

3499 No Fold'em Hold'em

The recent Texas Hold'em craze has spawned a bunch of on-line Poker sites. These sites tend to breed a group of poor players whose primary strategy is to never fold or throw away a hand. These players will always play and bet on every card on every hand no matter what the cost or what cards they have. You have decided to take advantage of these stupid players by developing your own poker site that allows you to always beat these players.

The way your site will work is: there are only two players in each game: you and your opponent (this is called a *heads-up* game). Each of the two players will get two “hole cards”; these cards are dealt face down so, presumably, only the player can see them. Four of the five community cards (cards shared by both players) will be dealt face up on the table. We will dispense with any betting for the purpose of this problem: it is irrelevant since each player will always call every bet right to the “river” (the fifth and final community card). You will always be seated at position 1, and your opponent at position 2. The program knows what seat you are sitting at and the hole cards for both players, therefore, it can compute all the cards that remain in the deck that will *maximize* the amount of money you would get playing the hand folding is *not* an option. In the event there is no card that would allow you to win or tie the hand, you will be forced to take a loss on that hand (after all, it looks good if you lose *sometimes*).

Input

The first line of input contains an integer N which is the number of data sets that follow ($1 \leq N \leq 100$). Each data set consists of three lines. The first line of each dataset specifies your two hole cards. The second line is your opponent's hole cards. The third line is the four community cards. A card is specified by two characters: it's rank (A,2,3,4,5,6,7,8,9,T,J,Q,K) and it's suit (S,D,H,C) for Spades, Diamonds, Hearts, and Clubs respectively. For example, the *Ace of Hearts* would be 'AH', the *Nine of Spades* would be '9S'. Within a line, there are no spaces or invalid characters. All input will be valid.

Output

For each data set, your program will output (on a new line) the data set number followed by a colon, followed by a space, and a list of card(s) that will *maximize* the amount of money you could get playing the hand. If no card will allow you to win or tie, just display 'LOSER' instead of the cards. Each card should be separated by a single space. Your program will display up to 15 cards on each output line. If more than one output line is required for a dataset, additional lines should be indented by exactly 3 spaces. Cards should be output in suit order (S,D,H,C) and rank order within each suit (Ace high).

Sample Input

```
3
JD3S
KH5S
KDAC2DQD
2S3S
2C3C
4D5DJHAH
4S7D
```

KHKD
AHADKSKC

Sample Output

1: TS 3D 4D 5D 6D 7D 8D 9D TD AD TH TC
2: 4S 5S 6S 7S 8S 9S TS JS QS KS AS 2D 3D 6D 7D
8D 9D TD JD QD KD AD 2H 3H 4H 5H 6H 7H 8H 9H
TH QH KH 4C 5C 6C 7C 8C 9C TC JC QC KC AC
3: LOSER