Poker Hand Sorter (Engineer role)

A poker hand consists of a combination of five playing cards, ranked in the following ascending order (lowest to highest):

Rank	Combination	Description
1	High card	Highest value card
2	Pair	Two cards of same value
3	Two pairs	Two different pairs
4	Three of a kind	Three cards of the same value
5	Straight	All five cards in consecutive value order
6	Flush	All five cards having the same suit
7	Full house	Three of a kind and a Pair
8	Four of a kind	Four cards of the same value
9	Straight flush	All five cards in consecutive value order, with the same suit
10	Royal Flush	Ten, Jack, Queen, King and Ace in the same suit

The cards are valued in the order:
2, 3, 4, 5, 6, 7, 8, 9, 10, Jack, Queen, King, Ace*

** For this exercise, Ace is considered high only. (i.e. cannot be used as a low card below 2 in a straight).*

Suits are:

Diamonds (D), Hearts (H), Spades (S), Clubs (C)

When multiple players have the same ranked hand then the rank made up of the highest value cards wins. For example, pair of kings beats a pair of queens, and a straight with a high card of Jack beats a straight with high card of nine.

If two ranks tie, for example, if both players have a pair of Jacks, then highest cards in each hand are compared; if the highest cards tie then the next highest cards are compared, and so on.

For example:

Hand	Player 1	Player 2	Winner
1	4H 4C 6S 7S KD Pair of Fours	2C 3S 9S 9D TD Pair of Nines	Player 2
2	5D 8C 9S JS AC Highest card Ace	2C 5C 7D 8S QH Highest card Queen	Player 1
3	2D 9C AS AH AC Three Aces	3D 6D 7D TD QD Flush with Diamonds	Player 2
4	4D 6S 9H QH QC Pair of Queens Highest card Nine	3D 6D 7H QD QS Pair of Queens Highest card Seven	Player 1
5	2H 2D 4C 4D 4S Full House With Three Fours	3C 3D 3S 9S 9D Full House with Three Threes	Player 1

Note - suits are not taken into account to break a tie for this exercise - only the value of the card determines a winner.

Your task

You are to build a command line program that takes, via STDIN, a "stream" of hands for a two player poker game. At the completion of the stream, your program should print to STDOUT the number of hands won by Player 1, and the number of hands won by Player 2.

Input

Each line read via STDIN will be a set of 10 cards. Each card is represented by 2 characters - the value and the suit. The first 5 cards in the line have been dealt to Player 1, the last 5 cards in the line belong to Player 2.

For example:

```
AH 9S 4D TD 8S 4H JS 3C TC 8D
|--Player 1--| |--Player 2--|
```

Output

At the completion of the stream into STDIN (EOF), the output of your file (in STDOUT) must clearly state how many hands Player 1 won, and how many hands Player 2 won. For example:

```
Player 1: 10 hands
Player 2: 12 hands
```

You will be provided a test file you can use to determine whether your solution is working correctly. The expected result for this file is:

Expected output of provided test file

```
Player 1: 263
Player 2: 237
```

You must provide....

- A URL of the Git/Mercurial/SVN repository where the source code to your solution can be found (e.g. a Github or Bitbucket repository)
- Instructions on how to build and execute your solution

You can write your solution in any language, but Java (or another JVM language) is preferred.

Example of how your tool could be executed:

```
$ cat poker-hands.txt | java -jar my-poker-solution.jar
```