IOI Training Camp 2011 – Test 4, 14 June, 2011

Problem 1 Find the Position

Consider the following game designed by Milind Uncle for Nikhil and Lavanya. Milind fixes a number N. He then gives Nikhil and Lavanya a sequence of N 0/1 values each. It is known that the sequences are almost identical: they differ at exactly one position. Nikhil and Lavanya have to determine the position at which their values differ.

The only way in which they are allowed to communicate with each other is as follows: Nikhil is allowed to write down an integer that Milind will pass on to Lavanaya. In turn she can write down an integer and Milind will pass this on to Nikhil. And this goes on till one of them declares that they know the position at which their values differ. However, Milind Uncle is old and he is willing to pass on at most K integers before he calls the game off.

In this problem you have to write programs to play the roles of Nikhil and Lavanya in this game. You should implement two program components, `nikhil` and `lavanya`. Nikhil must implement the functions `nikhil_start(N, arr)` and `nikhil_turn(msg)`. Lavanya must implement the functions `lavanya_start(N, arr)` and `lavanya_turn(msg)`.

The function `nikhil_start` will be called by the grader exactly once in the beginning supplying `N`, the number of 0/1's in the sequence given to Nikhil and an array `arr` of length `N` consisting of 0/1 values. The function `lavanya_start` is similar. The grader will first call `nikhil_start` and then call `lavanya_start`. After this the functions `nikhil_turn` and `lavanya_turn` are called alternately conveying the number returned by one player to the next player.

The first call to `nikhil_turn` will be made with the argument -1. The value returned by this call will be the argument used in the first call to `lavanya_turn` and the value returned by the first call to `lavanya_turn` is used as the argument to the second call to `nikhil_turn` and so on. On having determined the position at which the two sequences differ, Nikhil or Lavanya should call `found_position(pos)` (from within `nikhil_turn` or `lavanya_turn`) where $1 \leq pos \leq N$. This will also terminate the program and no further calls to `nikhil_turn` or `lavanya_turn` will be made.

Subtask 1 [40 points]

$1 \leq N \leq 10000$. At most 30 calls to `nikhil_turn` and `lavanya_turn` in total are allowed.

Subtask 2 [60 points]

$1 \leq N \leq 10000$. At most 5 calls to `nikhil_turn` and `lavanya_turn` in total are allowed.

Time and memory limits

The time limit for this task is 2 seconds. The memory limit is 32MB.

Shared variables, file access and network access are prohibited. In C or C++, you may declare persistent variables to be static to retain information for Nikhil or Lavanya, while preventing them from being shared.