HW12 Problem Set

CS-3160

Assignment
HANDWRITTEN – NONE.
PROGRAM – Due at midnight, via Blackboard, at midnight on day prior to due date.

Programming Portion

NOTE: For ALL of these programs, you are to use Racket

Program 15-A

Write a program that allows two human players to play a game of NIM. The board can be hardcoded with the format defined by the following examples:

```
(define board1 '((X)(X X X) (X X X X X) (X X X X X X X)))
(define board2 '((X X X X) (X X X) (X X X X X)))
(define board3 '(() (X X X X) (X X X) () (X X X X X) ()))
(define board4 '(() () () () () ()))
```

Each board is a list of lists. Each list defines one row. The contents of each list is one X for every stick in that row.

The top level function should be

```
(playNim board players)
```

Where **board** is a game board as defined above and players is a list consisting of the type of player, which is one of {**human**, **random**, **smart**}. The first player in the list is Player #1 who makes the first move. For instance, a particular game might be executed as follows:

```
(playNim board1 ‘(human smart))
```

In this problem, you do not need to implement either the **random** or the **smart** player types.

The game should follow this basic outline:

```
WHILE: Game NOT over
   DO:
      1) Display the current configuration of the board.
      2) Ask either Player #1 or Player #2 for their move.
         1) Ask, and get, the row number.
         2) Ask, and get, the number of sticks to remove from that row.
      3) Validate the proposed move to determine if it is legal.
   UNTIL: Legal move is entered
   4) Make the specified move.
 PRINT: Game-over message indicating which player won.
```

HW12 Problem Set

CS-3160

Program 15-B

Implement two additional player types. The first, `random`, is a random player that makes random legal moves. The second, `smart`, is a smart player that makes the optimal move to ensure a win (if such a move is possible) and otherwise makes a random move.

The appearance to someone watching the screen should be as though a player is actually playing. In other words, it should appear as though the player is being prompted and is entering their moves. You do not need to insert delays to reinforce this illusion, just make it so that someone reviewing the console output can clearly see the history of moves by all players.

GENERAL NOTES

You must write the functions recursively and you may not use the function `set!` or, in general, any predefined function that uses an exclamation point in the name (these functions usually violate referential transparency for the sake of performance).

Your console output might look something like:

```
1) X X
2) X X X X
3)
4) X X X
Player #2:
Which row do you wish to remove sticks from? 4
How many sticks do you wish to remove? 2

1) X X
2) X X X X
3)
4) X
Player #1:
Which row do you wish to remove sticks from? 2
How many sticks do you wish to remove? 4
```

Grading Rubric
The assignment is worth 25 pts (as a whole) and the score will be recorded as a percentage of that amount.

	Handwritten							Programming			
Problem								15-A	15-B		
Points								15	10		

10% Physical Format
50% Answers correct (and supported by work)
40% Effort evidenced by the submitted work