

2022 網際網路程式設計全國大賽

高中組模擬測試賽

- 本次比賽共 7 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。
- 任何題目內提到的「一行」，皆代表以換行字元「`\n`」結尾的字串。
- 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「相對或絕對誤差不超過 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。

Problem	Problem Name	Time Limit	Memory Limit
A	北極熊大遷徙	1 s	1024 MB
B	北極熊大遷徙研究	1 s	1024 MB
C	猜數字	1 s	1024 MB
D	南極企鵝大遷徙	1 s	1024 MB
E	小 P 打仗	2 s	1024 MB
F	圈圈叉叉	2 s	1024 MB
G	殿王的飲料王國	1 s	1024 MB

2022 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. 北極熊大遷徙

Problem ID: polarbear

因為全球暖化的關係，北極各處的浮冰正在慢慢融化之中。部份北極熊所在的浮冰已經融化到不堪居住的程度，於是這些北極熊興起遷徙的念頭。

已經融化到不堪居住的浮冰 A 上有 a 隻北極熊，牠們現在打算遷徙到有 b 隻北極熊居住的浮冰 B 。你要回答的是：經過北極熊大遷徙以後，浮冰 B 上總共會有多少隻北極熊。

Input

輸入只有一行，有兩個整數 a 和 b ，代表有 a 隻北極熊即將從浮冰 A 遷徙到原本有 b 隻北極熊的浮冰 B 。

- $1 \leq a, b < 2^{31}$

Output

輸出一行，表示浮冰 B 上最後會有多少隻北極熊。

Sample Input 1

24 47

Sample Output 1

71

Sample Input 2

33 20

Sample Output 2

53

This page is intentionally left blank.

B. 北極熊大遷徙研究

Problem ID: polarbear2

因為全球暖化的關係，北極各處的浮冰正在慢慢融化之中。部份北極熊所在的浮冰已經融化到不堪居住的程度，於是這些北極熊興起遷徙的念頭。

已經融化到不堪居住的浮冰 A 上有 a 隻北極熊，牠們曾經遷徙到有 b 隻北極熊居住的浮冰 B 。你是個學者，你正在研究北極熊的遷徙狀態。已知目前浮冰上已有 x 隻北極熊，你想知道在遷徙時有多少外來的北極熊 a 跟原生的北極熊 b 。

你要回答的是：經過北極熊大遷徙之前，浮冰 A, B 上可能分別會有多少隻北極熊，需要一個可能的答案，但你也知道以前的北極熊族群不會太大，不會超過 1000 隻。

Input

輸入只有一行，只有一個整數 x ，表示你要研究的浮冰上有 x 隻北極熊。

- $0 \leq x \leq 2000$

Output

輸出一行，有兩個整數 a, b 並以一個空白隔開，分別表示浮冰 A, B 原有可能會有多少隻北極熊。

- $x = a + b$
- $0 \leq a, b \leq 1000$

Sample Input 1	Sample Output 1
4	3 1

Sample Input 2

5

Sample Output 2

4 1

Sample Input 3

14

Sample Output 3

5 9

C. 猜數字

Problem ID: guess

我在心中想了一個介於 1 到 1000 的整數，你有辦法猜到這個數字是多少嗎？每當你猜了一個數字，我可以告訴你猜的過低、過高或正確。但你最多只能猜 10 次，所以你要好好選擇你猜的數字。

互動說明

當你的程式打算要猜數字時，輸出一行且包含一個整數，這個整數必須介於 1 到 1000 之間。當你猜完數字後，記得要清空 (flush) 標準輸出 (standard out)。

當我們收到你的猜測後，會把你猜的結果回覆到你的標準輸入 (standard in)。回覆會是下列三種：

- “lower” 如果我想的數字比你猜的數字小
- “higher” 如果我想的數字比你猜的數字大
- “correct” 如果你猜到了

當你猜到了正確數字後，你的程式必須立刻結束 (exit)。如果你 10 次都猜錯了，你的程式將會被強制中止。

以下是 C 程式 flush 的範例：

```
1 #include <stdio.h>
2 int main()
3 {
4     printf( "500\n" );
5     fflush( stdout );
6 }
```

以下是 C++ 程式 flush 的範例：

```
1 #include <iostream>
2 int main()
3 {
4     std::cout << "500\n";
5     std::cout << std::flush;
6 }
```


D. 南極企鵝大遷徙

Problem ID: penguin

因為全球暖化的關係，南極各處的浮冰正在慢慢融化之中。部份企鵝居住地的浮冰已經大量融化，導致他們重要的食物來源「磷蝦」數量銳減，已到不堪居住的程度。已經融化到不堪居住的浮冰 A 上有 a 公斤的企鵝，牠們現在打算遷徙到有 b 公斤的企鵝居住的浮冰 B 。

