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 up all 5 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 hearts = 1, spades = 2, clubs = 3 and diamonds = 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 `game.cs` or `game.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 game.js --in abc.txt --out xyz.txt
```

C#

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

```
game.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

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

Example Output

```
Player1:50
```

Example Input

```
Player1:3S,5C,5S,5C,2S
Player2:9C,QH,7D,JH,9C
Player3:5D,KH,QH,8H,8D
Player4:10S,3C,6H,2H,AC
Player5:7S,KC,7C,10D,AS
Player6:7H,4D,KH,AS,2D
Player7:3D,8S,2D,JD,2H
```

Example Output

```
Player5:51
```

Example Input

```
Player1:3C,3C,2S,4H,KH
Player2:QD,8D,7D,6C,2H
Player3:10H,9D,8H,AH,KS
Player4:JC,JC,KS,8C,8D
Player5:JD,5D,6D,5S,AD
Player6:QD,AC,KD,3S,JH
Player7:6S,4C,8S,KD,4S
```

Example Output

```
Player3,Player4:51
```