

## 2002 Sun Microsystems and TopCoder Collegiate Challenge – Problem Statement

### MuseumGuard PROBLEM STATEMENT:

A museum has a sculpture exhibit room, with a number of statues in it. A guard is assigned to the room. The museum would like to know where to place the guard so that he is as close to all statues as possible.

The NxN room is given as a String[]. Each element of the String[] represents an West-to-East cross section of N cells of the room. The elements of the String[] are ordered from North to South. Dashes (-) represent empty space. The letter S represents a statue. The position that the security guard should occupy is such that the average distance (integer average) between the guard and each statue is minimized. The distance between the guard and a statue is the sum of: the vertical distance between the guard and the statue and the horizontal distance between the guard and the statue.

The position at the North-West corner is (0,0). The position at the South-East corner is (N-1,N-1). The position at the North-East corner is (0,N-1). The position at the South-West corner is (N-1,0).

The distance between two points on the grid (a,b) and (c,d) is given by  $|c-a| + |d-b|$  where  $||$  is the absolute value symbol.

Implement a method to return the position at which the security guard should be placed. Return the position formatted as a string with the North-South coordinate (Y-coordinate) followed by a comma, followed by the West-East coordinate (X-coordinate).

If two or more positions have the same integer average distance, choose the western most of the two (or more). If they are equally western, choose the northern most of the two (or more) positions. If there are no statues, the northwest-most cell on the board is where the security guard should be stationed.

The security guard cannot be placed in the same cell as a statue.

### DEFINITION:

Class Name: MuseumGuard

Method Name: getBestPos

Parameters: String[]

Returns: String

Method Signature: String getBestPos(String[] roomData)

(Be sure your method is public)

TopCoder will ensure the validity of the inputs. Inputs are valid if and only if all of the following criteria have been met:

- \*roomData has between 1 and 10 elements.
- \*the length of each element of roomData is the same as the number of elements in roomData, i.e. the room is square.
- \*the elements of roomData are strings that contain only the characters 'S' and '-'.
- \*at least one character of one element of roomData will be a '-'.

### NOTES:

- \*it is possible to have no statues in the room.

\*in the examples, the first character of element 0 in roomData is considered the northwest-most cell of the room.

\*in the examples, the first character of the last element in roomData is considered the southwest-most cell of the room.

\*in the examples, the last character of element 0 in roomData is considered the northeast-most cell of the room.

\*in the examples, the last character of the last element in roomData is considered the southeast-most cell of the room.

\*the first coordinate in a cell's coordinate ordered pair is the North-South coordinate, the second coordinate is the West-East coordinate

#### WORKED EXAMPLES:

```
roomData =  
{ "--S",  
  "----",  
  "--S" }
```

The avg distance to the statues from each empty cell are given in this table:

Cell:	Avg Distance:
(0,0)	$(2 + 4) / 2 = 6/2 = 3$ (integer division).
(0,1)	$(1 + 3) / 2 = 4/2 = 2$
(1,0)	$(3 + 3) / 2 = 6/2 = 3$
(1,1)	$(2 + 2) / 2 = 4/2 = 2$
(1,2)	$(1 + 1) / 2 = 2/2 = 1$
(2,0)	$(4 + 2) / 2 = 6/2 = 3$
(2,1)	$(3 + 1) / 2 = 4/2 = 2$

There is only one cell whose average distance is closest to the statues: (1,2). Return "1,2".

```
roomData =  
{ "--S-",  
  "----S",  
  "-----",  
  "S----" }
```

The avg distance to the statues from each empty cell are given in this table:

Cell:	Avg Distance:
(0,0)	$(2 + 4 + 3) / 3 = 9/3 = 3$ (integer division).
(0,1)	$(1 + 3 + 4) / 3 = 8/3 = 2$
(0,3)	$(1 + 1 + 6) / 3 = 8/3 = 2$
(1,0)	$(3 + 3 + 2) / 3 = 8/3 = 2$
(1,1)	$(2 + 2 + 3) / 3 = 7/3 = 2$
(1,2)	$(1 + 1 + 4) / 3 = 6/3 = 2$
(2,0)	$(4 + 4 + 1) / 3 = 9/3 = 3$
(2,1)	$(3 + 3 + 2) / 3 = 8/3 = 2$
(2,2)	$(2 + 2 + 3) / 3 = 7/3 = 2$
(2,3)	$(3 + 1 + 4) / 3 = 8/3 = 2$
(3,1)	$(4 + 4 + 1) / 3 = 9/3 = 3$
(3,2)	$(3 + 3 + 2) / 3 = 8/3 = 2$
(3,3)	$(4 + 2 + 3) / 3 = 9/3 = 3$

There are 9 cells an average distance of 2 from the statues. The westernmost cells are on column 0 (X = coordinate 2 = 0). The northernmost of these cells is (1,0). Return "1,0".

```
roomData =  
{ "----S",  
  "-----",  
  "-----",  
  "----S" }
```

After calculating all the distances for each cell, we see that the avg distance for cell (1,3) and for cell (2,3) (Avg distance =  $(1 + 2) / 2 = 1$ ) are the same. We choose the western-most cell, of which there is none, and then choose the northern-most cell, which is (1,3). Return "1,3".

TEST CASES:

```
{ "-----", "---SSS", "--S---", "---S--", "SSSSSS", --S---" } returns "3,2".  
{ "SSSSS-----", "---SSSS---", "---SSSSS--", "-----S", "-----S-----", "-----  
SS", "SSS--SS---", "---SSSSSS-", "-----", "---SSSSSS-" } returns "4,4".  
{ "--", "--" } returns "0,0".
```