Finding a Top

There is a balanced binary tree with tops containing numbers from 1 to 101. At the beginning of the game, players are spawned at a certain top of this tree (start), their task – move to the top with a certain number (finish). Each player can move only to the nearby tops. The top, where the player is now, and nearby tops are visible.

At the beginning of the game 5 integers with a space are input: a, b, c, d and e, where:

a – number in a top to find;
b – number in a top, where the player is now;
c – number in top, which is a parent to current;
d – number in a left subsidiary top;
e – number in a right subsidiary top.

Each set of numbers comes in new line. Then, on each step only 4 numbers are output: b, c, d and e. Tops, which do not exist, have number -1 (parent top for root and subsidiary tops for leaves).

The player cannot get out of the tree's edge and get in a top with -1. If the player tries to get in a top with -1, then his move will be ignored, and he stays on position.

On each step a program should return a direction: «up», «left» or «right», where:
up – move into parent top;
left – move into left subsidiary top;
right – move into left subsidiary top.

Example:

Standard input	Standard output
25 75 50 63 88 50 -1 25 75 25 50 12 36	 up left left

N.B.: Do output with "new line" command («endl» in C/C++ and Console.WriteLine() in C#)