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 score for each player is calculated by adding up all 5 card values for each player, where the number cards have their face value, J = 11, Q = 12, K = 13 and 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 score based on its suit, with hearts = 1, spades = 2, clubs = 3 and diamonds = 4, and the player's score is 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.

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 Structure

- The input file will contain 7 rows, one for each player's hand of 5 cards.
- Each row will contain the player's name separated 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.
- Input is not case-sensitive.
- Spaces can be ignored.

Example input:

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

Output File Structure

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

- The name of the winner and their score (Face Value OR Face + Suit Value) (colon separated).
    - Example: **`Player1:35`**
- A comma separated list of winners in the case of a tie and the score (Face Value OR Face + Suit Value) (colon separated).
    - Example: **`Player1,Player2:35`**
- "Exception:[reason]" if the input file or its contents had any issue.
    - Example: **`Exception:Some reason why the input is wrong.`**