Jutge.org

The Virtual Learning Environment for Computer Programming

Battleship (1) X12847_en

GRAU-PRO1, FIB (2013-12-16)

Battleship (also Battleships or Sea Battle) is a guessing game for two players. It is known worldwide as a pencil and paper game which dates from World War I. The game takes place on a 10*x*10 board where the rows are numbered with letters (from *a* to *j*), and the columns are numbered with numbers from 1 to 10.

Each player has a board, on which he places 10 ships. Each ship occupies a number of consecutive squares on the grid, arranged either horizontally or vertically, but never in diagonal. A ship can neither touch nor overlap with any other ship. The ships can be 2, 3, 4 or 5-square long. The most common setting for the fleet is: 1 ship of length 5, 2 ships of length 4, 3 ships of length 3, and 4 ships of length 2.

As long as those rules hold, the player can place their ships wherever they want in the board. Then the game proceeds in rounds, and each player will try to sink the ships of the oppossite player by 'shooting' directly to a valid coordinate.

Input

- The first 10 lines define a valid board for player 1. Every line defines the position of a ship on the board. Every ship is defined by the coordinate of its initial square on the board, its size, and its alignment: the coordinate is a pair formed by a letter in $\{a..j\}$ and a number in $\{1..10\}$, the size is an integer number between 2 i 5, and the alignment is either a letter *h* (horizontal) or a *v* (vertical). The ship has to be place on the board starting on its initial coordinate, following its alignment, along as many squares as indicated in its size. For horizontal aligments, the initial coordinate represents its leftmost square of the ship, while for vertical alignments it represents the uppermost square of the ship.

  For example, the ship defined by ***a1 3 h*** occupies the squares ***a1, a2, a3***, while the ship defined by ***a1 3 v*** occupies the squares ***a1, b1, c1***.

- Following, there is a sequence (without repetitions) of valid coordinates representing the shoots of player 2 on player 1's board.

Output

- Draw player 1's board after placing on it the 10 ships specified on the input data.
- For every coordinate of the shooting sequence from player 2, write down whether there is a ship on that position. When no ship is placed on it, compute the distance from that position to the closest ship on the board.

Follow the format shown on the examples below.

Observation

- The board and the shooting coordinates from the testing bechmark follow the restrictions of the game. Then, you do not need to check them.
- Use the following definition:

  typedef vector< vector<**bool**> > *Board*;

- The distance between two squares (i,j) i (i',j') of the board is defined as the maximum difference between their x and y coordinates, i.e.,

  $$distancia((i,j),(i',j')) = max\{\mid i-i' \mid, \mid j-j' \mid\}$$

  Notice that, according to this formula, the positions at distance 1 are all the adjoining positions (horizontal, vertical and in diagonal), the positions at distance 2 are all the adjoining positions to those at distance 1, etc.

- An efficient solution for the algorithm computing the closest ship will be appraised.

Sample input

```
a1 5 h
a10 4 v
g10 4 v
j5 3 h
j1 3 h
f1 3 v
c2 2 h
e4 2 v
h5 2 h
a7 2 h

a1
a5
a6
d8
e8
f8
e6
e7
```

Sample output

```
  12345678910
a XXXXX.XX.X
b .........X
c .XX......X
d .........X
e ...X......
f X..X......
g X........X
h X...XX...X
i .........X
j XXX.XXX..X

a1 touched!
a5 touched!
a6 water! closest ship at distance 1
d8 water! closest ship at distance 2
e8 water! closest ship at distance 2
f8 water! closest ship at distance 2
e6 water! closest ship at distance 2
e7 water! closest ship at distance 3
```

Problem information

Author : Maria J. Blesa i Maria J. Serna
Translator : Maria Serna
Generation : 2013-12-16 13:41:04