

2017 網際網路程式設計全國大賽

國中組初賽

- 本次比賽共 6 題，含本封面共 16 頁。
- 全部題目的輸入都來自**標準輸入**。輸入中可能包含多組輸入，以題目敘述為主。
- 全部題目的輸出皆輸出到螢幕 (**標準輸出**)。
輸出和裁判的答案必須完全一致，英文字母大小寫不同或有多餘字元皆視為答題錯誤。
- 比賽中上傳之程式碼，使用 C 語言請用 `.c` 為副檔名；使用 C++ 語言則用 `.cpp` 為副檔名。
- 使用 `cin` 輸入速度遠慢於 `scanf` 輸入，若使用需自行承擔 Time Limit Exceeded 的風險。
- 部分題目有浮點數輸出，會採容許部分誤差的方式進行評測。一般來說「相對或絕對誤差小於 ϵ 皆視為正確」， ϵ 值以題目敘述為主。
舉例來說，假設 $\epsilon = 10^{-6}$ 且 a 是正確答案， b 是你的答案，如果符合 $\frac{|a-b|}{\max(|a|,|b|,1)} \leq 10^{-6}$ ，就會被評測程式視為正確。

	題目名稱	時間限制 (秒)
題目 A	瀚瀚得第一	1
題目 B	決戰賓果巔峰	1
題目 C	NPSC 魔法學院	1
題目 D	吃點心	6
題目 E	輪盤	1
題目 F	數數問題	1

2017 網際網路程式設計全國大賽

輸入輸出範例

C 程式範例：

```
1 #include <stdio.h>
2 int main()
3 {
4     int cases;
5     scanf("%d", &cases);
6     for (int i = 0; i < cases; ++i)
7     {
8         long long a, b;
9         scanf("%lld %lld", &a, &b);
10        printf("%lld\n", a + b);
11    }
12    return 0;
13 }
```

C++ 程式範例：

```
1 #include <iostream>
2 int main()
3 {
4     int cases;
5     std::cin >> cases;
6     for (int i = 0; i < cases; ++i)
7     {
8         long long a, b;
9         std::cin >> a >> b;
10        std::cout << a + b << std::endl;
11    }
12    return 0;
13 }
```

A. 瀚瀚得第一

Problem ID: rank

瀚瀚是一名天龍國認真的學生，他從小的目標就是在 NPSC 上得第一。不論是初賽還是決賽，只要是 NPSC 的比賽，瀚瀚都要拿第一！

而今天，剛好就是今年 NPSC 的初賽，瀚瀚也不意外的參加了。NPSC 初賽總共有六題，比賽的排名是以誰通過的題數越多，排名便排在比較前面，當通過的題數相同時，由較早達到該題數的人排在前面。

舉例來說，如果瀚瀚通過了五題，艾迪通過了三題，則當下的排名，瀚瀚無論如何都會在艾迪前面；而如果瀚瀚在第 100 分鐘時通過六題，艾迪在第 239 分鐘時才通過六題，則瀚瀚的排名會因為他先通過六題，所以排名會在艾迪前面。

現在，比賽恰好進行到一半，記分板上顯示著今年參加 NPSC 初賽的六個隊伍各自通過的題數以及排名，你想知道瀚瀚今年是否有機會得到第一。

只要有任何可能發生的情況使得瀚瀚得到第一，瀚瀚就是有機會得到第一！

Input

測試資料共有六行，第一行為瀚瀚目前所通過的題數以及排名，其餘五行為其他五隊目前所通過的題數以及排名。這六行，每行包含兩個整數 P_i, R_i ，分別代表通過的題數以及目前的排名。

- $0 \leq P_i \leq 6$
- $1 \leq R_i \leq 6$
- $R_i \neq R_j \forall i \neq j$ ，任兩隊排名不相等
- $R_i < R_j$ if $P_i > P_j$ ，通過題數較多的隊伍，排名一定會排在比較前面

