Chess

Your task is to write a program that reads a chess board configuration and output every black piece position that can be attacked by white piece. White pieces will be represented by uppercase letters whereas black pieces will be represented by lowercase letters.

For those unfamiliar with chess, here are the movements of each piece:

- **Pawn** (p or P): can only move straight ahead, one square at a time. But it takes pieces diagonally (and that's what concerns to you in this problem).
- **Knight** (n or N) : have a special movement and it's the only piece that can jump over other pieces. The knight movement can be viewed as an "L". See the example bellow.
- **Bishop** (b or B) : can move any number of squares diagonally (forward or backward).
- **Rook** (r or R) : can move any number of squares vertically or horizontally (forward or backward).
- **Queen** (q or Q) : can move any number of squares in any direction (diagonally, horizontally or vertically, forward or backward).
- **King** (k or K) : can move one square at a time, in any direction (diagonally, horizontally or vertically, forward or backward).

Movements examples (asterisk character indicates where the piece can take another pieces):

Pawn	Rook	Bishop	Queen	King	Knight
....	...*....	.......*	...*...*	....	....
....	...*....	*.....*.	*..*..*.	....	....
....	...*....	.*...*..	.*.*.*..	....	..*.*...
....	...*....	..*.*...	..***...	..***...	.*...*..
...p....	***r****	...b....	***q****	..*k*...	...n....
..*.*...	...*....	..*.*...	..***...	..***...	.*...*..
....	...*....	.*...*..	.*.*.*..	....	..*.*...
....	...*....	*.....*.	*..*..*.	....	....

Format Input

There will be an integer T, indicating the number of board configurations on the input. Each board will consist of 8 lines of 8 characters each. A '.' character will represent an empty square. Upper and lower case letters (as defined above) will represent the pieces. There will be no invalid characters. You may assume that the starting position for black piece in upper board.

Format Output

For each board configuration you read, you must output the number of testcase start from 1 (Case #X), followed by every black piece position that can be attacked by white piece. If there're no black piece that can be attacked by white piece, output "-1". If there are more than 1 piece that can be attacked by white piece, you need to output **the piece position** using the standard chessboard positioning (A 1, A 2, B 1, …. , H 8) lexicographically.

Constraints

1 <= T <= 10

Sample Input	Sample Output
3 ..k..... ppp.pppp .... .R...B.. ....Q..... PPPPPPPP K....... rKkkrbnP NbQbbK.p pqQ.KBQr Q.RpkR.b ..Qnp..p nbKBrkP. Qq..pNb. prK.npRn bnppR..N Bq..kPKB k.bP.nPn pNBP.Rk. rPR.N..K .qN..rp. B..R.Q.B .RrRnKBk	Case #1: C 8 B 7 C 7 G 7 H 7 Case #2: A 8 C 8 D 8 E 8 F 8 G 8 B 7 D 7 E 7 H 7 A 6 B 6 H 6 D 5 E 5 H 5 D 4 E 4 H 4 A 3 B 3 F 3 B 2 E 2 G 2 A 1 B 1 F 1 H 1 Case #3: B 8 D 8 E 7 C 6 F 6 H 6 A 5 G 5 A 4 B 3 F 3 G 3 C 1 E 1

Note:
It's easier to do this problem by modularize your program using **functions**. Please **learn to use** function **properly** using this problem.

See https://en.wikipedia.org/wiki/Chessboard for the chessboard diagram notation.

Given two different sequences of the same length, $a_1a_2...a_k$ and $b_1b_2...b_k$, the first one is smaller than the second one for the lexicographical order, if $a_i<b_i$ (for the order of A), for **the first i** where a_i and b_i **differ**.

For better visualization, please look at the movement of each piece from these links:
https://en.wikipedia.org/wiki/King_(chess)
https://en.wikipedia.org/wiki/Queen_(chess)
https://en.wikipedia.org/wiki/Rook_(chess)
https://en.wikipedia.org/wiki/Bishop_(chess)
https://en.wikipedia.org/wiki/Knight_(chess)
https://en.wikipedia.org/wiki/Pawn_(chess)