



E. Plegma

time limit per test: 5 seconds

memory limit per test: 512 megabytes



This is a run-twice (communication) problem.

There are two players: Player A and Player B. The jury will first interact with player A. After player A ends their interaction, the jury will interact with player B. Note that player A and player B may not directly pass information to each other; both players are only able to send information or receive information from the jury, but they may agree on the strategy they will use to communicate.

The jury has a **binary grid** G of n rows and n columns (each cell of this grid has value either 0 or 1). Row 1 is the top-most row, and column 1 is the left-most column. The **connectivity** of this grid is defined as 1 if there is a path going left, right, up, or down going through only cells with value 1 connecting each pair of cells (i_1, j_1) and (i_2, j_2) with $G_{i_1, j_1} = G_{i_2, j_2} = 1$. Note that moving diagonally is not allowed. It is guaranteed that there exists at least one cell with value of 1 in this grid.

The jury first interacts with player A. The jury will give player A the grid G . After inspecting the grid, player A must determine **two integers** r and c and send them to the jury. At the start of player B's interaction, player B will receive **the values of all cells in the r 'th row and all cells in the c 'th column** from the jury. Note that player B is not given the values of r and c .

Codeforces Round 1063 (Div. 2).


比赛进行中

01:06:15

Contestant



→ 提交?

语言: GNU G++20 13.2 (64 bit, v )

选择文件: [选择文件](#) 未选择文件

Be careful: there is 50 points penalty for submission which fails the pretests or resubmission (except failure on the first test, denial of judgement or similar verdicts). "Passed pretests" submission verdict doesn't guarantee that the solution is absolutely correct and it will pass system tests.

提交

→ 评分表

	Score
Problem A	394

Player A wants to ensure player B can determine the connectivity of G . Your task is to act as both players and find a strategy so that player B is able to determine the connectivity correctly. Note that the communicator of this task is not adaptive – that is, the grid given to you on the first run will be the same as the grid used to evaluate the connectivity.

有道 翻译

✖

≡

📄

这是一个运行两次（通信）的问题。

有两个玩家：玩家A和玩家B。陪审团首先与玩家A互动。在玩家A结束互动后，陪审团将与玩家B互动。注意，玩家A和玩家B不能直接相互传递信息；双方玩家都只能发送信息或从陪审团那里接收信息，但他们可能会同意他们将使用的沟通策略。

陪审团有一个由 n 行和 n 列组成的**二进制网格** G （该网格的每个单元格的值要么为 0，要么为 1）。行 1 是最上面的行，列 1 是最左边的列。如果存在一条仅通过值为 1 的单元格的向左、向右、向上或向下的路径，将每对单元格 (i_1, j_1) 和 (i_2, j_2) 与 $G_{i_1, j_1} = G_{i_2, j_2} = 1$ 连接起来，则该网格的**连通性**定义为 1。注意，不允许对角线移动。保证在此网格中至少存在一个值为 1 的单元格。

陪审团首先与玩家A互动，陪审团将给玩家A网格 G 。在检查网格后，玩家A必须确定**两个整数** r 和 c ，并将它们发送给陪审团。在玩家B的交互开始时，玩家B将从陪审团获得 r 第一行和 c 第列**中所有单元格的值。注意，玩家B没有得到 r 和 c 的值。

玩家A希望确保玩家B能够确定 G 的连通性。你的任务是扮演两个玩家，并找到一个策略，使玩家B能够正确地确定连接。注意，此任务的通信器不是自适应的——也就是说，在第一次运行时提供给您的网格将与用于评估连通性的网格相同。

↕

📄

🔍

Input

Your code will be ran exactly two times on each test. On the first run, you will be Player A, and on the second Player B.

First Run Input

The first line of the input contains the string `first`. The purpose of this is so your program recognizes that this is its first run, and it should act as Player A.

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \leq t \leq 10^4$). The description of the test cases follows.

