

Assignment 1

CSE 2031 3.0 Software Tools, Fall 2012, Section E

Due: Thursday, October 25, 2012, 5pm.

Format: Individual.

Specification

Your task is to create a program that reads a sequence of bridge hands from standard input. For each valid bridge hand, your program must print out the number of points in the hand (see below). If the input is not a valid hand, an appropriate message must be displayed before exiting. In more detail:

A valid bridge hand consists of 13 playing cards. A playing card is represented by a single character (possible values in descending order are A,K,Q,J,T,9,8,7,6,5,4,3,2). Each card also has a suit (possible values are S,H,D,C - the four suits are spades, hearts, diamonds, and clubs). The correct input format is shown in the following example:

```
SAJ43
HQJT
DKT6
C432
```

Each valid bridge hand must span four lines of input, one for each suit. The order of the suits must be exactly as shown above. The cards in each suit must also be ordered in descending value as in the example. There is no whitespace between two cards, between the suit and the first card, before the suit, or after the last card. Only upper case characters are acceptable. You can assume that all input lines end with a newline character.

For each 4 lines of input, your program must produce one line of output. The output starts with `Hand X:` , where X is the number of hands read so far including the current one, followed by one of the following two options:

1. The number of points, a SPACE, and the word `points` (if the hand is valid)
2. An error message (if the hand is invalid)

The possible error messages are listed below. If more than one type of error occurs in a single hand, your program can output any one of the appropriate error messages. As soon as an error occurs, your program must stop executing, i.e. do not try to process any further bridge hands.

- `Invalid card: ?` - If one of the cards is not valid, the program must output the unrecognized card in place of the question mark. Example: `Invalid card: B`
- `Invalid suit: ?` - If one of the suits is not valid, the program must output the unrecognized suit in place of the question mark. Example: `Invalid suit: J`
- `Incorrect suit` - If the suits are not in the correct order, or if there is a duplicate suit.
- `Incorrect card` - If the cards are not in descending order, or there is a duplicate card.

- Wrong number of cards: ? - If the input does not contain exactly 13 cards, the program must output the number of cards in place of the question mark.
Example: Wrong number of cards: 14
- Incomplete hand - If there are less than 4 lines of input remaining.

If no error occurs, i.e. the input is a valid bridge hand, the program must print the number of points in the hand, as in:

```
Hand 3: 11 points
```

There must be single SPACE characters between the different parts of the output, and no SPACES before or after it. The output must be followed by a single newline character. There must not be any empty lines in your output.

Points are calculated as follows: An ace (represented by A) is worth 4 points, a king (K) 3 points, a queen (Q) 2 points, and a jack (J) 1 point. Other cards do not contribute any points. Also, if the hand contains 5 cards in a particular suit, e.g. 5 spades, an extra point should be added. A suit that contains 6 cards is worth 2 points, a 7-card suit is 3 points etc...

Submission

Name your program `a1.c`. Before submitting, make sure that it compiles and runs correctly in the lab. Once ready, submit with

```
submit 2031 A1 a1.c
```

Grading

This assignment will be marked automatically. This means:

1. There will be no marks for comments, meaningful identifier names, etc. However, you should still follow appropriate style guidelines for your own benefit.
2. Submissions that do not compile will receive an automatic F.
3. The output produced by your program must match the expected output exactly, otherwise you will receive an F. Even whitespace matters. The course webpage contains a sample input file (`a1.in`) and the corresponding output file (`a1.out`). You must execute your program providing `a1.in` as the input. Assuming the output of your program is in `myoutput.out`, the following command must produce no output:

```
diff a1.out myoutput.out
```

If there is output, you are certain to receive an F for this assignment.

4. Passing the `diff` test above is only a necessary condition for getting a good grade, not a sufficient one. Your program will be tested with lots of different input. It has to behave according to the specification in every case. You must test your program with any possible input you can think of to ensure it works correctly.
5. You must work on this assignment on your own. Submissions will be tested for similarity. Suspicious cases will be reported to the Faculty of Science and Engineering for plagiarism.