```
// Victor Narvaez
import java.util.Scanner;
public class PokerHands {
    public static void main(String[] args) {
        int[] hand = new int[5];
        Scanner scanner = new Scanner(System.in);
        while (true) {
            System.out.println();
            System.out.println("Enter five numeric cards, no face cards. Use 2 -
9.");
            for (int i = 1; i <= 5; i++) {
                System.out.print("Card " + i + ": ");
                hand[i - 1] = scanner.nextInt();
                System.out.println();
            }
            if (containsStraight(hand)) {
                System.out.println("Straight!");
                break;
            }
            if (containsFullHouse(hand)) {
                System.out.println("Full House!");
                break;
            }
            if (containsFourOfaKind(hand)) {
                System.out.println("Four of a kind!");
                break;
            }
            if (containsThreeOfaKind(hand)) {
                System.out.println("Three of a kind!");
                break;
            }
            if (containsTwoPair(hand)) {
                System.out.println("Two Pair!");
```

```
                break;
            }
            if (containsPair(hand)) {
                System.out.println("Pair!");
                break;
            }
            System.out.println("High Card!");
        }
    }
/**
* Private method for transforming input array. Each position of returned array
*contains amount of cards in hand.
* So, if card with  value "2" appears in hand 3 times, then value of returned
array *with position 0 content "3",
* or if card with value "5" appears in hand 2 times, then value of returned
array *with position 3 content "2".
 */
    private static int[] handTransform(int hand[]) {
        int[] tHand = new int[8];
        for (int i=0; i<5; i++) {
            tHand[hand[i]-2] += 1;
        }
        return tHand;

    }


    public static boolean containsPair(int hand[]) {
        int[] tHand = handTransform(hand);
        for (int i=0; i<8; i++)
            if (tHand[i]>=2) return true;
        return false;
    }
```

```
public static boolean containsTwoPair(int hand[]) {
    boolean isOnePair = false;
    int[] tHand = handTransform(hand);
    for (int i=0; i<8; i++) {
        if (tHand[i] >= 2) {
            if (isOnePair) return true;
            else isOnePair = true;
        }
    }
    return false;
}

public static boolean containsThreeOfaKind(int hand[]) {
    int[] tHand = handTransform(hand);
    for (int i=0; i<8; i++)
        if (tHand[i]>=3) return true;
    return false;
}

public static boolean containsStraight(int hand[]) {
    int[] tHand = handTransform(hand);
    for (int i=0; i<8; i++)
        if (tHand[i]==1) {
            if (i>3) return false;
            for (int j=i+1; j<i+5; j++) {
                if (tHand[j]!=1) return false;
            }
            return true;
        }
```

```
        return false;
    }

    public static boolean containsFullHouse(int hand[]) {
        boolean isPair = false;
        boolean isThree = false;
        int[] tHand = handTransform(hand);
        for (int i=0; i<8; i++) {
            if (tHand[i]==2) isPair = true;
            if (tHand[i]==3) isThree = true;
        }
        return isPair & isThree;
    }

    public static boolean containsFourOfaKind(int hand[]) {
        int[] tHand = handTransform(hand);
        for (int i=0; i<8; i++)
            if (tHand[i]>=4) return true;
        return false;
    }


}
```