

Program Name: ship.java      Input File: ship.in

Playing the game of battleship requires players to place ships on a rectangular grid hidden from the other player. Players then take turns taking 'shots' at their opponent's board by calling out grid positions (e.g., E3) and having the opponent indicate whether the shot was a hit or miss.

Write a program that will read in a given starting position for one player and then process the shots made by that player's opponent. The program will indicate whether each shot is a miss or a hit. If it is the last hit on a given ship, it will print a special message indicating that the given ship has been sunk.

### Input

The first line of input will contain a single integer,  $n$ , indicating the number of data sets to process. The remainder of the input consists of those  $n$  data sets.

Each data set will consist of three parts:

1. Player A's game board. This consists of 10 lines of 10 characters each. Each character either shows empty ocean (represented by a period) or a piece of a ship. There is exactly one of each type of ship on the board, and each is placed either horizontally within a single row or vertically within a single column. Different ships are represented by different capital letters, as shown below along with the ship lengths:  
'C' – carrier (5 units)  
'B' – battleship (4 units)  
'D' – destroyer (3 units)  
'S' – submarine (3 units)  
'P' – patrol boat (2 units)
2. A line containing a single integer,  $m$ , indicating the number of shots Player B will make (less than 20).
3. A line containing a space-separated list of all of Player B's shots. Shots are of the format <row><col> where <row> is A, B, C, etc. starting from the top, and columns are 1, 2, 3, etc. starting from the left. For example, if Player B makes 3 shots, this line might look like "A5 D2 G7".

### Output

For each data set in the input display the following:

1. A single line, "Data Set #X" where X is 1 for the first data set, 2 for the second, etc.
2. The list of Player B's shots, one per line, followed by the result of that shot. If the shot hits the last remaining piece of one of Player A's ships, the result of the shot is "sank my <stype>!" where <stype> is the type of ship. If the shot hits a ship which still has some portion that has not been hit, the result of the shot is "hit!". Otherwise, the result of the shot is "miss".

### Example Input File

```
2
.....
.....
..CCCCCB..
..D....B..
..D....B..
..DSSSPB..
.....P...
.....
.....
.....
3
A5 D2 G7
....C.....
....C.....
....C.....
....C.....
....CBBBB.
..DDDS....
.....S....
.....S....
P.....
P.....
14
A1 A10 J10 J1 I1 E5 E4 E6 F6 G6 H6 F5 F4 F3
```

### Example Output To Screen

```
Data Set #1
A5 miss
D2 miss
G7 hit!
Data Set #2
A1 miss
A10 miss
J10 miss
J1 hit!
I1 sank my patrol boat!
E5 hit!
E4 miss
E6 hit!
F6 hit!
G6 hit!
H6 sank my submarine!
F5 hit!
F4 hit!
F3 sank my destroyer!
```