Assignment 02 (Due: Friday, March 9, 2018, 11 : 59 : 00PM Central Time)

CSCE 322

THIS ASSIGNMENT IS ONLY WORTH 10% OF YOUR FINAL GRADE.

1 Instructions

In this assignment, you will be required to write JavaScript functions that simplify playing of the variation of basic train Dominoes.

1.1 Data File Specification

An example of properly formatted file is shown in Figure 1. The first file encodes the hands, the second file encodes the trains that are to be changed during every move.

test.hands.doc

```
8,11;0,9;2,3;9,12;5,8;4,5;4,8;1,11;11,12;3,7;5,10;5,12;4,12;10,12;7,10;0,0;1,2
1,6;3,5;6,9;4,4;5,5;10,10;2,4;5,9;2,10;9,11;9,10;0,10;4,7;3,6;7,11;2,8;6,10
0,12;2,2;0,6;3,11;3,3;7,9;0,8;1,7;0,3;3,12;7,8;0,11;1,12;7,7;5,7;2,5;1,10
5,6;8,12;6,11;7,12;5,11;8,10;0,7;2,9;4,10;3,8;0,1;2,7;10,11;6,8;1,4;11,11;6,12
1,5;3,4;1,8;4,11;3,9;0,4;8,9;0,2;6,7;4,9;4,6;3,10;1,9;2,6;1,3;0,5;2,11
```

test.trains.doc

```
12,12
1,1
6,6
9,9
8,8
```

Figure 1: A properly formatted game encoding

2 One Player, One Move

The first part (`onePlayerOneMove` in the file `csce322Homework02Part01.js`) will take in one (1) argument (a train) and return a function that takes in one (1) argument (a hand), and returns the train that is the result of making a move based on the follow rules (in order of precedence).

1. Move from First to Last Domino in Hand
2. Place the Domino as is on the Front of the Train
3. Place the Domino flipped on the Front of the Train

4. Place the Domino as is on the Back of the Train
5. Place the Domino flipped on the Back of the Train

A train is valid when neighboring dominoes share the same number for touch halves.

```
3,11;2,6;1,10;4,5;0,0;1,6;4,8;7,10;2,9;3,9
2,4;0,9;9,9;3,4;2,10;0,6;7,11;1,9;1,2;6,7
5,7;0,12;2,11;10,12;4,6;0,3;6,6;6,10;2,3;9,11
0,5;0,11;1,8;6,12;3,7;1,7;1,11;6,11;5,10;5,9
5,12;10,11;4,9;1,12;4,7;2,5;2,12;4,12;4,10;0,8
0,2;6,8;3,8;9,12;5,6;1,4;7,9;0,4;3,6;1,5
7,8;4,11;8,10;5,11;5,8;9,10;7,12;3,3;0,7;7,7
3,12;1,3;8,9;3,10;8,11;8,12;0,1;2,7;2,8;3,5

2,2
1,1
10,10
5,5
4,4
8,8
12,12
11,11
```

Figure 2: Before `onePlayerOneMove`

```
[ [ 6, 2 ], [ 2, 2 ] ]
```

Figure 3: After `onePlayerOneMove`

3 One Player, Many Moves

The second part (`onePlayerManyMoves` in the file `csce322Homework02Part02.js`) will take in one (1) argument (a train) and return a function that takes in one (1) argument (a hand), and returns the train that is the result of attempting to place as many dominoes from the hand as possible into the train. If the hand has already been emptied, or the train cannot be extended, the train is returned unchanged. The same rules of precedence apply.

```
0,9;2,11;6,9;1,2;4,7;5,12;3,5;4,9;5,11;4,5;2,8;11,12
1,7;2,12;6,7;1,8;4,8;0,0;5,10;10,11;3,11;8,9;2,3;2,9
2,10;0,6;6,10;11,11;2,5;1,11;1,10;9,11;8,8;0,5;0,1;6,8
4,6;0,12;4,10;7,11;1,4;0,3;1,3;8,11;8,10;0,2;3,9;1,12
5,6;6,11;9,12;0,11;0,10;9,10;1,6;1,9;3,12;10,12;9,9;4,11
5,9;2,7;7,8;6,12;5,7;8,12;3,6;2,4;3,4;2,6;2,2;5,5
7,10;5,8;4,12;7,9;7,12;3,10;0,7;3,8;3,7;0,4;0,8;1,5

7,7
4,4
1,1
12,12
10,10
3,3
6,6
```

Figure 4: Before `onePlayerManyMoves`

```
[ [ 0, 9 ], [ 9, 4 ], [ 4, 7 ], [ 7, 7 ] ]
```

