Poker Hands

A standard deck of playing cards consists of 52 cards. These are divided into four suits called clubs, diamonds, hearts, and spades. Each suit consists of 13 different cards numbered 1 to 13. Card 1 may also be called Ace, card 11 may also be called Jack, card 12 may also be called Queen, and card 13 may also be called King.

For the purpose of this problem the following is valid *input*:

> A poker hand consisting of a single line with five different cards, separated by single separators.

A separator may be any one of: − / <space> (with only one separator *type* allowed to be used in one hand).

A card consists of a number immediately followed by a letter to indicate suit (c, d, h, s, C, D, H, S). Cards 1, 10, 11, 12, and 13 may alternatively be represented by a letter (a, t, j, q, k, A, T, J, Q, K).

The order of cards is determined first by their rank:
$$2 < 3 < 4 < 5 < 6 < 7 < 8 < 9 < 10 < 11 < 12 < 13 < 1,$$
and for cards of equal rank the suit ordering is
$$C < D < H < S.$$

The *standard output* format is:

> Five cards in increasing order on a single line, separated by single spaces, with each card consisting of:
>
> - A number for cards 2–10, but a capital letter for cards 1, 11, 12 and 13 (A, J, Q, and K respectively).
>
> - A capital letter to indicate suit (C, D, H, or S).

Task

Write a program that performs the following tasks for each line of input:

1. Reads the line of input.

2. Checks whether the input is a valid poker hand.

3. If the input is valid, outputs the poker hand in standard format, otherwise outputs `Invalid:` (there is a single space after the colon) followed by the input.

For example these two lines of input:

```
6s/3d/KC/13S/Ad
```

```
hello
```

should produce output like:

```
3D 6S KC KS AD
Invalid: hello
```

The input/output requirements for this étude are **strict**. That means that I will simply be using `diff` (or equivalent) to compare the output of your program to the correct output. You must work alone on your program, but you should feel free to share test data etc. with others (indeed, this is encouraged).

Relates to Objectives

1.1, 1.2, 1.3, 2.2, 2.7, 2.8, 3.4, 3.5, 4.1, 4.5

(2, Individual)