Problem 6: Black Box Ray Tracing

Source filename:	`blackbox.(cpp|java)`
Input filename:	`blackbox.in`
Output filename:	`blackbox.out`

Since you are in a hurry, we'll first describe the objective of this problem. However, to fully understand the solution, you'll need to read the background material on pages 2 through 4 of this problem.

Objective

Steve enjoys playing the Parker Brothers' *Black Box* game; however, he would like to automate the task of tracing the rays that are sent into the black box. Write a program that reads input from a file named **`blackbox.in`** and produces the output described below.

Input File Specifications (`blackbox.in`)

The input file contains one or more test cases. The end of the input file is marked by a line that contains a single capital '`X`'. There are no spaces, no tabs, and no blank lines in the file

Each test case in the file begins with 8 lines that describe a configuration of the black box. Each of these 8 lines contains 8 ASCII characters. Each character is either a period '`.`' (ASCII = 46) or an at-sign '`@`' (ASCII = 64). An at-sign represents the presence of an "atom"; while a period represents a space. Every one of the 64 positions may be either a period or an ampersand.

The next line contains a single integer, `n` ($0 < n \leq 32$), that represents the number of rays that should be sent into the black box. The next `n` lines contain entry points for the rays that are to be inserted into the black box. Each entry point is designated by 2 characters - one of the letters '`T`', '`B`', '`L`', '`R`' followed by an integer between 1 and 8, inclusive. The letters indicate whether the ray should be inserted on the `T`op edge, the `B`ottom edge, the `L`eft edge or the `R`ight edge. The integer indicates which of the 8 positions along the designated edge should be used. All 32 entry points are illustrated in the figure below.

```
    T1 T2 T3 T4 T5 T6 T7 T8
L1  .  .  .  .  .  .  .  .   R1
L2  .  .  .  .  .  .  .  .   R2
L3  .  .  .  .  .  .  .  .   R3
L4  .  .  .  .  .  .  .  .   R4
L5  .  .  .  .  .  .  .  .   R5
L6  .  .  .  .  .  .  .  .   R6
L7  .  .  .  .  .  .  .  .   R7
L8  .  .  .  .  .  .  .  .   R8
    B1 B2 B3 B4 B5 B6 B7 B8
```

Output Specifications (`blackbox.out`)

For each entry point listed in each test case, output on a line by itself the result of sending a ray into the given black box configuration from that entry point. There are three possible types of results: (1) a Hit – which is reported using a single capital ‘`H`’, (2) a Reflection – which is reported using a single capital ‘`R`’, and (3) the resulting exit point – using the same syntax used to designate an entry point.

Example Input File

```
........
........
.@....@.
........
........
...@....
........
......@.
2
R1
L5
........
........
.@@.....
........
........
........
........
......@.
2
T3
B8
X
```

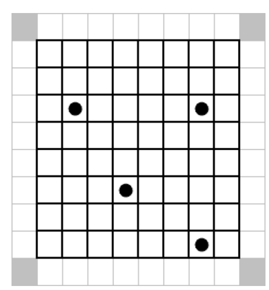
Resulting Output

```
L1
B5
H
R
```

Background[1]

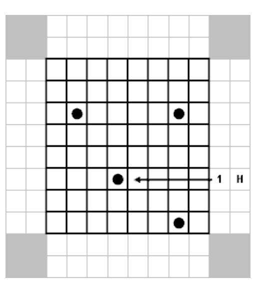
Black Box is a game of "hide and seek" for two players, which simulates shooting rays into a black box to deduce the locations of atoms hidden inside. It was created by Eric Solomon. The board game was published by Parker Brothers in the late 1970s. *Black Box* was inspired by the work of Godfrey Hounsfield who was awarded the 1979 Nobel Prize in Medicine for his invention of the CAT scanner.

Black Box is played on a two-dimensional 8-by-8 grid. The object of the game is to discover the location of objects ("atoms", represented by yellow balls in the Parker Brothers version) hidden within the grid, by the use of the minimum number of probes ("rays"). The atoms are hidden by one person, the “hider”. The other person, the “seeker”, designates where a ray should enter the black box and the hider announces the result (a Hit, Reflection, or its exit point). The seeker uses these results to deduce the position of the atoms in the black box



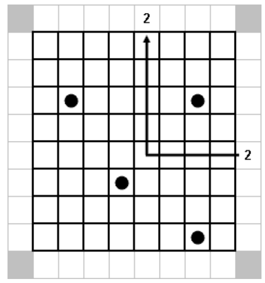
There are 32 input positions into the 8-by-8 grid, eight each at the top, bottom, right, and left. A ray is "fired" into one of these positions and the result is used to help deduce the location of a known number of hidden atoms.

