



acm International Collegiate Programming Contest



2009 ACM ICPC South Central USA Regional Programming Contest

Know When to Hold Them

Introduction:

In the game of poker, each player holds 5 playing cards in their hand. At the end of the final round each player reveals their cards, and the player with the most valuable hand wins the game. Your goal is to write a program that reads in the poker hands of each player, determines their relative worth, and then prints them back out in sorted order from most to least valuable.

Poker is played with a deck of 52 cards, evenly divided into 4 suits: *hearts* (♥), *spades* (♠), *diamonds* (♦), and *clubs* (♣). Each suit has 13 cards, and each card within the suit has a unique rank that determines the relative worth of the card. From highest to lowest (i.e. from most to least valuable) these ranks are: *ace* (high), *king*, *queen*, *jack*, *10*, *9*, *8*, *7*, *6*, *5*, *4*, *3*, *2*, *ace* (low). Although only one *ace* card exists within each suit, and it is usually ranked as the highest card, under some circumstances the *ace* card will be treated as the lowest. The suit itself has no bearing on the worth of the card and is only used to determine which category a particular hand belongs to (in particular the *flush* and *straight flush* categories; see below).

Each poker hand can be classified into one of nine categories depending on the combination of cards within that hand. A higher category hand always beats any other lower category hand regardless of the individual card ranks within either hand. If both hands belong to the same category, then the individual card ranks are used to decide which hand is more valuable. Note that each category has slightly different rules about how the individual card ranks break the tie. Also note that the category classification does not depend on the permutation of cards within a hand (i.e. the position of individual cards within a hand does not matter). The possible hand categories are listed below from most to least valuable:

1. *Straight Flush*

All cards are of the same suit and have sequential rank. Between two straight flushes, the hand with the highest card wins. Note that the low ace rule applies to this category when a hand consists of the following cards: *5, 4, 3, 2, ace* (i.e. the *ace* has the lowest rank in this hand and the *5* has the highest).

Example:

A♥ K♥ Q♥ J♥ 10♥ beats K♠ Q♠ J♠ 10♠ 9♠ and 6♦ 5♦ 4♦ 3♦ 2♦ beats 5♣ 4♣ 3♣ 2♣ A♣.

2. *Four Of a Kind*

Four cards of the same rank (i.e. a quad) with a fifth card of a different rank. Between two hands in this category, the hand containing the higher ranking quad wins. If both quads have equally ranked cards, then the hand with a higher unmatched card wins. Example:

8♣ 8♥ 8♠ 8♦ Q♣ beats 2♣ 2♥ 2♠ 2♦ A♦ and 4♣ 4♥ 4♠ 4♦ A♥ beats 4♣ 4♥ 4♠ 4♦ J♥.

3. *Full House*

Three matching cards of one rank plus two matching cards (i.e. a pair) of a different rank. Between two full houses, the hand with a higher ranked group of three wins. If both groups of three have equal ranking, then the hand with the higher ranked pair wins. Example:

4♦ 4♥ 4♣ Q♠ Q♣ beats 3♣ 3♠ 3♥ K♦ K♥ and Q♥ Q♦ Q♠ J♦ J♣ beats Q♥ Q♦ Q♠ 10♥ 10♠.

4. *Flush*

All cards are of the same suit. Between two flushes, the hand with the highest card wins. If the highest card in both hands is the same rank, then the second highest card decides the winner, and so on.

Example:

Q♦ 8♦ K♦ J♦ 2♦ beats J♠ 10♠ 9♠ 2♠ 8♠ and Q♥ J♥ 9♥ 8♥ 3♥ beats Q♣ J♣ 7♣ 3♣ 2♣.

5. *Straight*

All cards have sequential rank. Between two straights the hand with the highest card wins. Note that the low ace rule applies to this category when a hand consists of the following cards: 5, 4, 3, 2, *ace*.

Example:

Q♠ J♠ 10♠ 9♥ 8♦ beats 7♥ 6♦ 5♥ 4♠ 3♦ and A♣ K♠ Q♣ J♣ 10♥ beats 5♦ 4♥ 3♣ 2♠ A♥.

