

USA Computing Olympiad

OVERVIEW

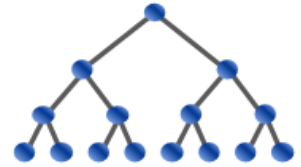
TRAINING

CONTESTS

HISTORY

STAFF

RESOURCES


[Return to Problem List](#)

USACO 2022 DECEMBER CONTEST, BRONZE PROBLEM 3. REVERSE ENGINEERING

Time Remaining: 3 hrs, 59 min, 16 sec

Not submitted yet

English (en) ▼

Elsie has a program that takes as input an array of N ($1 \leq N \leq 100$) variables $b[0], \dots, b[N-1]$, each equal to zero or one, and returns the result of applying a sequence of if / else if / else statements on the input. Each statement examines the value of at most one input variable, and returns either zero or one. An example of such a program might be:

```
if (b[1] == 1) return 1;
else if (b[0] == 0) return 0;
else return 1;
```

For example, if the input to the program above is "10" (that is, $b[0] = 1$ and $b[1] = 0$), then the output should be 1.

Elsie has told Bessie the correct output for M ($1 \leq M \leq 100$) different inputs. Bessie is now trying to reverse engineer Elsie's program. Unfortunately, Elsie might have lied; it may be the case that no program of the form above is consistent with what Elsie said.

For each of T ($1 \leq T \leq 10$) test cases, determine whether Elsie must be lying or not.

INPUT FORMAT (input arrives from the terminal / stdin):

The first line contains T , the number of test cases.

Each test case starts with two integers N and M , followed by M lines, each containing a string of N zeros and ones representing an input (that is, the values of $b[0] \dots b[N-1]$) and an additional character (zero or one) representing the output. Consecutive test cases are separated by newlines.

OUTPUT FORMAT (print output to the terminal / stdout):

For each test case, output "OK" or "LIE" on a separate line.

SAMPLE INPUT:

```
4

1 3
0 0
0 0
1 1

2 4
00 0
01 1
10 1
11 1

1 2
0 1
0 0

2 4
00 0
01 1
10 1
11 0
```

SAMPLE OUTPUT:

```
OK
OK
LIE
LIE
```

Here's a valid program for the first test case:

```
if (b[0] == 0) return 0;
else return 1;
```

```
else return 1;
```

Another valid program for the first test case:

```
if (b[0] == 1) return 1;  
else return 0;
```

A valid program for the second test case:

```
if (b[1] == 1) return 1;  
else if (b[0] == 0) return 0;  
else return 1;
```

Clearly, there is no valid program corresponding to the third test case, because Elsie's program must always produce the same output for the same input.

It may be shown that there is no valid program corresponding to the last test case.

SCORING:

- Inputs 2 and 3 have $N = 2$.
- Inputs 4 and 5 have $M = 2$.
- Inputs 6 through 12 have no additional constraints.

Problem credits: Benjamin Qi

Language: **Source File:**