Rays interact with atoms in three ways. A direct impact on an atom by a ray is a "hit" and the ray is absorbed. Thus, ray 1 fired into the box configuration below strikes an atom directly, generating a "hit", designated by an "H". A ray which hits an atom does not emerge from the box.



[1] Most of the wording and examples provided by Wikipedia.

The interaction resulting from a ray which does not actually hit an atom, but which passes directly to one side of the ball is called a "deflection". The deflection actually takes place in the space that is on the diagonal nearest the atom. The angle of deflection for this ray/atom interaction is 90 degrees. Ray 2 is deflected by the atom below, exiting the box as shown. Note that there is an entry and an exit location.



The final type of interaction of a ray with an atom is a "reflection", designated by an "R". This occurs in two circumstances. First, if an atom is at the edge of the grid, any ray which is aimed into the grid on either side of it causes a reflection.

In figure 1 below, rays 3 and 4 would each generate a reflection, due to the atom at the edge; ray 5 would be a hit on the atom.

The other circumstance leading to a reflection is when two deflections cancel out. In figure 2, ray 6 results in a reflection due to its interaction with the atoms in the grid. Reflections and hits can be more complex, too. In figure 3, ray 2 gets deflected by the first atom, reflected by the next two atoms and again deflected by the original atom, yielding a reflection. In figure 4, ray 3 below gets deflected by the first atom, then by the second atom, and then hits the third atom, yielding a hit.

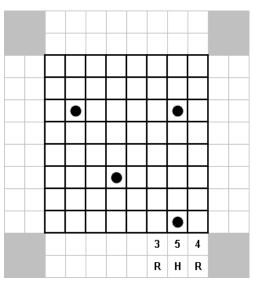


Figure 1

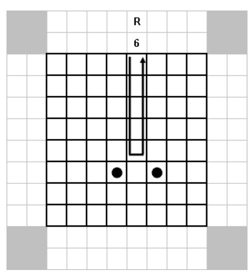


Figure 2

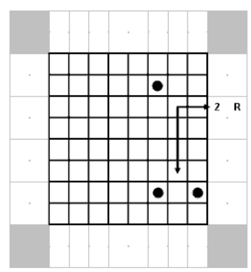


Figure 3

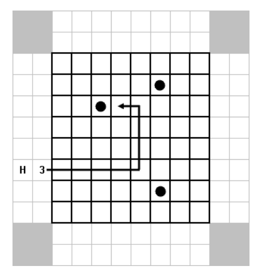
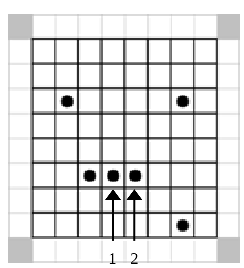
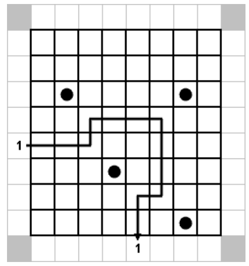


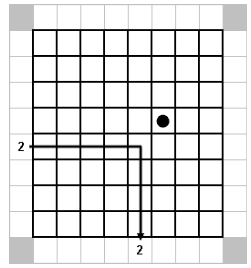
Figure 4

If two or more atoms are adjacent to each other, a hit takes precedence over a deflection or reflection. For instance, in the following figure, rays 1 and 2 would both be recorded as a hit:



Some rays travel a twisted course, like ray 1 below (this example corresponds to 2nd entry point for the 1st test case in the input file described on page 1). Notice that the exit point of this complex set of five deflections is exactly the same as that due to a single deflection, as shown by ray 2 below. Things are not always as simple as they seem within a black box.





The complete set of interactions of rays with a sample black box is shown below. Note that for deflections, the input and output locations are interchangeable - it does not matter if ray 2 below enters the box from the left side, or the top.