Output

如果瀚瀚有機會得到第一，請輸出 "Yes" 於一行，否則請輸出 "No" 於一行，皆不包含引號。輸出後請記得換行。

Sample Input 1	Sample Output 1
6 1 6 2 6 3 6 4 6 5 6 6	Yes

Sample Input 2	Sample Output 2
1 6 1 5 1 4 1 3 1 2 1 1	Yes

B. 決戰賓果巔峰

Problem ID: bingo

段考結束了！蘿班善良的姜姜老師決定進行一場遊戲——歡樂分數大方送之決戰賓果巔峰。遊戲按照如下方式進行：

1. 蘿班每位學生先在紙上畫好 5×5 的方格
2. 蘿班每位學生在方格中填上 1 至 25 的數字，每種數字恰填入一次
3. 蘿班每位學生將填好數字的紙交給姜姜老師
4. 姜姜老師根據他的心情選擇一個 1 至 25 的排列 P 作為開獎順序
5. 對於每張紙，依照 P 中的順序一一進行標記，直到集滿一條線為止
6. 每標記一個數字，該張紙對應的學生段考分數就加一分（歡呼）

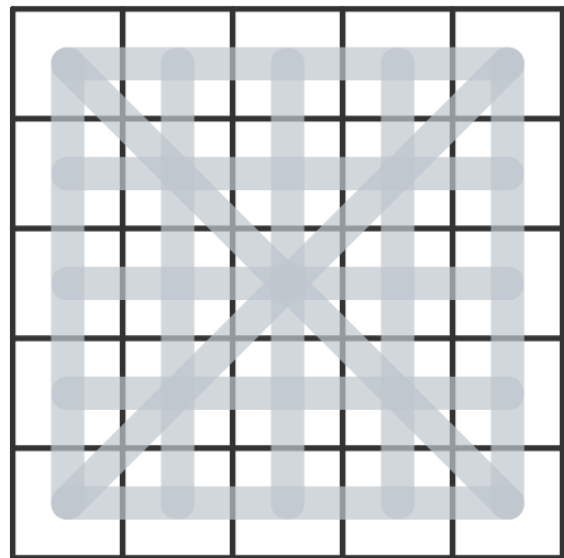
在本遊戲中一張紙共有十二條可能的線：五條橫線、五條直線、兩條對角線。

由於人工一張一張進行開獎實在太辛苦了，也容易有錯漏之疑慮。因此希望你能寫個開獎統計程式幫幫姜姜老師。

Input

測試資料第一行有一個整數 n ，代表蘿班共有幾位學生，依序編號為 $1, 2, \dots, n$ 。接下來 $5n$ 行，每五行為一組，依序代表每位學生紙上所填入的數字。最後一行包含 25 個數字，代表姜姜選擇的開獎順序 P 。

- $1 \leq n \leq 30$
- 為求版面美觀，紙上的數字以及開獎順序均補零至兩位數
- 1 至 25 在每位學生紙上以及開獎順序中均恰出現一次



Output

為了方便姜姜老師統計，請對於每個存在的加分分數，輸出有哪些學生符合條件。

輸出格式為："score: id id ... id" (不包含引號)，其中每個學生編號前恰有一個空白，且行尾無空白。請按照加分分數由小到大輸出，同樣加分分數的學生編號也請由小到大輸出。

Sample Input 1

```
3
01 02 03 04 05
06 07 08 09 10
11 12 13 14 15
16 17 18 19 20
21 22 23 24 25
01 06 11 16 21
02 07 12 17 22
03 08 13 18 23
04 09 14 19 24
05 10 15 20 25
15 20 19 10 17
01 12 18 23 04
06 08 25 22 16
24 21 07 11 02
14 05 09 03 13
11 14 15 05 08 10 18 22 07 20 21 19 02 25 12 03 13 09 17 24 16 04 01 23 06
```

Sample Output 1

```
14: 1 2
15: 3
```

C. NPSC 魔法學院

Problem ID: college

NPSC 魔法學院在今年正式成立囉！

埃迪身為 NPSC 魔法學院的第一任校長，希望能為魔法學院奠定許多良好的基礎。其中一項便是十分重要的班級數量問題！

在大魔導世界中，魔法學院第一屆的班級數量與學生人數的關係將會是決定校內魔力源是否穩定的關鍵。幸運的是，不久之前埃迪發現了一個神秘的數字 M ，如果第一屆的班級數恰滿足「每班人數皆不超過 M 人且至少有一個人時最小的班級數量」，那麼校內的魔力源將會比以往已知的方式穩定 2017 倍，這麼好的事情埃迪怎麼會不去追求呢？

