

2022 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 20 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	巫醫巫醫畫畫畫畫	1 s	1024 MB
B	猜排列	1 s	1024 MB
C	NPSC	1 s	1024 MB
D	數數挑戰	1 s	1024 MB
E	比賽轉播	1 s	1024 MB
F	七對子	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. 巫醫巫醫畫畫畫畫

Problem ID: paint

巫醫巫醫是個傳奇人物，聽說他遇到很困難的問題的時候，都只要喊一聲咒語巫醫巫醫殿殿，問題就會被解決！於是，小 Y 給了他一個挑戰。

有一個 $N \times N$ 的地圖，上面充滿了危險，在這個地圖上面有四種土地：

1. 劇毒之地，上面充滿了恐怖的毒氣，任何人只要一走到這塊地就會立刻死掉。
2. 人人之地，上面有一個無辜的人，想要趕快回家。
3. 超級傳送門，可憐的人只要走上來就會被傳送出這個危險的地圖。
4. 安全之地，原本是劇毒之地，經過巫醫巫醫的法術之後，變成可以行走的地方。

而這個地圖是四連通的，也就是說無辜的人可以從座標 (i, j) 走到以下四個座標：

1. $(i + 1, j)$
2. $(i - 1, j)$
3. $(i, j + 1)$
4. $(i, j - 1)$

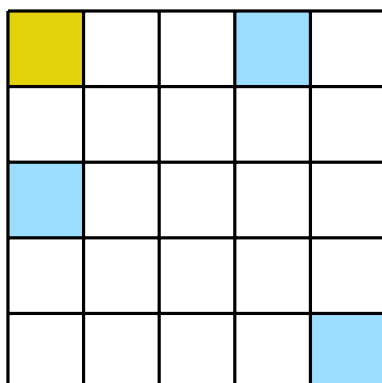
而在地圖之外，全部都是可怕的無底洞，所以你只要走到一個超出範圍的座標 (x, y) 滿足 $x \notin [1, N]$ 或 $y \notin [1, N]$ 那你就會立刻掉下去摔死。註： $n \notin [1, N]$ 表示 $n < 1$ 或 $n > N$ 。

為了拯救無辜的人，巫醫巫醫必須施展法術，每次施展法術都只能把一個劇毒之地的毒氣消除，變成安全之地，讓人可以安全的走在這塊地上。巫醫巫醫希望在施展法術過後，每個無辜的人，都可以不經過劇毒之地，走到超級傳送門。

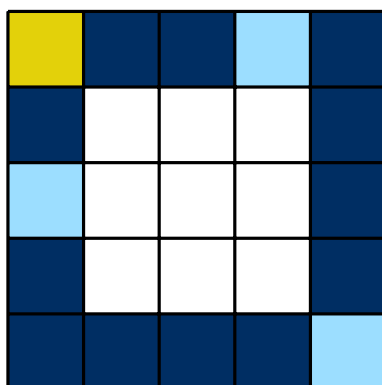
但是巫醫巫醫的法術能量有限，他最多只能施展 $\lfloor \frac{N \times N}{2} \rfloor$ 次法術，請你幫他決定好要在哪些地方施法術，才能夠讓所有無辜的人都能夠安全抵達超級傳送門。

