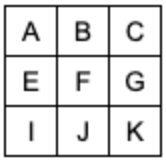
Albert enjoys playing "Mini BINGO" on a 3x3 grid. You don't need to know about BINGO to solve this problem.



To play Mini BINGO, Albert begins by writing down 9 distinct English upper-case alphabets on the 3x3 grid. Then, he arbitrarily shuffles these 9 alphabets to choose a "seed" string S of length 9.

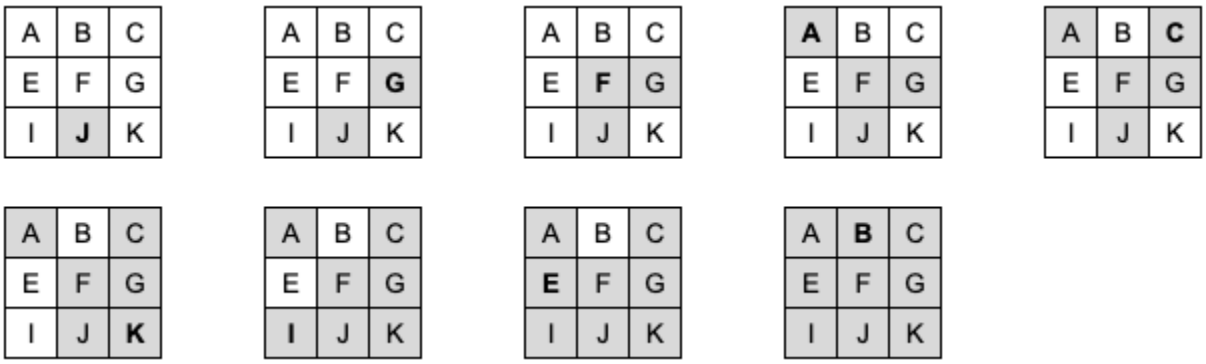
Albert will then color the alphabet cells according to the order given by the seed string, and calculate the score for each cell as follows.

- If all 3 cells in the same row as the cell being colored have been colored, then add 1 point to the cell's score.
- If all 3 cells in the same column as the cell being colored have been colored, then add 1 point to the cell's score.
- If all 3 cells in the main diagonal (A, F, and K in the example above) have been colored and the cell being colored is also in the main diagonal, then add 1 point to the cell's score.
- If all 3 cells in the anti diagonal (C, F, and I in the example above) have been colored and the cell being colored is also in the anti diagonal, then add 1 point to the cell's score.

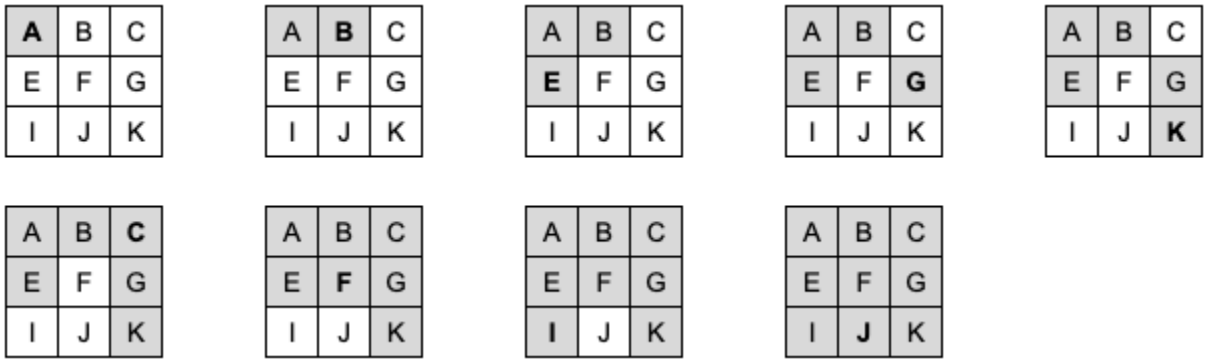
Given these rules, each cell's score will always be between 0 and 4 (inclusive), and we can obtain a string of length 9 by writing the scores of the cells in the same order as the seed string -- let us call this score string T(S).

For instance, consider the seed string S_0 = "JGFACKIEB". In the figure below, the first row's five images and the second row's four images show how the grid will be colored in the order given by the seed string.

- The first five cells being colored are "J", "G", "F", "A", and "C", each cell's score is 0.
- The sixth cell to be colored is "K", which yields 2 points due to the main diagonal and column 3.
- The seventh cell to be colored is "I", which yields 2 points due to the anti diagonal and row 3.
- The eighth cell to be colored is "E", which yields 2 points due to column 1 and row 2.
- The final cell to be colored is "B", which yields 2 points due to column 2 and row 1.
- As a result, the score string Albert obtains will be $T(S)_0$ = "000002222".



In the same grid, if the seed string is S_0 = "ABEGKCFIJ", then the grid will be colored as shown below, and the score string will be "000002222".



As these examples show, different seed strings can yield the same score string.

Albert believes that it is too trivial to compute $T(S)$ given the grid's alphabets and the seed string S. Hence, after computing the score string of S, Albert wants to find the seed string that yields the same score string $T(S)$) and comes lexicographically first. Let's help Albert.

Input

The first line will contain S, the number of test cases.

Each test case's first line will contain the seed string S of length 9.
The next three lines will describe the 3x3 grid by containing one string per line without whitespace.

Output

For each test case, output in a single line the score string $T(S)$ and the seed string that yields it and comes lexicographically first.

Limit

- $1 \le S \le 100$
- The seed string S will be of length 9 and will only contain alphabets 'A'-'Z'. S will not contain duplicate alphabets.

- The 9 alphabets that describe the game grid will not contain duplicates, and these are exactly the alphabets given by S.

Sample Input 1 Copy

```
4
JGFACKIEB
ABC
EFG
IJK
ADSFGHJKL
ASD
FGH
JKL
QPWOEIRUT
QWE
RTU
IOP
AZSXDCFVG
ZFC
DGX
ASV
```

Sample Output 1 Copy

```
000002222 ABEGKCFIJ
001001213 ADSFGHJKL
000011114 EIOQPRUWT
```

```
000010124 ACDSVFXZG
```

Case 1: Described in the problem statement.

Case 2: The input seed string may come lexicographically first.

Cases 3-4: No explanation provided.

Hints

Lexicographic Order: Given two different strings A and $B_{.}$ of the same length $K_{.}$ where the two strings first differ at position i, let A[i] and B[i] be the characters of A and B at position i, respectively. Then, the lexicographic order of the strings A and B follow the lexicographic order of the alphabets A[i] and B[i].

Time Limit

- Java 8: 3 seconds
- PyPy3: 3 seconds