

2222 AC. The Earth Space Agency (ESA) is preparing a robot mission to Planet Atlantis, discovered just one century ago. The planet was named upon the old legend of the disappeared continent because its surface is almost entirely occupied by an immense sea. There are no continents in that planet, yet a significant part of its surface is covered by land: millions of small, unexplored islands are known to exist all over Atlantis.

One of the goals of the mission is to build a complete map of the planets islands. As the visibility conditions are not suitable for aerial observation, ESA decided to build a set of rover robots specifically intended to make coast tracking by land. The idea is to leave one of these robots at one point in the coast of an island and let him autonomously discover the map of the entire coast, then come back as the robot returns to the same place and take it to the next island.

You are working with the team that is programming the navigation software for these robots, and some decisions were already taken: the surface will be discretised into squared, equal-sized areas; for simplification purposes, each area will be considered as being entirely covered by land or water. Also, the robot will always start its job in a square of the coast, with the sea at its right; therefore, the coast will be tracked in a counter clockwise direction (see Figure 1).

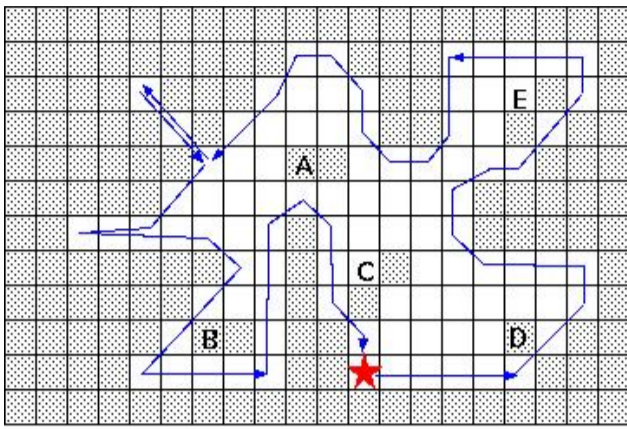


Figure 1

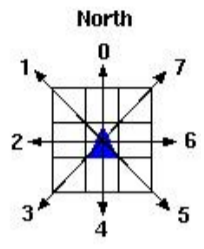


Figure 2

The robot will have a limited perception system that will be able to determine the kind of surface of the 8 areas neighbouring the one where the robot stands. Its movement capabilities will also be limited: the robot will only have the possibility of (i) rotating to one of 8 fixed directions (coded as integers from 0 to 7, 0 being North – see Figure 2) and (ii) moving to the neighbour position in front of him. Each time the robot moves to a new position, the perception system gets a new percept. There are no obstacles to worry about: God conveniently arranged the islands to have smooth, sand coasts. There are lakes (e.g., A, B, C, D and E in Figure 1), but the robot must not waste time with them: the goal is to track the seacoast only.

Given the current position and direction of the robot, and a percept of the surrounding world, decide the direction of the next move. Note that you are not being asked to program the whole coast-tracking software; you are just programming a part of it. The direction must be computed in such a way that an iterative call of the program, with percepts from an environment as the one described, would allow the robot to track the coast.

Your program must be able to process several scenarios. Each scenario comprises the robots current position and direction, and a percept. Figure 3 illustrates three hypothetical scenarios and the intended result: the direction of the next move.

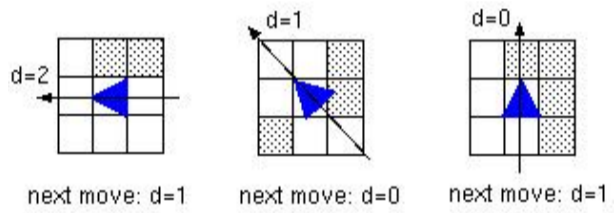


Figure 3

Input

The input file represents several scenarios. Input for each scenario consists of 9 lines as follows:

- First line: $x y d$, where x and y are the (always positive) coordinates of the robots current position, and d is the robots current direction, an integer such that $0 \leq d \leq 7$ (see Figure 2);
- Next 8 lines: $x_i y_i s_i$, where s_i is a number representing the surface type of the neighbouring position (x_i, y_i) ; land surface is represented by '1', and water surface by '0'.

Successive values in a line are separated by one or more blanks. The integer '-1' follows the data of the last scenario.

Output

For each given scenario, your program has to output the direction of the next robot's movement. Output for successive scenarios should be written in successive lines.

Sample Input

```
22 25 2
22 26 0
21 26 1
21 25 1
21 24 1
22 24 1
23 24 1
23 25 1
23 26 0
21 26 1
21 27 1
20 27 1
20 26 1
20 25 0
21 25 1
22 25 1
22 26 0
22 27 0
21 27 0
21 28 0
20 28 1
20 27 1
20 26 1
21 26 1
22 26 0
22 27 0
22 28 0
-1
```

Sample Output

```
1
0
1
```