以下為範測說明，深藍的格子是劇毒之地，天藍色的格子是人人之地，黃色的格子是超級傳送門，紫色的格子是安全之地。

下圖是範測 1 原本的樣子



下圖是範測 1 施展過法術的樣子



Input

輸入的第一行有一個正整數 N 。

接下來 N 行，每行都有一個長度為 N 的字串 S_i ，其中 $S_{i,j}$ 代表地圖上座標 (i,j) 的土地類型。

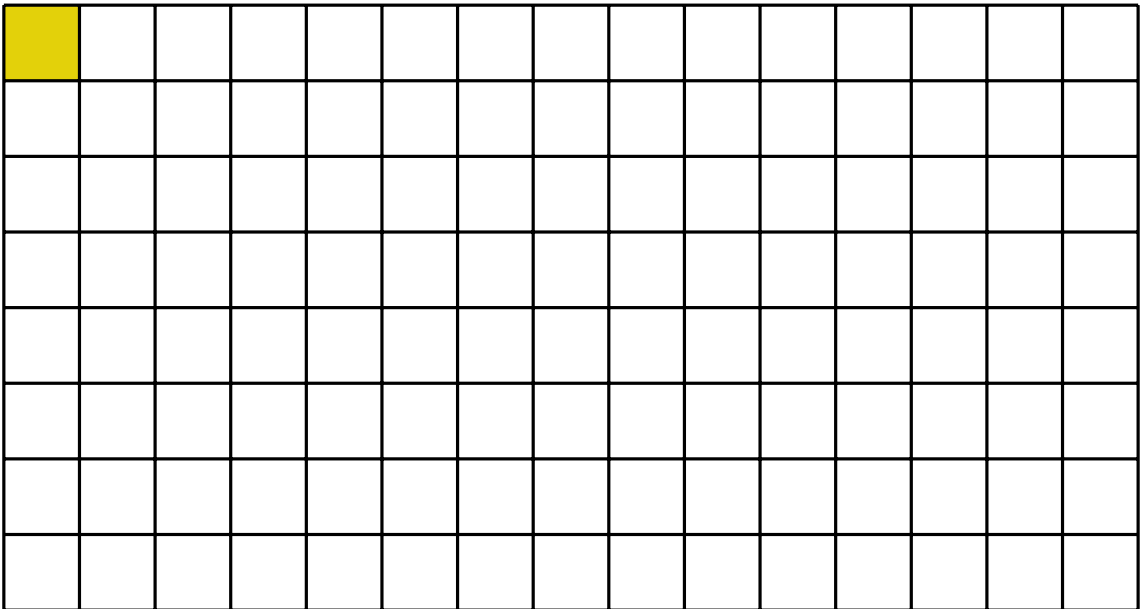
- $1 \leq N \leq 4 \times 10^3$
- $S_{i,j} \in \{., P, X\}$ ，其中 . P X 分別代表劇毒之地、人人之地和超級傳送門
- 保證 $S_{1,1} = X$

Output

請輸出 N 行，每一行一個長度為 N 的字串 S'_i 代表施展過法術之後的地圖。字元 0 代表有施展法術的土地，其他規則和輸入一樣。如果你的地圖上面有超過 $\lfloor \frac{N \times N}{2} \rfloor$ 個土地被施展法術、有土地被不合法的施展法術、或是有一個未被施展法術的土地與原本不同，那你會得到 Wrong Answer 的結果。注意到如果有多組解，任意一組都會得到 Accepted。

Hint

你可以利用以下的東西幫助思考。



Sample Input 1	Sample Output 1
5 X..P. P.....P	X00P0 0...0 P...0 0...0 0000P

This page is intentionally left blank.

B. 猜排列

Problem ID: guessperm

我在心中想了一個 1 到 N 的排列 p_1, p_2, \dots, p_N ，你有辦法猜到這個排列是多少嗎？每當你猜了一個序列，我可以告訴你猜的序列是不是這個排列的子序列。但你最多只能猜 $\lceil N \log_2 N \rceil + 1$ 次，所以你要好好選擇你猜的序列。

排列

「1 到 N 的排列 p_1, p_2, \dots, p_N 」是指 p_1, p_2, \dots, p_N 兩兩互不相同，1 到 N 每個數字都恰好在 p_i 出現一次。

子序列

我們說一個序列 a_1, a_2, \dots, a_k 是排列 p_1, p_2, \dots, p_N 的子序列，若且唯若存在 $1 \leq i_1 < i_2 < \dots < i_k \leq N$ 使得 $a_j = p_{i_j}$ 。例如，排列 $[2, 4, 1, 3, 5]$ 的子序列有 $[2]$ 、 $[2, 3]$ 、 $[4, 3, 5]$ 、 $[2, 4, 1, 5]$ 、 $[2, 1, 3]$ 等等，而 $[3, 3, 3]$ 、 $[1, 3, 2, 1, 1]$ 、 $[3, 4, 5]$ 、 $[2, 1, 5, 4]$ 、 $[1, 3, 3]$ 、 $[4, 3, 1]$ 等等則不是 $[2, 4, 1, 3, 5]$ 的子序列。

互動說明

你的標準輸入第一行會有一個正整數 N 代表我想的排列的長度。

當你的程式打算要猜一個長度 K 的序列時，輸出一行且包含 $K + 1$ 個整數，以空白隔開。第一個整數是 K 代表你要猜的序列的長度，接下來 K 個整數 a_1, a_2, \dots, a_K 是你猜的序列的內容。你的猜測必須符合 $1 \leq K \leq N$ 且 $\forall 1 \leq i \leq K, 1 \leq a_i \leq N$ ，否則你可能會得到 Wrong Answer。當你猜完序列後，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：