你要回答的是：經過企鵝大遷徙以後，浮冰 B 上總共會有多少公斤的企鵝。

Input

輸入只有一行，有兩個浮點數 a 和 b ，代表有 a 公斤的企鵝即將從浮冰 A 遷徙到原本有 b 公斤重的企鵝的浮冰 B 。

- $0 \leq a, b \leq 50$
- a, b 的小數點後最多有五位。

Output

輸出一行，表示浮冰 B 上最後會有多少公斤的企鵝。

如果你的答案的絕對或相對誤差不超過 10^{-6} 都會被當作正確。

Sample Input 1	Sample Output 1
24.23 47.33	71.56
Sample Input 2	Sample Output 2
24.23000 47.33000	71.56000000000000
Sample Input 3	Sample Output 3
24.230 47.330	71.560001

This page is intentionally left blank.

E. 小 P 打仗

Problem ID: war

在小 P 國裡，有 N 個士兵隨時防守著城牆。由於疫情險峻，所以小 P 決定要替每位士兵施打疫苗。已知第 i 位士兵的防禦力為 s_i ，且在施打疫苗後，必須休息 r_i 天。也就是說，如果第 i 位士兵在第 D 天施打疫苗，那在第 $D, D+1, \dots, D+r_i-1$ 天，都必須休息。

讓我們定義 $w(i)$ 為所有在第 i 天不必休息的士兵的防禦力和。

為了避免新冠肺炎擴散，小 P 在接下來的 N 天中，每天必須**恰好為一位**士兵施打疫苗，且每位士兵都必須被施打**恰一劑**疫苗。

因為敵人每天都有可能攻打過來，所以小 P 想問你，如果敵人在第 k 天攻打過來，那 $w(k)$ 最大可以是多少？

請對於 $k = 1, 2, 3, \dots, N$ 都輸出一個整數代表答案。注意對於每個 k ，你可以獨立的決定對於這個 k 的施打順序。

Input

輸入的第一行有一個正整數 N ，代表有幾名士兵。

第二行有 N 個以空格分開的整數 s_1, s_2, \dots, s_N ，代表第 i 個士兵的防禦力為 s_i 。第三行有 N 個以空格分開的整數 r_1, r_2, \dots, r_N ，代表第 i 個士兵打完疫苗要休息 r_i 天。

- $1 \leq N \leq 3 \times 10^5$
- $1 \leq s_i \leq 10^9$
- $1 \leq r_i \leq N$

Output

請輸出 N 個以空白隔開的整數，第 i 個整數代表最大的 $w(i)$ 。請注意不要輸出多餘的行尾空白。

Sample Input 1

5 1 1 1 1 1 1 5 5 4 4	Sample Output 1 4 4 3 2 2
-----------------------------	-------------------------------------

Sample Input 2

10 20 9 1 38 2 1 3 4 8 23 1 3 3 2 9 9 8 9 10 10	Sample Output 2 108 108 108 108 108 106 103 99 94 75
---	--

F. 圈圈叉叉

Problem ID: tictactoe

你有玩過圈圈叉叉嗎？圈圈叉叉，又稱井字棋，是一個在大小 3×3 的棋盤上玩的雙人遊戲。兩個玩家一人代表圈、一人代表叉，並且輪流在棋盤空的格子中放上自己的符號，而誰先連成一條線就贏了。

小 Y 是一個天才兒童，所以傳統 3×3 的圈圈叉叉對他來說已經太無趣了。因此，他發明了一種新的版本，並邀請他的好朋友小 P 一起來玩。小 Y 發明的版本是在一個 $N \times N$ 的棋盤上玩，與傳統的規則一樣，一人畫圈、另一人畫叉，並且拿圈的玩家先手。但與傳統的圈圈叉叉規則不一樣的是，小 Y 的遊戲比的不是誰先連成一條線，而是兩人會下到不能再下為止（也就是棋盤已經完全被佔滿），再看誰的連線比較多，一個玩家的得分就是他的連線數量，而很自然的，得分較高的玩家就勝出。

補充說明，在 $N \times N$ 的棋盤上，連線是指有同排、同列、或是同對角線的 N 個相同符號。

小 Y 跟小 P 都覺得新版的圈圈叉叉實在比舊版的好玩太多了，但他們馬上就遇到了一個問題：由於棋盤實在是太大了，遊戲結束後實在無法慢慢的用人力來算分。當然，寫個程式來處理這個問題對於天才兒童小 Y 來說可以說是易如反掌，但可想而知，他還有更重要的事情要做。因此，身為程式競賽參賽者的你們，何不幫幫小 Y 解決這個麻煩的問題呢？

