

```

==== tennis.clj: ====
(ns tennisgame.core)

(def new-game [0 0])

(defn score [game player]
  (let [[score-a score-b] game]
    (if (= :A player)
        [(inc score-a) score-b]
        [score-a (inc score-b)])))

(def points {0 "Love"
             1 "Fifteen"
             2 "Thirty"
             3 "Fourty"})

(defn result
  "Returns a string representation of the game."
  [[score-a score-b]]
  (cond
    (= score-a score-b) (if (>= score-a 3)
                           "Deuce"
                           (str (points score-a) " all"))
    (and (> score-a 3) (= score-a (inc score-b))) "Advantage A"
    (and (> score-b 3) (= score-b (inc score-a))) "Advantage B"
    (and (> score-a 3) (> score-a (inc score-b))) "Game A"
    (and (> score-b 3) (> score-b (inc score-a))) "Game B"
    :else (str (points score-a) " " (points score-b))))

```