- “Nie” 如果你猜的序列不是任何是我想的排列的子序列

- “Tak” 如果你猜的序列是我想的排列的子序列
- “Gotowe” 如果你猜到了

當你猜到了正確排列後，你的程式必須立刻結束 (exit)。如果你在限制次數內都沒有猜對，你的程式將會被強制中止。

- $1 \leq N \leq 500$

以下是 C++ 程式 flush 的範例：

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

以下是 C 程式 flush 的範例：

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

你可以用以下的程式碼來得到每個 N 相對應的次數限制。

```
1 #include <iostream>
2 #include <cmath>
3 int main() {
4     int N;
5     std::cin >> N;
6     int X = ceil(N * log2l(N)) + 1;
7     std::cout << X << std::endl;
8 }
```

```
1 #include <stdio.h>
2 #include <math.h>
3 int main() {
4     int N;
5     scanf("%d", &N);
6     int X = ceil(N * log2l(N)) + 1;
7     printf("%d\n", X);
8 }
```

Sample Input 1	Sample Output 1
5	
Nie	3 1 2 3
Tak	3 2 1 3
Nie	3 3 3 3
Nie	5 1 3 2 1 1
Tak	3 4 3 5
Gotowe	5 2 4 1 3 5

範例測資說明：左側表示評測系統的輸出（你可以用標準輸入 (standard in) 讀入），右側則代表你一種合法的詢問。在範例測資中，我心裡所想的祕密排列是 $[2, 4, 1, 3, 5]$ 。評測系統首先輸出 5 表示這個排列的長度是 5。接著你依序向評測系統猜了 $[1, 2, 3]$ 、 $[2, 1, 3]$ 、 $[3, 3, 3]$ 、 $[1, 3, 2, 1, 1]$ 、 $[4, 3, 5]$ ，並依序得到 Nie、Tak、Nie、Nie、Tak 的回覆。最後，花了一次猜測來猜中正確的排列，即 $[2, 4, 1, 3, 5]$ 。

在你得到 Gotowe 的回覆，即你得知你猜中正確答案後，你的程式必須立即結束。

補充來說，你至少會需要花費一次 $K = N$ 的猜測來猜出正確答案。

C. NPSC

Problem ID: npsc

NPSC 國經常舉辦各式各樣的 NPSC 活動，例如：全國配對甜蜜典禮（National Pairing Sweet Ceremony）、全國餡餅搶購大會（National Pie Shopping Conference）、全國爬山大賽（National Pa Shan Contest）等。

根據 NPSC 國法律規定，一個「NPSC 活動」的英文名稱必須要由四個字串組成，且四個字串的開頭依序為 N、P、S、C。上一段所提到的三個活動，括號內的英文名稱皆符合 NPSC 活動的命名條件。

為了在 NPSC 國舉辦網際網路程式設計全國大賽，你想知道你的活動名稱有沒有符合以上的規定，請你寫一支程式檢查吧！

Input

輸入僅一行，包含四個字串 S_1, S_2, S_3, S_4 ，表示活動的名稱，字串間以一個空白隔開。

- S_1, S_2, S_3, S_4 長度不超過 10 個字元，且每個字串的第一個字元為大寫字母，其餘字母皆為小寫字母

Output

輸出一行，若活動名稱符合規定，請輸出 Yes；否則，請輸出 No。

Sample Input 1

Sample Output 1

National Problem Solving Contest	Yes
----------------------------------	-----

Sample Input 2

Sample Output 2

Not An Npsc Event	No
-------------------	----

Sample Input 3	Sample Output 3
Ntu Cool Super Poor	No

D. 數數挑戰

Problem ID: counting

Zisk 是一位天才兒童，他常常給自己一些與數字有關的挑戰來訓練自己，而今天的挑戰是這樣的：

首先，Zisk 會先收到兩個正整數 N, M ，接著，他會把「 $1 \sim N$ 選一個數字和 $1 \sim M$ 選一個數字所形成的數對」全部數一遍，也就是從 $(1, 1)$ 開始，數 $(1, 2), (1, 3), \dots$ 一直到 $(1, M)$ ，接著再數 $(2, 1), (2, 2), \dots, (2, M)$ ，最後數到 $(N, 1), (N, 2), \dots, (N, M)$ ，共 $N \times M$ 個數對。

每當他數到一個數對 (a, b) 時，他必須把這兩個數字**串接起來**，在這個挑戰的最後，Zisk 必須算出他所有串接起來的數字和。

更詳細地說，串接兩個數字的方式是直接把兩個數字寫在一起當成一個新的數字，舉例來說，串接 2 和 3 就會得到 23、串接 48 和 763 就會得到 48763、串接 100 和 1 就會得到 1001。

