

Programming Task

We've created a simple multiplayer card game where:

- 7 players are dealt 5 cards from two 52 card decks, and the winner is the one with the highest score.
- The base score for each player is calculated by adding the highest 3 card values for each player, where:
 - The number cards have their numerical value.
 - Face card values are J = 11, Q = 12, K = 13
 - A = 11 (not 1).
- In the event of a tie, the scores are recalculated for only the tied players by calculating a "suit score" for each player to see if the tie can be broken (it may not).
 - Each card is given a value based on its suit, with diamonds = 1, hearts = 2, spades = 3 and clubs = 4, and the player's score is calculated as the suit value of the player's highest card.

You are required to write a command line application using C# or JavaScript (Node application) that needs to do the following:

- Run on Windows.
- Be invoked with the name of the input and output text files.
- Read the data from the input file, find the winner(s) and write them to the output file.
- Handle any problems with the input or input file contents.

How to Submit Code

Please just send us an email with **this pdf** and a **single file** named `addemupgame.cs` or `addemupgame.js` file attached.

Your application will be tested using an automated test runner, so **your application must run to completion without any user input beyond the initial command parameters.**

JavaScript

It will be run from console as follows:

```
node addemupgame.js --in abc.txt --out xyz.txt
```

C#

It will be compiled and the resultant exe will be run from console as follows:

```
addemupgame.exe --in abc.txt --out xyz.txt
```

Command Parameters

- The command parameters can be in any order.
- The file names can vary.
- It can be assumed that if they are not located in the same directory as the application, that the path will be supplied as part of the file name.

```
--in abc.txt --out xyz.txt
```

Input File Format

- The input file will contain 7 rows, one for each player's hand of 5 cards.
- Each row will contain the player's name followed by a colon then a comma separated list of the 5 cards.
- Each card will consist of the face value and the suit (H = Hearts, S = Spades, C = Clubs and D = Diamonds).
 - Example: KD = King of Diamonds, 4C = Four of Clubs.
- Input is not case-sensitive.
- Spaces can be ignored.
- Do not make assumptions about the correctness of the input.

Output File Format

The output file should contain a single line, with **one** of the following 3 possibilities:

- The name of the winner and their score separated by a colon. (Base Value only if there's a clear winner OR Base + Suit Value if there was a broken tie)
 - Example:

```
Player1:35
```

- A comma separated list of winners in the case of a tie that can't be broken and their score separated by a colon. (Base + Suit Value)
 - Example:

```
Player1,Player2:38
```

- "Exception:[reason]" if the input file or its contents had any issues (validate the input).
 - Example:

```
Exception:<Some nice reason why the input is wrong.>
```

Examples

Example Input

Player1:QC,2C,7S,10S,6C
Player2:2C,6D,3H,3C,6S
Player3:QD,3C,AC,2H,2S
Player4:5D,6C,JC,KS,4H
Player5:3S,9D,5S,9H,4H
Player6:6H,KD,4C,5H,7D
Player7:10C,10C,QD,JS,3D

Example Output

Player7:33

Example Input

Player1:KH,QD,9H,8H,7S
Player2:QC,6S,KH,10D,8D
Player3:2D,2S,5C,5C,6D
Player4:KC,8S,JS,10S,2S
Player5:4H,9H,JH,4D,2H
Player6:4H,AD,KD,AD,2C
Player7:9S,JH,QH,6C,AC

Example Output

Player2:37

Example Input

Player1:8D,8S,3C,8S,7C
Player2:KH,QH,AC,10C,AC
Player3:AS,7S,9S,7D,10S
Player4:4C,AH,QD,KH,8C
Player5:3S,KC,2D,5H,2S
Player6:JC,JS,3C,AH,10H
Player7:9D,7S,6C,10D,4D

Example Output

Player2,Player4:38