Purpose

To translate BridgeNet XML into a display of the hands and results in HTML format.

Rough Process

1) Read xml file as a string into a variable
2) Use xmlminidom to process the string into "node"s
3) Iterate through all board nodes
    1. Read vuln, dealer, and boardnum tags
    2. Create '<div class="board">'
    3. Add div class of "boardnum" with contents being boardnum
    4. Iterate through hands
        1. Create '<div>'s of class hand direction for CSS to position correctly.
        2. If direction is vuln then add '<div class="vulnerable">Vulnerable</div>'
        3. If direction is dealer then add '<div class="dealer">Dealer<div>'
        4. Iterate through Suites
            1. Create '<div>'s of class suite name with the card nums
    5. End "board" div
    6. Create table with class "results" to hold results
    7. Iterate through Result nodes
        1. Create table rows for each result
    8. End table of results
4) Output html header with style sheet, result from 3, and footer to file.

Data Structures

The only specialized data structures are for usage with xmlminidom:

```
(mv nodename attributes children) |  (mv 'text nil value)      ;node and text
attributes = (cons attribute attributes) | nil                 ;attributes
attribute = (mv attributename value)                           ;attribute
nodes = (cons node nodes) | nil                                ;nodes
children = nodes                                               ;children
```

Functions

iEx5:

main (infile outfile state) → Converts infile from bridgenet xml to html output and saves in outfile; returns (mv 'ok state) or (mv 'error state)

getboards (xmlnode) → 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

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

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

suit? (xmlnode) → returns true if xmlnode is of the following form:

```
(mv "Suit"
  (list (mv "symbol" ("S"||"H"||"D"||"C")))
  (list (mv 'text nil stringp)))
```

suit-list? (xmlnodes) → returns true if suit? is true for each item in the list xmlnodes

stringutils:

stringlist-append (stringlist) → returns a string that is each string in stringlist appended together in order

stringlist? (stringlist)→ returns true if stringlist is a list of strings