由於埃迪目前已經寫魔法教科書寫得頭昏腦脹了，所以埃迪想請你幫助他。埃迪將告訴你第一屆招收的學生人數 N 以及他所發現的神秘數字 M ，你能幫他算出班級數量應該決定為多少才能擁有 2017 倍穩定的魔力源嗎？

Input

測試資料共有一行，包含兩個正整數 N, M ，分別代表第一屆魔法學院的學生人數以及埃迪發現的神秘數字。

- $1 \leq N \leq 2^{31} - 1$
- $1 \leq M \leq 2^{31} - 1$

Output

輸出共一行，包含一個正整數代表「班級數量應該決定為多少才能擁有 2017 倍穩定的魔力源」。

Sample Input 1	Sample Output 1
10 3	4

Sample Input 2

Sample Output 2

3 3	1
-----	---

D. 吃點心

Problem ID: snacks

眾所皆知，全國網際網路程式設計大賽（NPSC）的決賽會場會有很多好吃的點心，瑤瑤是一位喜歡吃點心的可愛國中生，一直很希望能夠吃到 NPSC 決賽的點心，夜夜為了獲得瑤瑤的歡心，便努力的練習程式並順利取得了晉級決賽的資格，但就在獲得資格的同時夜夜發現了瑤瑤吃點心有個習慣「如果有 N 個點心排在桌上排成一排，從左至右的種類分別是 A_1, A_2, \dots, A_N ，瑤瑤會從位置 L 開始一直往右吃，直到位置 R 為止，並且滿足每一種點心吃的數量都恰好是偶數個」，夜夜很好奇，如果已經知道了點心的配置方式，那瑤瑤有幾種可能的吃點心的方式呢？

舉例來說， $N = 5$ ，點心從左至右種類分別為 1, 1, 2, 2, 3，那麼瑤瑤可能吃的方式有以下三種：

1. 從第一個點心吃到第二個點心
2. 從第一個點心吃到第四個點心
3. 從第三個點心吃到第四個點心

Input

測試資料共有兩行，第一行包含一個正整數 N ，代表有 N 個點心放在桌上。第二行包含 N 個正整數 A_1, A_2, \dots, A_N ，代表從左至右桌上放的點心種類。

- $1 \leq N \leq 10^6$
- $1 \leq A_i \leq 10^6$

Output

輸出共一行，包含一個整數代表「瑤瑤有幾種可能的吃點心的方式」。

Sample Input 1	Sample Output 1
3 1 1 2	1

Sample Input 2	Sample Output 2
5 1 2 3 2 1	0

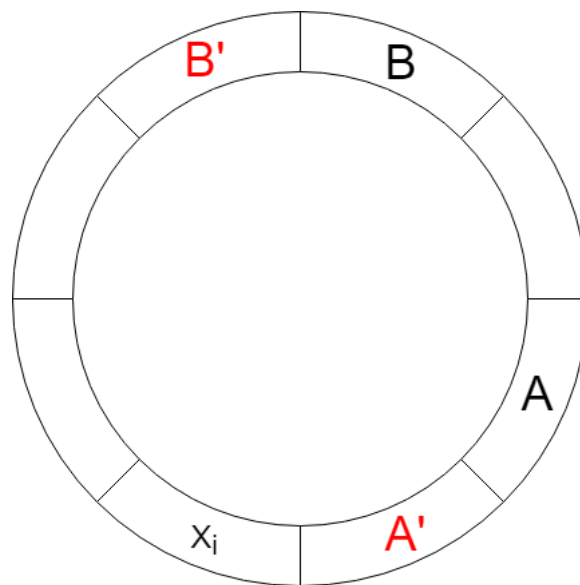
E. 輪盤

Problem ID: roulette

輪盤是一種賭場常見的博彩遊戲，一般會有 37 或 38 個數字，由莊荷負責在轉動的輪盤邊打珠，然後珠子落在該格的數字就是得獎號碼。

瀚瀚是一個可憐的小賭徒，賭輸了所有的家產。這讓瀚瀚重新開始思考人生，覺得為了追求刺激而賭博賭輸身家不太好。因此瀚瀚發明了一個新的輪盤遊戲，既可以跟朋友一起刺激的玩，還可以不用賭博。

