```
// A program that implements a simple simulation of High-Card for 4 players.

Include
{
        "stdlib/stdlib.cgl";
}

CardEntities
{
        dealer;
        player0;
        player1;
        player2;
        player3;
}

Globals
{
        var players;
}

Start // Deal cards, set chips
{
        var i;
        var e;

        << "Hello and Welcome to PCGSL Highcard!";

        #players = [$player0, $player1, $player2, $player3];

        << "Shuffling deck";
        shuffle($dealer);
        << "Dealing Cards";
        i = 0;
        while (i < |#players|) {
              e = #players[i];

              // Deal five cards to the player.
              e <- $dealer[0];
              e <- $dealer[0];
              e <- $dealer[0];
              e <- $dealer[0];
              e <- $dealer[0];

              // Print out the player's hand.
              << "Player " ^ i ^ " hand: " ^ e[0] ^ " " ^ e[1] ^ " " ^ e[2]
                 ^ " " ^ e[3] ^ " " ^ e[4];

              i++;
        }
}

Play
{
}

WinCondition
{
        var comp;
        var highplayer;
        var highcard;
        var card1;
        var card2;
        var card3;
        var card4;

        card1 = high_card($player0);
        << "Player 0 high card : " ^ card1;
        card2 = high_card($player1);
        << "Player 1 high card : " ^ card2;
        card3 = high_card($player2);
        << "Player 2 high card : " ^ card3;
        card4 = high_card($player3);
        << "Player 3 high card : " ^ card4;

        comp = card_compare(card1, card2);
        if (comp > 0) {
           highplayer = $player0;
           highcard = card1;
        } else {
           highplayer = $player1;
           highcard = card2;
        }
        comp = card_compare(highcard, card3);
        if (comp < 0) {
           highplayer = $player2;
           highcard = card3;
        }
        comp = card_compare(highcard, card4);
        if (comp < 0) {
           highplayer = $player3;
           highcard = card4;
        }

        << "The winner is: " ^ highplayer;
        return [highplayer];
}

// Simply returns the highest value card (by value, and ties broken by suit)
// of a player. Assumes exactly 5 cards in the player's card pile.
high_card(var e)
{
        var comp;
        var card;

        comp = card_compare(e[0], e[1]);
        if (comp > 0) {
           card = e[0];
        } else {
           card = e[1];
        }
        comp = card_compare(card, e[2]);
        if (comp < 0) {
           card = e[2];
        }
        comp = card_compare(card, e[3]);
        if (comp < 0) {
           card = e[3];
        }
        comp = card_compare(card, e[4]);
        if (comp < 0) {
           card = e[4];
        }

        return card;
}

// Returns 1 if c1 is higher, -1 if c1 is lower, 0 if equal.
card_compare(var c1, var c2) {
  var s1;
  var s2;
  var f1;
  var f2;

  s1 = cardsuit(c1);
  s2 = cardsuit(c2);
  f1 = cardface(c1);
  f2 = cardface(c2);
  if (f1 > f2) {
    return 1;
  }
  if (f1 < f2) {
    return 0 - 1;
  }
  if (s1 > s2) {
    return 1;
  }
  if (s1 < s2) {
    return 0 - 1;
  }
  return 0;
}
```