IOITC 2021

Hidden Cell

There is an $N \times N$ matrix. The rows are numbered $0, 1, 2, \ldots, N-1$ top to bottom and the columns are numbered $0, 1, 2, \ldots, N-1$ left to right. There is a hidden cell (a, b). It is known that this cell doesn't lie on the boundary, that is $\min(a, b) > 0$ and $\max(a, b) < N - 1$.

Your task is to recover the hidden cell. You can ask queries, in which you give the judge a matrix M of size $N \times N$, consisting of zeroes and ones. Let's call a cell (i, j) valid if $(i, j) = (a, b)$ or $M[i][j] = 1$. The judge replies whether there exists a path from $(0, 0)$ to $(N-1, N-1)$ consisting of only valid cells, and going either down or right. Formally, it tells whether there exits a sequence of $2N-1$ cells $(0 = u_0, 0 = v_0), (u_1, v_1), \ldots, (N-1 = u_{2N-2}, N-1 = v_{2N-2})$, such that (u_i, v_i) is a valid cell for all i and either $(u_{i+1} = u_i, v_{i+1} = v_i + 1)$ or $(u_{i+1} = u_i + 1, v_{i+1} = v_i)$. To make such a query, you can call the function:

```
bool doesPathExist(const vector<string>& M)
```

where you pass the matrix M as a vector of strings of length N each, consisting of characters 0 and 1. The function returns true if there is a path as described above and false otherwise.

You have to implement the function:

```
pair<int, int> findHiddenCell(int N)
```

that makes queries and returns the cell (a, b).

Do NOT read anything from stdin or write something to stdout/stderr.

Test Data and Scoring

Each testfile consists of ≤ 25 testcases, all having $N = 50$.
If in any testcase of any testfile, you make an invalid query (where M is not an $N \times N$ matrix consisting of only zeroes and ones) or if you return incorrect hidden cell, you get a score of 0. Else, let Q be the maximum number of queries asked by you over all the testcases of all the testfiles.

- If $Q > 120$, you get 0 points.
- If $61 \leq Q \leq 120$, you get 9 points.
- If $51 \leq Q \leq 60$, you get 24 points.
- If $Q \leq 50$, you get $54 + \left\lfloor \frac{46 \times (50 - \max(Q, 14))}{36} \right\rfloor$ points, where $\lfloor x \rfloor$ denotes the largest integer $\leq x$. In particular, you get 54 points if $Q = 50$ and 100 points if $Q \leq 14$.

Local testing

You are provided with a dummy grader with the name `dummy_grader.cpp`.You should compile your solution (assumed to be in the file `solution.cpp`) as:

```
g++ solution.cpp dummy_grader.cpp -o grader
```

Then you can run `./grader`, and give input of the form as given in the sample input:

- The first line contains T, the number of testcases.
- The first line of each testcase contains N.
- The second line contains two integers, a and b, the row and column numbers of the hidden cell.

Limits

Time: 1 seconds

Memory: 256 MB