Figure 5: After `onePlayerManyMoves`

4 Many Players , One Move

The third part (`manyPlayersOneMove` in the file `csce322Homework02Part03.js`) will take in one (1) argument (an array of trains) and return a function that takes in one (1) argument (an array of hands), and returns the game that is the result of each player making one move . The same rules of precedence apply.

```
2,3;4,5;2,10;6,12;11,12;0,2;6,8;4,11;6,6;0,12;8,11;5,5;4,4;0,4
1,3;1,4;1,9;2,11;0,7;5,7;1,2;3,5;1,6;3,10;3,4;3,9;7,12;1,11
2,7;4,9;8,12;4,6;10,10;9,12;5,11;0,11;7,11;0,8;2,6;8,9;5,6;5,8
1,5;2,9;2,8;3,6;4,10;0,6;9,11;1,12;10,11;0,10;7,10;5,10;4,7;7,9
7,8;5,12;5,9;0,9;6,10;1,10;1,7;10,12;9,10;6,7;3,8;6,9;6,11;9,9
3,11;3,7;2,4;0,1;0,5;2,12;4,8;3,3;3,12;8,10;8,8;1,8;0,3;4,12

11,11
12,12
2,2
7,7
0,0
1,1
```

Figure 6: Before `manyPlayersOneMove`

```
[ [ [ 12, 11 ], [ 11, 11 ] ],
  [ [ 7, 12 ], [ 12, 12 ] ],
  [ [ 7, 2 ], [ 2, 2 ] ],
  [ [ 10, 7 ], [ 7, 7 ] ],
  [ [ 9, 0 ], [ 0, 0 ] ],
  [ [ 0, 1 ], [ 1, 1 ] ] ]
```

Figure 7: After `manyPlayersOneMove`

5 Many Players , Many Moves

The fourth part (`manyPlayersManyMoves` in the file `csce322Homework02Part04.js`) will take in one (1) argument (an array of trains) and return a function that takes in one (1) argument (a list of hands), and returns the game that is the result of each player making a move in turn until someone has emptied their hand or nobody can add to their train. The same rules of precedence apply.

```
8,12;2,9;1,8;9,11;9,12;3,9;8,10;0,5;1,11;3,10;7,12;4,5
5,6;2,12;10,11;6,9;7,11;0,12;4,8;5,8;3,5;6,7;11,12;6,12
1,5;6,10;12,12;2,7;7,7;6,6;8,11;4,10;3,6;1,10;6,8;10,12
1,1;0,6;1,7;4,12;2,3;9,10;2,10;0,1;1,3;3,12;4,6;8,9
0,2;2,6;3,4;1,12;5,7;5,10;0,7;0,9;5,11;3,7;0,11;0,3
7,8;4,9;6,11;5,12;7,9;1,2;2,5;7,10;1,9;4,7;0,4;4,11
1,4;3,11;5,9;0,0;0,10;2,11;0,8;2,4;1,6;9,9;2,8;3,8

3,3
11,11
5,5
2,2
4,4
8,8
10,10
```

Figure 8: Before `manyPlayersManyMoves`

```
[ [ [ 2, 9 ], [ 9, 3 ], [ 3, 3 ], [ 3, 10 ], [ 10, 8 ], [ 8, 12 ] ],
  [ [ 10, 11 ], [ 11, 11 ], [ 11, 7 ], [ 7, 6 ], [ 6, 5 ], [ 5, 8 ]
   ],
  [ [ 3, 6 ], [ 6, 6 ], [ 6, 10 ], [ 10, 1 ], [ 1, 5 ], [ 5, 5 ] ],
  [ [ 1, 1 ], [ 1, 3 ], [ 3, 2 ], [ 2, 2 ], [ 2, 10 ], [ 10, 9 ] ],
  [ [ 10, 5 ], [ 5, 7 ], [ 7, 3 ], [ 3, 4 ], [ 4, 4 ] ],
  [ [ 7, 4 ], [ 4, 9 ], [ 9, 7 ], [ 7, 8 ], [ 8, 8 ] ],
  [ [ 2, 8 ], [ 8, 0 ], [ 0, 0 ], [ 0, 10 ], [ 10, 10 ] ] ]
```

Figure 9: After `manyPlayersManyMoves`

6 Naming Conventions

Your files should follow the naming convention of `csce322Homework02Part01.js`, `csce322Homework02Part02.js`, `csce322Homework02Part03.js`, and `csce322Homework02Part04.js`.

6.1 `helpers.js`

A file named `helpers.js` has been provided with the functionality to read the `.doc` files into numerical matrices. If a modified `helpers.js` file is not included with your submission, the default will be used in its place.

7 webgrader Note

Submissions will be tested with `node.js`, note the browser. `cse.unl.edu` is currently running version 4.8.7 of `node`.

8 Point Allocation

Component	Points
`csce322Homework02Part01.js`	
Test Cases	1×20
Total	20
`csce322Homework02Part02.js`	
Test Cases	1×20
Total	20
`csce322Homework02Part03.js`	
Test Cases	1×30
Total	30
`csce322Homework02Part04.js`	
Test Cases	1×30
Total	30
Total	100

9 External Resources

JavaScript Tutorial