

五個數列分別為：

- 2, 3, 4
- 0, 1
- 3, 4
- 1
- 4

在建立第一個數列時，我們先將 $a_1 = 2$ 放進數列裡。接著依序考慮 $a_1$ 後面的數字，因為 $|a_2 - a_1| = 2 > 1$ ，所以我們不將 $a_2$ 加入數列。接著 $|a_3 - a_1| = 1 \leq 1$ ，因此 $a_3$ 被加入數列。接著 $|a_4 - a_3| = 2 > 1$ ，因此 $a_4$ 不被加入，最後 $|a_5 - a_3| = 1 \leq 1$ ，我們將 $a_5$ 加入數列的後面，第一個數列完成。後續的數列使用同樣的方式建立。

### 輸入範例 3

```
2 10  
1 1  
2  
0 10
```

### 輸出範例 3

```
3  
0
```