2002/2003 SOUTHERN CALIFORNIA REGIONAL
ACM INTERNATIONAL COLLEGIATE PROGRAMMING CONTEST

Problem 5
Cribbage Pegging

Cribbage is a card game for two or more players that has several distinct phases of play, among them are *the play* and *the show*, sometimes called *pegging* and *counting*, respectively. The show is a wonderful example of combinatorics, and makes an excellent programming contest problem. In 1999, the judges of the Southern California Region posed just such a problem, and therefore already have dozens of cribbage hand counting programs to choose from when verifying their hands during a game. To assist the judges in scoring during the play, your team is to write a program that evaluates the points earned for given plays in a deal.

Pegging starts with the player to the dealer's left, who lays down one card face up and announces its value. All face cards (Jack, Queen, and King) are ten points, each number card takes its value, and Ace is always one. Proceeding clockwise, all players take turns playing one card face up, adding its value to the running sum and announcing the sum as they play. The object is twofold: to get the sum as close to 31 without exceeding 31, and to play cards that result in scoring points along the way.

Play continues until the sum hits 31 or no one can play cards that don't exceed 31. At this point, the cards played so far are turned face down and the sum is reset to zero. Pegging resumes using the unplayed cards, starting with the player *following* the one who played the most recent card. During the play, if a player cannot lay down a card (either because the lowest unplayed card in his hand would force the sum to exceed 31, or all his cards are played), that player states "go" and the turn proceeds to the next player. If a player *can* lay down a card without exceeding 31, he *must* play.

Scoring during the play is based on the face-up cards only. Scoring opportunities are:

15	If the running sum hits exactly 15, the player to make 15 pegs two points.
31	If the running sum hits exactly 31, the player to make 31 pegs two points. The *go* and hitting 31 are exclusive.
go	The *go* is named for the point scored by the player who forces all others to state "go." The player who lays down the card that sums closest to 31 (but not hitting 31) when no one else can play scores one point. The *go* and hitting 31 are exclusive.
pair	(Two-of-a-kind) If the player pairs (in rank) the previously played card, the player making the pair scores two points.
pair royal	(Three-of-a-kind) If the player pairs (in rank) the previously played card, which itself formed a pair, the player making the pair royal scores six points.
double pair royal	(Four-of-a-kind) If the player pairs (in rank) the previously played card, which itself formed a pair royal, the player making the double pair royal scores 12 points.
run	If the player lays down a card that, when taken with the previous 2 or more cards, forms an unbroken sequence of three or more consecutive ranks (a *run*), the player scores one point for each card in the run. The actual order of play is not important—only that the previous cards form a run.

Problem 5
Cribbage Pegging (continued)

Input to the program is a list of successive independent (not from the same game) deals. A deal starts with a single integer on a line indicating n, the number of players, followed by all the card plays for the deal, each play per line. Because a cribbage hand contains four cards, a deal must contain $4n$ plays. A card play consists of the player number (1 through n), a single space, a card specification, then end-of-line. Player 1 is always to the dealer's left; the dealer is always player n. A card specification contains two characters: the rank followed by the suit. The rank in ascending order is A=Ace, 2 through 9, T=10, J=Jack, Q=Queen, K=King. The suit is C=Clubs, D=Diamonds, H=Hearts, S=Spades. It is possible that as the sum approaches 31, some players may not be able to play, and that some player might end up playing several cards in a row. The input will have only legal plays. The last deal is terminated by end-of-file.

Output is a list of the results for each deal, one deal per line. Each deal contains the player numbers and their scores, in increasing order of player number. For each player, print the player number, a colon, and that player's total pegged points for the deal. Separate each player's results by a single space. The output lines should *not* contain any trailing spaces. See the Sample Output for examples.

The italicized commentary accompanying the Sample Input on this page does not occur in the actual input data.

Sample Input

```
2
1 JD
2 QH
1 JC    player 2 states "go" and player 1 pegs a one-point go; the sum resets to 0
2 9S
1 KD
2 6C    player 1 states "go" after player 2 plays the Six of Clubs
2 6H    player 2 pairs himself for two points and pegs two more points for hitting 31; sum=0
1 KS    player 2 states "go" (no more cards to play), player 1 pegs a one-point go
4
1 4C
2 6C
3 5D    player 3 gets two points for hitting 15, and pegs three more points for the 4-5-6 run
4 7C    player 4 pegs four points for the 4-5-6-7 run
1 9H    player 1 gets two points for hitting 31; sum=0
2 7D
3 7H    player 3 makes two points for pairing the Seven of Diamonds
4 7S    player 4 pegs six points by forming a pair royal upon the existing pair of Sevens
1 9S    players 2, 3, and 4 state "go", player 1 gets a one-point go; sum=0
2 8C
3 6D
4 3H
1 2H
2 8D    player 3 states "go"
4 2S    players 1, 2, and 3 state "go", player 4 pegs a one-point go; sum=0
3 JS    players 4, 1, and 2 state "go", player 3 pegs a one-point go
```

Sample Output

```
1:2 2:4
1:3 2:0 3:8 4:11
```