Input

輸入第一行有一個正整數 N ，代表圈圈叉叉棋盤的大小。

接著 N 行，每行有一個長度為 N 的字串，代表圈圈叉叉的盤面。保證字串只有 O、X 兩種字元，分別代表圈圈以及叉叉。

- $3 \leq N \leq 2000$

Output

若輸入的盤面不可能為一個合法的結束盤面，輸出 Impossible。否則，輸出兩個以空白隔開的整數，分別代表先手玩家的得分以及後手玩家的得分。

Sample Input 1	Sample Output 1
3 000 XXX OXO	1 1

Sample Input 2	Sample Output 2
3 000 000 000	Impossible

Sample Input 3	Sample Output 3
4 OXXO XOOX XOOX OXXO	2 0

G. 殿王的飲料王國

Problem ID: drinks

殿王是個天才兒童，他在一個月大的時候就學會數數、六個月大的時候就學會乘法跟除法、一歲時學會寫程式、一歲又六個月時養了可愛的拉布拉多、一歲又十個月時養了可愛的貓咪、兩歲時發明了「吃餅乾」的遊戲、三歲又三個月大時成功地對貓咪做了排序。現在要講的是殿王三歲又四個月的故事。

殿王在三歲又四個月的時候創立了一個飲料王國，這個王國一共有 N 種面額的貨幣，面額大小分別為 $1, 2, \dots, N$ 元。某一天，王國中的 $N \times M$ 位小朋友想一起去飲料王國中最有名的飲料店「億園」買飲料，「億園」之所以有名是因為他販售了史上最好喝的飲料「QQ捏捏好喝到殿王茶」，而且只需要花 1 元就能購買。這 $N \times M$ 位小朋友中，有 M 位恰有一張面額為 1 元的貨幣、 M 位小朋友恰有一張面額為 2 元的貨幣、 \dots 、 M 位小朋友恰有一張面額為 N 元的貨幣。這 $N \times M$ 位小朋友都想去買「QQ捏捏好喝到殿王茶」，然而因為人數實在太多，殿王只好將小朋友們分成紅白兩隊，讓紅隊的人先去買飲料，等紅隊的所有人買完再讓白隊的人去買飲料。已知殿王在幫小朋友們分完隊伍後，紅白兩隊當中，分別有 a_i, b_i 位小朋友持有 i 元貨幣。

身為「億園」飲料店唯一店員的你，每次只能接受一張訂單，將手上唯一一張訂單處理完後才能受理下一位客人點餐。然而店面開張當天，你發現了一個大問題，那就是收銀機完全沒有任何的零錢，也就是沒有辦法找零。無奈之下你只能使用客人們支付的零錢進行找零，於是你希望小朋友們能以「完美的」順序排隊，使得每一位小朋友在點餐時，你都有足夠的零錢可以支付找零，也就是說，如果一位小朋友支付了面額為 x 元的貨幣，你必須能夠恰好找給他 $x - 1$ 元。

好奇心旺盛的你，想知道小朋友們有幾種「完美的」排隊順序可以讓你順利的為每位小朋友進行找零。在兩種排隊順序中，只要存在正整數 k 使得你在這兩個排隊順序中服務的第 k 位小朋友是不同人，那它們就是不相同的排隊順序。因為答案可能很大，所以只需要計算其除以 998244353 之後的餘數就可以了。

Input

輸入第一行包含兩個正整數 N, M 。

接下來共有 N 行，其中第 i 行包含兩個整數 a_i, b_i ，分別代表紅隊與白隊中持有 i 元貨幣的小朋友人數。

- $1 \leq N \times M \leq 10^6$
- $0 \leq a_i, b_i \leq M$
- $a_i + b_i = M$

Output

請輸出一個整數，代表一共有幾種「完美的」排隊順序除以 998244353 後的餘數。

Notes

在 Sample Input 1 中，假設 x_1, x_2 是兩位恰有一張面額為 1 元的小朋友， y_1, y_2 是兩位恰有一張面額為 2 元的小朋友，並且 $\{x_1, x_2, y_1\}$ 隸屬於紅隊， $\{y_2\}$ 隸屬於白隊，那麼所有完美的排隊順序如下：

- x_1, x_2, y_1, y_2
- x_2, x_1, y_1, y_2
- x_1, y_1, x_2, y_2
- x_2, y_1, x_1, y_2

Sample Input 1	Sample Output 1
2 2 2 0 1 1	4

Sample Input 2	Sample Output 2
3 2 2 0 2 0 1 1	20