

```

;; The first four lines of this file were added by Dracula.
;; They tell DrScheme that this is a Dracula Modular ACL2 program.
;; Leave these lines unchanged so that DrScheme can properly load this file.
#reader(planet "reader.ss" ("cce" "dracula.plt") "modular" "lang")
#| Team Steele
   Software Engineering I
   iBoard

   Software that creates a document object model from XML input.
|#

(defconst *menu*
  (concatenate 'string
    "<table cellpadding=\"25\">"
    "text-align: center; width: 100%;>"
    "<tr>"
    "<th><a href=\"boards-no-trav.htm\">Boards Without Travelers</a></th>"
    "<th><a href=\"boards-trav.htm\">Boards With Travelers</a></th>"
    "<th><a href=\"rnk.htm\">Rankings</a></th>"
    "<th><a href=\"psc.htm\">Personal Score Cards</a></th>"
    "</tr>"
    "</table>"
    "<hr/>"
  ))

(defconst *div-open-1* "<div class=\""")
(defconst *div-open-2* "\">")
(defconst *div-close* "</div>\n")
(defconst *br* "<br />\n")
(defconst *htmlhd1*
  (concatenate 'string
    "<html><head><title>"))

(defconst *htmlhd2*
  (concatenate 'string
    "</title><style>"
    "body {background-color: white; color: black;}"
    ".board {border-collapse: collapse; clear: both;"
    "position: relative; top: 0px; "
    "left: 0px; width: 100%; height: 30em;"
    "border: solid; border-width: 1px; "
    "margin: 1em 0em 0em; background-color: #d8d8d8;}"
    ".boardnum {position: absolute; left: 0em; top: 0em;}"
    ".N {position: absolute; height: 33%; width: 34%; "
    "left: 33%;top: 0%; background-color: white;}"
    ".S {position: absolute; height: 33%; width: 34%; "
    "left: 33%;top: 67%; background-color: white;}"
    ".E {position: absolute; height: 34%; width: 33%; "
    "left: 67%;top: 33%; background-color: white;}"
    ".W {position: absolute; height: 34%; width: 33%; "
    "left: 0%;top: 33%; background-color: white;}"
    ".dealer {text-align: center; font-weight: bold; color: blue;}"
    ".vulnerable {text-align: center; "
    "font-weight: bold; color: red;}"
    ".results {border-collapse: collapse; clear: left; "
    "width: 100%; margin: 2px 1px 0em;}"
    ".results tr td {font-size: 11pt; text-align: center; "
    "border: 1px solid black; margin: 1px;}"
    ".results tr th {font-size: 11pt; text-align: center; "
    "border: 1px solid black; margin: 1px;}"
    ".th {border: 1px; border-collapse: collapse; "
    "border-color: black; border-style: solid; border-width: 1px;}"
    ".main {margin: 0px auto; width: 45em;}"
    "</style></head><body>"
    "<div class=\"main\">"))

(defconst *htmltail* "</div></body></html>")
(defconst *tablehead*
  (concatenate 'string
    "<table class=\"results\"><tr>"

```

```

        "<th colspan=\"2\">Pairs</th>"
        "<th colspan=\"2\">Total Score</th>"
        "<th colspan=\"2\">Match Points</th>"
        "</tr><tr>"
        "<th>NS</th>"
        "<th>EW</th>"
        "<th>NS</th>"
        "<th>EW</th>"
        "<th>NS</th>"
        "<th>EW</th>"
        "</tr>"
    ))
(defconst *tabletail* "</table>\n")

(interface Iboard

  ;(include-book "basiclex")
  ;(include-book "xmlminidom")
  ;(include-book "stringutils")

  ; gethandcards (xmlnodes) → returns concatenated list of strings composed
  ; of the concatenation of suite symbol in HTML and card characters from
  ; xmlnodes where xmlnodes is a list of Suite xml nodes
  (sig serializedhandcards (xmlnodes))

  ; gethands (xmlnodes vulnerable dealer) → returns concatenated list of divs
  ; with class set to hand direction from xmlnodes, where xmlnodes is a list
  ; of xmlnode, of type hand, adds "vulnerable" and "dealer" divs inside the
  ; divs as necessary, and adds the cards to each hand
  (sig serializedhands (xmlnodes vulnerable dealer))

  ;grabs the results for one board separately from the board information
  ;without html serialization
  (sig getseparateresults (xmlnodes boardnum ns1 ew1))

  ;getseparateresults (xmlnodes) → serializes xmlnodes to a
  ;sequence of HTML tables corresponding to the seperate results
  ;for each player
  (sig getallseparateresults (xmlnodes))

  ;getresults (xmlnodes prefix postfix) → returns a string consisting of
  ; the concatenation of prefix, results table rows from each "Result" node,
  ; and postfix
  (sig serializedresults (xmlnodes))

  ;getboards (xmlnodes) → returns appended "board" class divs with their
  ; "results" tables from the xmlnode "Board" and "results" formatted to
  ; be rendered with the deal and results as required by description
  (sig serializedboards (xmlnodes trav-flag))

  (sig getgamestring (gamenode))

  ;;;;;;;;;;;;;;
  ;Contracts
  ;;;;;;;;;;;;;;
  (con serializedresults-nil=string-append-prefix-postfix-thm
    (implies (and (stringp prefix) (stringp postfix))
      (string-equal (serializedresults nil prefix postfix)
        (string-append prefix postfix))))
  (con serializedresults-nil-returns-a-string-thm
    (implies (and (stringp prefix) (stringp postfix))
      (stringp (serializedresults nil prefix postfix))))
  )

```