```
      case '0':           exit(EXIT_SUCCESS);
      case '2':           rank = 0; break;
      case '3':           rank = 1; break;
      case '4':           rank = 2; break;
      case '5':           rank = 3; break;
      case '6':           rank = 4; break;
      case '7':           rank = 5; break;
      case '8':           rank = 6; break;
      case '9':           rank = 7; break;
        case 't': case 'T': rank = 8; break;
        case 'j': case 'J': rank = 9; break;
        case 'q': case 'Q': rank = 10; break;
        case 'k': case 'K': rank = 11; break;
        case 'a': case 'A': rank = 12; break;
        default:            bad_card = true;
      }

      suit_ch = getchar();
      switch (suit_ch) {
        case 'c': case 'C': suit = 0; break;
        case 'd': case 'D': suit = 1; break;
        case 'h': case 'H': suit = 2; break;
        case 's': case 'S': suit = 3; break;
        default:            bad_card = true;
      }

      while ((ch = getchar()) != '\n')
        if (ch != ' ') bad_card = true;

      if (bad_card)
        printf("Bad card; ignored.\n");
      else if (card_exists[rank][suit])
        printf("Duplicate card; ignored.\n");
      else {
        num_in_rank[rank]++;
        num_in_suit[suit]++;
        card_exists[rank][suit] = true;
        cards_read++;
      }
    }
  }

/**********************************************************
 * analyze_hand: Determines whether the hand contains a   *
 *               straight, a flush, four-of-a-kind,       *
 *               and/or three-of-a-kind; determines the   *
 *               number of pairs; stores the results into *
 *               the external variables straight, flush,  *
 *               four, three, and pairs.                  *
 **********************************************************/
void analyze_hand(void)
{
  int num_consec = 0;
  int rank, suit;
```