現在，Zisk 收到了他的兩個正整數，但他不知道自己完成挑戰後算出的數字和是不是正確的，請你幫他撰寫一支程式，在輸入兩個正整數後能夠告訴 Zisk 他應該要算出的正確數字和是多少。

舉例而言，若 $N = 2, M = 3$ ，那 Zisk 必須要數出 $2 \times 3 = 6$ 個數對，分別是 $(1, 1), (1, 2), (1, 3), (2, 1), (2, 2), (2, 3)$ ，這些數對串接起來之後分別是 11, 12, 13, 21, 22, 23，其總和是 $11 + 12 + 13 + 21 + 22 + 23 = 102$ ，因此若輸入是 2 3，那你應該要輸出 102 作為 Zisk 應該要算出的正確數字和。

Input

輸入只有一行，包含兩個正整數 N, M ，代表 Zisk 收到的兩個整數。

- $1 \leq N, M \leq 100$

Output

輸出 Zisk 應該要算出的正確數字和於一行。

Sample Input 1	Sample Output 1
2 3	102

Sample Input 2	Sample Output 2
5 3	480

Sample Input 3	Sample Output 3
10 10	11000

E. 比賽轉播

Problem ID: race

「我們摯愛的 AT7，於民國 111 年 11 月 12 日中午，悄悄的離開這個競賽，我們痛徹心扉，就僅僅一眨眼的時間，天人永隔。AT7 安祥的走完了 80000000 年的競賽人生旅程，他彷彿在沉睡中做了一個美夢，夢醒了，留下陪伴我們成長過程中的點點滴滴，留下我們永恆的追思與感恩。」

慙乙己是一名專業的競賽轉播員，常常轉播各大資訊競賽的比賽過程。轉播除了嘴砲選手、發複製文和討論題目之外，也會有關於參賽者的資訊，例如：

「喔喔喔，第 a_i 隊的分數又上升了 b_i 分！」

你在看轉播的時候聽到了 N 則這樣的訊息，你能根據這些資訊來推出第一名是誰嗎？

備註：第一名是分數最高的隊伍，而所有隊伍在第一則轉播訊息前都是 0 分。你可以假設轉播員有轉播到所有的分數變化。

Input

輸入第一行有兩個正整數 N, M ，代表有幾則訊息以及總共有幾隊。接下來有 N 行，第 i 行有兩個正整數 a_i, b_i 代表第 i 則訊息提到的隊伍編號以及上升分數。

- $1 \leq N \leq 10^5$
- $2 \leq M \leq 10^5$
- $1 \leq a_i \leq M$
- $1 \leq b_i \leq 10^4$
- 保證最終的第一名只有一個（不會有兩隊同分並列第一）
- 所有輸入都是正整數

Output

輸出一行包含一個數字，代表最後獲得第一名的隊伍編號。

Sample Input 1

3 2 1 2 2 3 1 2	1
--------------------------	---

Sample Output 1**Sample Input 2**

4 3 1 1 2 1 3 1 1 1	1
---------------------------------	---

Sample Output 2**Sample Input 3**

5 2 1 1000 1 1000 1 1000 2 10000 1 1000	2
--------------------------------------------------------	---

Sample Output 3

F. 七對子

Problem ID: 7pairs

「榮，立直一發自摸斷么七對清一色赤寶一，役滿。」

熱愛日本麻將（以下簡稱「日麻」）的優希，對於「七對子」這個役種有著異常的喜愛。於是，她想請你判斷她的手牌距離達到「七對子」還有多少「向聽」。

或許你正擔心著你對日麻不夠了解——不用擔心，優希會講解的比四年前還要更詳細。

日麻是一種麻將遊戲，使用的牌分為萬子牌、餅子牌、索子牌及字牌。萬子牌、餅子牌、索子牌為數字牌，每個類別皆有九張不同的牌，分別為「一萬」到「九萬」、「一餅」到「九餅」、「一索」到「九索」。而字牌一共由七張不同的牌組成，依序為「東」、「南」、「西」、「北」、「白」、「發」、「中」。在一副日麻牌中，所有的萬子牌、餅子牌、索子牌、字牌分別各有 4 張，整副日麻一共由 $(9 + 9 + 9 + 7) \times 4 = 136$ 張牌組成。

為了方便表示，每張牌皆以一個數字搭配一個字母作為其代碼。數字表示其序數，字母表示其種類：萬子為 m、餅子為 p、索子為 s、字牌為 z。舉例來說：「七萬」的代碼為 7m、「二索」的代碼為 2s、「中」的代碼為 7z。

	1	2	3	4	5	6	7	8	9
萬子 (m)									
餅子 (p)									
索子 (s)									
字牌 (z)									