6. *Three Of a Kind*

Three matching cards of the same rank plus two unmatched cards (known as the *kickers*) whose rank doesn't match each other or the group of three. Between two such hands, the hand with a higher group of three wins. If both groups of three have equal rank, then the hand with the highest kicker wins (if the highest kicker cards in both hands are equal then compare the other two kickers). Example:

7♥ 7♣ 7♠ J♦ 10♠ beats 4♥ 4♠ 4♦ 7♣ Q♦ and 8♠ 8♥ 8♦ 3♠ 6♦ beats 8♣ 8♦ 8♠ 3♦ 5♥.

7. *Two Pair*

Two matching cards of one rank, two matching cards of a different rank (which match each other but not the first pair), and a fifth card (the *kicker*) whose rank does not match either pair. Between hands in this category, the hand with the higher ranked pair wins. If the highest pairs in each hand have equal ranking, then compare using the other pairs in each hand. If both pairs in one hand match both pairs in the other hand, then the hand with the highest kicker (the fifth unmatched card) wins. Example:

K♥ K♣ 3♦ 3♣ 2♠ beats J♦ J♥ Q♠ Q♣ K♠ and 8♠ 8♣ 5♦ 5♣ 3♥ beats 8♥ 8♦ 4♠ 4♦ 3♣.

8. *One Pair*

One matching pair of cards with the same rank, plus three cards (*kickers*) whose rank matches neither the pair nor each other. Between these categories, the hand with the highest pair wins. If both hands have equally ranked pairs, then compare using the highest unmatched kicker card (or second highest, or third highest). Example:

4♥ 4♦ 8♥ 10♦ J♠ beats 3♥ 3♠ Q♣ 8♥ 2♦ and 9♠ 9♦ 10♥ J♥ 4♠ beats 9♥ 9♣ J♠ 7♥ 8♦.

9. *High Card*

If a poker hand does not fall into any other category above, then it is classified as this category. In other words, each card is of a different rank, all the cards are not of the same suit, and are not in ranked sequence. Between two hands in this category, the hand with the highest ranked card wins (or second highest, or third highest, and so on). Example:

8♣ K♠ 3♥ J♦ 2♥ beats 10♥ J♠ 8♣ 7♥ 6♦ and A♠ 3♦ 4♥ Q♥ J♠ beats K♠ 8♣ Q♥ 10♦ 9♥.

Input:

Input to this problem will begin with a line containing a single integer N ($1 \leq N \leq 100$) indicating the number of data sets. Each data set will consist of a line with an integer M ($1 \leq M \leq 100$) specifying the number of poker hands in this data set, followed by a description of the cards in each player's hand with each hand specified on a separate line. Each playing card is described by a two character string, and a poker hand consists of five such strings separated by spaces:

1. The first character specifies the card's rank as follows:

- The letters "A", "K", "Q", and "J" respectively specifying *ace*, *king*, *queen*, and *jack*.
- A "0" (number zero) specifying a card of rank *10*.
- The numbers "2" through "9" specifying a card of the same numeric rank.

2. The second character gives the card's suit with "H", "S", "D", and "C" respectively specifying *hearts*, *spades*, *diamonds*, and *clubs*.

You may assume that in the input the same card (i.e. having the same rank and suit) will never appear more than once within a single poker hand.

Output:

For each data set in the input, output the heading "Data Set #*k*" where *k* is 1 for the first data set, 2 for the second, etc. Following the heading, output the poker hands (one per line) from the input in sorted order from the most to least valuable. You may assume that no two poker hands in the input have the same relative worth, therefore a unique sorted order will exist for each data set. The individual cards within each hand must be printed in the same order as read in from the input.

Sample Input:

```
1
4
AS 3C 3S JC AH
7C 4C 8C 6C 5C
QD 2H 8S 9C QH
0S KS 2S 6S 7S
```

Sample Output:

```
Data Set #1
7C 4C 8C 6C 5C
0S KS 2S 6S 7S
AS 3C 3S JC AH
QD 2H 8S 9C QH
```

The statements and opinions included in these pages are those of the Hosts of the ACM ICPC South Central USA Regional Programming Contest only. Any statements and opinions included in these pages are not those of Louisiana State University or the LSU Board of Supervisors.

© 1999, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009 ACM ICPC South Central USA Regional Programming Contest