Problem B	788
Problem C	985
Problem D1	1182
Problem D2	985
Problem E	2167
Successful hack	100
Unsuccessful hack	-50
Unsuccessful submission	-50
Resubmission	-50

* If you solve problem on 00:53 from the first attempt

The first line of each test case contains two integers n and C ($2 \leq n \leq 1000, 0 \leq C \leq 1$) – the size of the grid and the connectivity respectively.

The following n lines contain information about the grid. The i th of these lines contains a binary string G_i of length n , indicating the i -th row of the grid.

It is guaranteed that:

- The sum of n^2 does not exceed $2 \cdot 10^6$ over all test cases
- The value of C matches with the information in the grid – that is, if $C = 1$, then the grid has connectivity 1, and if $C = 0$, then the grid has connectivity 0.
- Each grid has at least one 1 in the input.

Second Run Input

The first line of the input contains the string `second`. The purpose of this is so your program recognizes that this is its second run, and it should act as Player B.

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \leq t \leq 10^4$). The description of the test cases follows. Note that this number is equal to t from the first run input.

The first line of each test case contains exactly one integer n — the size of the grid given on the i 'th test case of the first run input.

The second line of each test case contains a binary string $G_{r,1}G_{r,2} \dots G_{r,n}$ — the contents in row r of the grid. Note that the integer r is sent by player A to the jury at the end of their interaction of the i 'th test case

The third line of each test case contains a binary string $G_{1,c}G_{2,c} \dots G_{n,c}$ — the contents in column c of the grid. Note that the integer c is sent by player A to the jury at the end of their interaction of the i 'th test case.

Hacks

To make hacks, use the following format:

The first line should contain exactly one integer t ($1 \leq t \leq 10^4$) — the number of grids. Then, t blocks of input should follow.

The first line of each block of input must contain a single integer n ($2 \leq n \leq 1000$) — the size of the grid the jury will choose.

Each of the next n lines should contain a binary string of size n . The i 'th of these lines should contain $G_{i,1}G_{i,2}\dots G_{i,n}$ — the contents of the i 'th row of the grid the jury will choose.

There must exist at least a single cell with value of 1 in each grid, and the sum of n^2 over all test cases should not exceed $2 \cdot 10^6$.

Note that the connectivity of each grid is determined by the jury, and you do not need to output it for a hack.



Output

For the first run, for each test case, output two integers r and c ($1 \leq r, c \leq n$). This indicates that you want the second run to receive the r -th row and the c -th column of the grid.

For the second run, for each test case, output an integer C ($0 \leq C \leq 1$) — the connectivity of the grid.

有道 翻译

** **输出

对于第一次运行，对于每个测试用例，输出两个整数 r 和 c ($1 \leq r, c \leq n$)。这表明您希望第二次运行接收网格的 r 第一行和 c 第列。

对于第二次运行，对于每个测试用例，输出一个整数 C ($0 \leq C \leq 1$) — 网格的连通性。

Examples

input	Copy
first 2 2 1 11 10 2 0 10 01	
output	Copy
2 2 2 1	

```
second
2
2
10
10
2
01
10
```

```
1
0
```

Note

On the first input example, the first grid is the following:

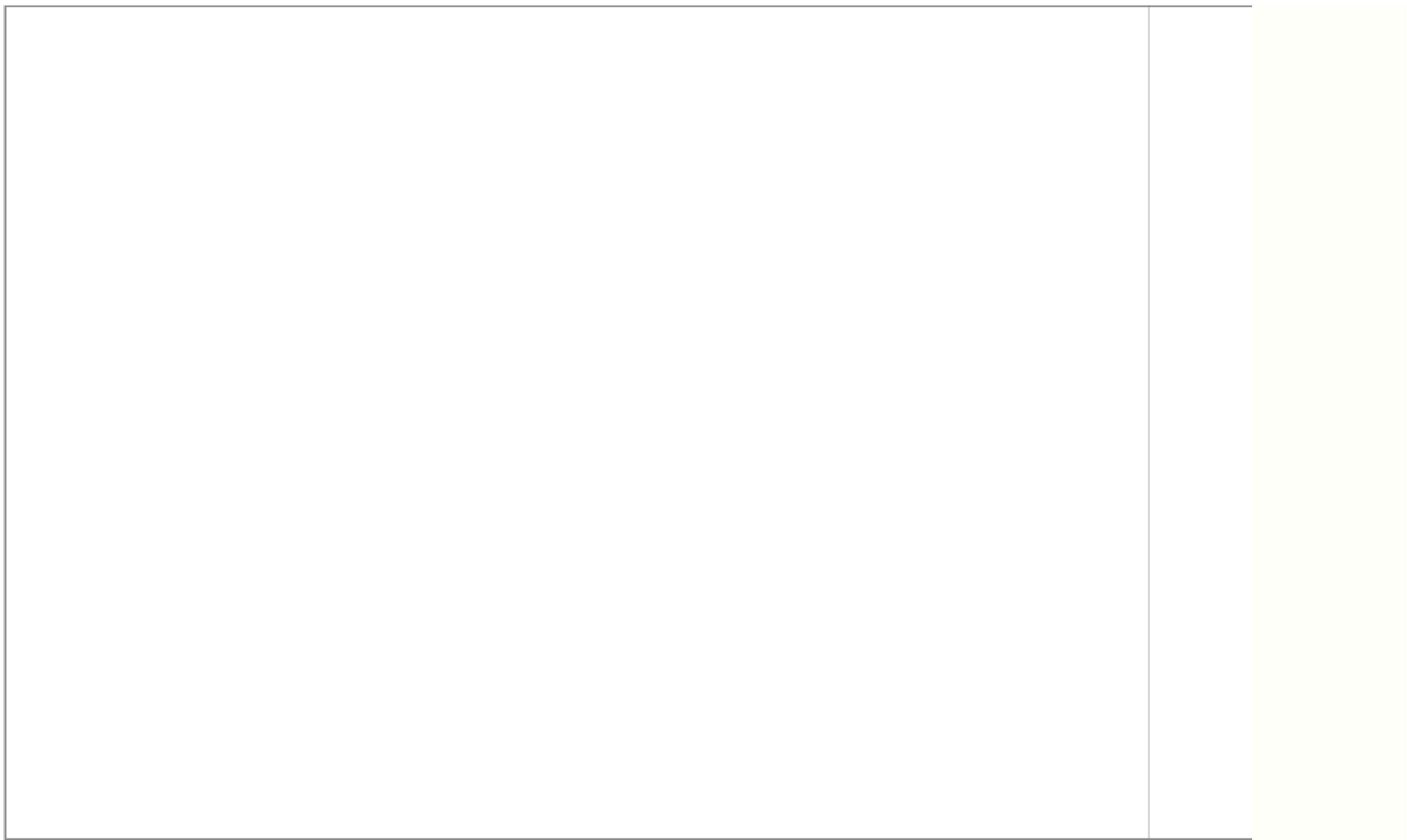
1	1
1	0

On the first run, we know that $n = 2$. After inspecting the example, we can determine that the connectivity is 1.

For the sake of this example, suppose Player A and Player B have agreed to some strategy where if the connectivity is 1, then player A should send a row and a column that both end with 0. In this case, row 2 and column 2 satisfies this strategy. **Note that this is an example strategy for the sake of demonstration, and using this strategy will not work for all cases.**

Then, take a look at the second run. Now, we are Player B. We receive that $n = 2$ and that the chosen row has values $r = [1, 0]$ and the chosen column has values $c = [1, 0]$. Since the last number of each row is 0, player B uses the agreed upon strategy to determine that the connectivity is 1.

GNU G++20 13.2 (64 bit, winlibs)



► 自定义测试数据(自动保存)

▶📄

Desktop version, switch to [mobile version](#).
[Privacy Policy](#) | [Terms and Conditions](#)

Supported by



ITMO