

```

import java.util.ArrayList;
import java.util.Arrays;
import java.util.Scanner;

public class Agram {

    public static void main(String args[]) {
        Scanner input = new Scanner(System.in);
        System.out.print("Input sequence (comma-separated): ");
        String sequence = input.nextLine();
        sequence = sequence.trim();

        String leadCard = sequence.substring(0, 2);

        String[] dealerCards = sequence.split(", ", 0);

        for (int i = 1; i < dealerCards.length; i++) {
            dealerCards[i-1] = dealerCards[i];
        }

        String result = PlayCard(leadCard, dealerCards);
        System.out.println(result);
    }

    public static String PlayCard(String leadCard, String[]
dealerCards) {

        String order = "A23456789TJQK";
        String leadCardSuit = leadCard.substring(1, 2);
        String leadCardValue = leadCard.substring(0, 1);

        // Case 1: There exist cards with same suit

        ArrayList<String> cardsSameSuit = new ArrayList<>();

        for (int i = 0; i < dealerCards.length; i++) {

            if(dealerCards[i].substring(1, 2)
.equals(leadCardSuit)) {
                cardsSameSuit.add(dealerCards[i]);
            }
        }
    }
}

```

```

}

// Iterate through dealer cards with same suit

String lowestValueGreaterThanLead = "";
String lowestValueSmallerThanLead = "";

if (cardsSameSuit.size() != 0) {
    for (int j = 0; j < cardsSameSuit.size(); j++) {

        if (order.indexOf(cardsSameSuit
            .get(j).substring(0, 1)) >
            order.indexOf(leadCardValue)) {

            // if the testing card is greater than opponent
            card

                if
(lowestValueGreaterThanLead.isEmpty()){

                    lowestValueGreaterThanLead =
                    cardsSameSuit.get(j).substring(0,
1);

                    // sets card if empty "first card"

                } else if (order.indexOf(cardsSameSuit
                    .get(j).substring(0, 1)) <
                    order.indexOf(lowestValueGreaterThanLead))
                {

                    lowestValueGreaterThanLead =
                    cardsSameSuit.get(j).substring(0,
1);

                    // if the new card is less than the
                    previous cards, sets that value (at
                    at this point, it is greater than
                    opp. yet smallest)

                }

            } else {

```

```

        if (lowestValueSmallerThanLead.isEmpty()){

            lowestValueSmallerThanLead =
                cardsSameSuit.get(j).substring(0, 1);

        } else if (order.indexOf(cardsSameSuit
            .get(j).substring(0, 1)) <
            order.indexOf(lowestValueSmallerThanLead))
        {

            lowestValueSmallerThanLead =
                cardsSameSuit.get(j).substring(0,
                    1);

            // if new card is less than the
            // previous cards, sets that value
            // (smallest card overall)

        }
    }

    if (lowestValueGreaterThanOrEqualToLead != "") {

        return lowestValueGreaterThanOrEqualToLead +
            leadCardSuit;

    } else {

        return lowestValueSmallerThanLead +
            leadCardSuit;

    }

}

// Case 2: No cards with same suit

else {

    String lowestCard = "";
    String suitOrder = "CDHS";

    for (int i = 0; i < dealerCards.length; i++) {

```

```
if (lowestCard.isEmpty()) {  
  
    lowestCard = dealerCards[i];  
  
} else if (order.indexOf(dealerCards[i]  
.substring(0, 1)) <= order.indexOf(lowestCard  
.substring(0, 1))) {
```

```
        if (order.indexOf(dealerCards[i]
            .substring(0, 1)) <
            order.indexOf(lowestCard
                .substring(0, 1))) {

                lowestCard = dealerCards[i];

        } else if (suitOrder
            .indexOf(dealerCards[i].substring(1, 2)) <
            order.indexOf(lowestCard.substring(1,
2))) {

                lowestCard = dealerCards[i];

        }

    }

}

return lowestCard;

}

}
```