一個輪盤有 N 個格子，我們順時鐘地以 1 到 N 來編號。一開始輪盤遊戲有兩個瀚瀚的朋友分別站在兩個**不同格子** A 與 B 上，接著瀚瀚會發出 M 個指令，每個指令會是一格子編號 X_i ，我們保證 X_i 不是 A 也不是 B ，接著瀚瀚的兩個朋友會一起**朝著 X_i 的方向前進一格**，其中一個會是逆時鐘方向，而另一個是順時鐘方向，如下圖所示。



其中 A 往 X_i 前進了一格，因此走到了 A' ，而 B 走到了 B' 。

瀚瀚保證他的朋友們都會遵守遊戲規則，且過程中 A, B 不會重疊在一起，即 A' 與 B' 不會相同。

現在瀚瀚想要記錄整個遊戲的過程，你能否幫忙記錄這兩個人位置呢？

Input

測試資料第一行，包含四個正整數 N, M, A, B ，代表輪盤有 N 個格子，瀚瀚一共發出了 M 個指令，一開始瀚瀚的兩個朋友分別在編號 A, B 的格子。

測試資料接下來包含 M 行，每行一個正整數 X_i ，代表第 i 次瀚瀚所發出的指令，測試資料保證不會有不合法的指令。

- $4 \leq N \leq 100000$
- $1 \leq M \leq 100000$
- $1 \leq A, B, X_i \leq N$
- 任何時候的指令前後都滿足 $A \neq B$
- 任何時候的指令 X_i 都不等於 A, B

Output

對於每個指令輸出兩個以空白隔開的正整數，代表當前瀚瀚兩個朋友的位置，其中第一個數字是 A 的新位置，第二個數字是 B 的新位置。

Note

瀚瀚的兩個朋友絕對不是左手與右手。

Sample Input 1	Sample Output 1
10 5 1 2	10 3
3	9 4
4	8 5
5	7 6
6	8 5
1	

Sample Input 2	Sample Output 2
4 6 1 4 2 1 2 1 2 1	2 3 1 4 2 3 1 4 2 3 1 4

This page is intentionally left blank.

F. 數數問題

Problem ID: hole

殿王是個天才兒童，他在一個月大的時候就已經學會數數了。

當時的他只要看到一個數字，就會迅速的數出這個數字當中有幾個「洞」。也就是把一個非負整數用十進制且前面沒有多餘的零的方法寫下來，計算有幾個區域被數字圍住。

舉例來說，8 有 2 個洞、1 有 0 個洞、6 有 1 個洞而 1234567890 這個非負整數則有 6 個洞（因為 4, 6, 9, 0 各有 1 個洞而 8 有 2 個洞）。

現在瀚瀚被請來當殿王的家教老師，他想試試看殿王能不能順利計算出 x 這個數字，所以打算找出一個恰有 x 個洞的非負整數。

不過這樣的數字可能有很多個，例如當 x 為 4 時，就會有 514514514514, 9600, 88, 10000 等許多個數字可以寫。不過瀚瀚有點懶惰，所以想要寫下這些數字當中最小的一個，也就是 88。

現在瀚瀚請你寫個程式來幫他完成這個出題給殿王練習的任務。

Input

測試資料只有一行，當中包含一個整數 x ，表示瀚瀚想要產生有 x 個洞的數字。

- $0 \leq x \leq 514$

Output

輸出一行，當中包含一個最小且恰有 x 個洞的非負整數 K 。

Sample Input 1	Sample Output 1
4	88

Sample Input 2**Sample Output 2**

1	0
---	---

Sample Input 3**Sample Output 3**

3	48
---	----