（日麻牌圖片取自維基百科，由 Cangjie6 上傳，以創用 CC 姓名標示-相同方式分享 4.0 國際授權條款釋出。）

在大部分的日麻牌局中，通常會加入一種特別的牌——赤寶牌。赤寶牌一共有三種：赤五萬、赤五餅、赤五索，代碼分別為 0m、0p、0s。一個常見的赤寶牌規則為，取出五萬、五餅、五索各一張，並加入赤五萬、赤五餅、赤五索各一張。這樣一來整副牌的數量沒有變動，只是其中一張五萬、五餅、五索被換成了赤寶牌。在本題中我們也將採用上述的三張赤寶牌的日麻規則。赤寶牌與對應的一般牌視為兩張相同的牌，但在計分時手上的赤寶牌數量將提升得到的點數。

在瞭解了整副日麻牌的組成後，優希終於可以開始講解日麻遊戲的流程了。在日麻的遊戲過程中，玩家大部分時候手上皆持有 13 張牌，當輪到玩家「摸牌」時，玩家會取一張牌加入手牌。當手上的 13 張牌與取來的牌組成特定的組合時，即為「自摸」。若取來的牌沒能達成特定組合，玩家必須從 14 張牌中選擇其中一張打出，回到手上只持有 13 張牌的狀態。

在所有的組合中，優希最喜歡的便是「七對子」了。「對子」指的就是兩張相同的牌，顧名思義，七對子就是手上的 14 張牌分別能組成 7 個不同的對子。更精確的說，無論手牌的順序為何，只要手上的 14 張牌能分成七組，且每組牌為兩張相同的牌，並且每一組之間的牌皆相異，則這 14 張牌形成「七對子」。特別注意的是，赤寶牌與其對應的一般牌可以組成一個對子，並且其與兩張對應的一般牌組成的對子被視為是相同的。例如：五餅和赤五餅可以組成一個對子，但其與兩個五餅組成的對子被視為是相同的。

最後，優希要教你什麼是「聽牌」與「向聽」。對於手上的 13 張牌，若存在某一張牌在取進手牌後可以滿足「七對子」條件，這 13 張牌就被稱為是七對子「聽牌」。而七對子的「向聽」數量即表示至少還要再「摸牌」幾次才能達到「聽牌」的條件。在本題中，我們將「聽牌」狀態的「向聽」數量視為 0。

舉例來說，1m 1m 2m 2m 3m 3m 4m 4m 5m 5m 6m 6m 7m 這 13 張牌達到七對子的「聽牌」，因為只要再取進一張 7m，即可滿足七對子的條件；而 1m 1m 2m 2m 3m 3m 4m 4m 0m 5m 6m 7m 8m 的七對子「向聽」數量即為 1，因為如果在下次「摸牌」時取進 6m 並打出 7m 或 8m，即可達到「聽牌」條件。

在說明完了一大串的日麻規則後，想必你已經對日麻躍躍欲試了！給你優希手上的 13 張牌，她想請你判斷這些牌的七對子向聽數。為了向優希學到更多日麻知識，請你寫一支程式幫優希吧！

Input

輸入僅一行，包含 13 個字串 S_1, S_2, \dots, S_{13} ，表示優希手上的十三張牌的代碼，字串間以一個空白隔開。

- S_i 皆為題目敘述中說明的代碼，且每種牌出現的數量不會超過題目敘述中說明的數量。

Output

輸出一行，包含一個整數，表示優希手牌的七對子向聽數。

Sample Input 1

1m 1m 2m 2m 3m 3m 4m 4m 5m 5m 6m 6m 7m	0
----------------------------------------	---

Sample Output 1

Sample Input 2

1m 1m 2m 2m 3m 3m 4m 4m 0m 5m 6m 7m 8m	1
----------------------------------------	---

Sample Output 2

Sample Input 3

1m 4z 2p 3s 0m 1m 3s 2p 4z 5m 6s 5m 5m	2
----------------------------------------	---

Sample Output 3

Sample Input 4

0s 1m 2m 3m 5s 1m 2m 3m 5p 1m 2m 3m 0p	3
----------------------------------------	---

Sample Output 4

Sample Input 5

1m 1m 1m 1m 2p 2p 2p 2p 3s 3s 4z 4z 0m	4
----------------------------------------	---

Sample Output 5

Sample Input 6

5p 9m 8m 7m 6m 5m 0p 5p 4m 3m 2m 1m 5p	5
----------------------------------------	---

Sample Output 6

This page is intentionally left blank.