
Submit: `game.c` / `game.cpp` / `game.pas`
Input: via library
Output: via library

Time limit: seconds
Memory limit: 64 MB
Points: 100

Gods sometimes play exciting games with the destinies of the mortals. However, some of them are known for not being the sharpest strategists, so our only chance is to find a better strategy and win. We will describe an earthly equivalent of one such game, in the meantime look around carefully and beware of stray lightnings.

The game is played on a rectangular board with M rows and N columns. In each row, there is a single white stone and a single black stone. The black one belongs to the god, the white one is yours.

At the beginning of the game, all white stones are in the leftmost column and all black ones in the rightmost column. In each turn, you select one row and move the white stone to a different square in the same row, but you are not allowed to skip the opponent's stone nor place two stones on the same square. Then, the god makes the same with a black stone in some row.

Whoever cannot make a move, loses. If a game continues for 10 000 moves without a winner, a draw is declared.

Your goal is to play the game and to win.

Library

Your program has to use the following functions for communicating with the game board¹:

In C/C++:

```
#include "gamelib.h"
void g_init(int *pM, int *pN) – initializes the game. This function must be called before all other game
    functions. It sets the variables pointed to by pM and pN to the number of rows and columns of the game
    board. It's guaranteed that  $1 \leq M \leq 31$ ,  $2 \leq N \leq 32$  and that  $M$  is odd.
void g_move(int row, int delta) – moves your stone in the given row by the specified number of squares to
    the right (negative deltas are allowed). Rows are numbered from 0 to  $M - 1$ , columns from 0 to  $N - 1$ .
void g_giveup(void) – give up the game. Call if you cannot make a move.
void g_ask(int *prow, int *pdelta) – ask for the opponent's move and set the variables pointed to by prow
    and pdelta to the row in which the opponent plays and to the number of squares he moves his stone to the
    right by.
```

In Pascal: (the functions do the same as their C counterparts)

```
uses gamelib
procedure g_init(var M, N: longint)
procedure g_move(row, delta: longint)
procedure g_giveup
procedure g_ask(var row, delta: longint)
```

During the contest, you are supplied with an example implementation of the library, which reads board dimensions from the standard input and plays as the God of Chaos. This version also contains a `g_show()` function displaying the current state of the game.

Caveat: Your program must not print anything to its standard output nor read anything from its standard input! Also, all symbols starting with `g_` are reserved for use by the library.

¹ I.e., with the universe.