```java
// Author: Aashay Pawar
// NUID: 002134382

package edu.northeastern.csye6200;

import java.util.Scanner;

public class LAB3_P1 {
    public static void main(String[] args) {
        // TODO: write your code here
        System.out.print("Enter a credit card number as a long integer: ");
        try (Scanner sc = new Scanner(System.in)) {
            long cardNo = sc.nextLong();
            boolean check = isValid(cardNo);
            if(check)
                System.out.println(cardNo + " is valid");
            else
                System.out.println(cardNo + " is invalid");
        }
    }

    /** Return true if the card number is valid */
    public static boolean isValid(long number) {
        // TODO: write your code here
        int size = getSize(number);
        boolean pValue = (prefixMatched(number, 4) || prefixMatched(number, 5) || prefixMatched(number, 37) || prefixMatched(number, 6));
        int evenSum = sumOfDoubleEvenPlace(number);
        int oddSum = sumOfOddPlace(number);
        if((size >= 13 && size <= 16) && (pValue) && (pValue) && ((evenSum+oddSum)%10 == 0))
            return true;
        return false;
    }

    /** Get the result from Step 2 */
    public static int sumOfDoubleEvenPlace(long number) {
        // TODO: write your code here
        String mString = number+"";
        int evenSum = 0;
        for(int i = mString.length()-2; i>=0; i-=2)
            evenSum += getDigit(Integer.parseInt(mString.charAt(i) + "")*2);
        return evenSum;
    }

    /**
```

```
45      * Return this number if it is a single digit, otherwise,
  return the sum of
46      * the two digits
47      */
48     public static int getDigit(int number) {
49         // TODO: write your code here
50         if(number < 9)
51             return number;
52         return number/10 + number%10;
53     }
54
55     /** Return sum of odd place digits in number */
56     public static int sumOfOddPlace(long number) {
57         // TODO: write your code here
58         String mString = number+"";
59         int oddSum = 0;
60         for(int i = mString.length()-1; i>=0; i-=2)
61             oddSum += Integer.parseInt(mString.charAt(i) + "");
62         return oddSum;
63     }
64
65     /** Return true if the digit d is a prefix for number */
66     public static boolean prefixMatched(long number, int d) {
67         // TODO: write your code here
68         long n = getPrefix(number, getSize(d));
69         if(n == d)
70             return true;
71         return false;
72     }
73
74     /** Return the number of digits in d */
75     public static int getSize(long d) {
76         // TODO: write your code here
77         String mString = d + "";
78         return mString.length();
79     }
80
81     /**
82      * Return the first k number of digits from number. If the
  number of digits
83      * in number is less than k, return number.
84      */
85     public static long getPrefix(long number, int k) {
86         // TODO: write your code here
87         if(getSize(number) > k) {
88             String mString= number + "";
89             return Long.parseLong(mString.substring(0, k));
90         }
91         return number;
```

```
92     }
93 }
94
```