```java
import java.util.Scanner;
import java.util.concurrent.ThreadLocalRandom;

public class Lab2p1 {
    public static void main(String[] args)
    {
        int choice;
        Scanner sc = new Scanner(System.in);
        do {
            System.out.println("Perform the following methods:");
            System.out.println("1: multiplication test");
            System.out.println("2: quotient using division by subtraction");
            System.out.println("3: remainder using division by subtraction");
            System.out.println("4: count the number of digits");
            System.out.println("5: position of a digit");
            System.out.println("6: extract all odd digits");
            System.out.println("7: quit");
            choice = sc.nextInt();
            switch (choice) {
                case 1: /* add mulTest() call */
                    mulTest();
                    break;
                case 2: /* add divide() call */
                    System.out.println("Enter m");
                    int m = sc.nextInt();
                    System.out.println("Enter n");
                    int n = sc.nextInt();
                    System.out.println(m+"/"+n+"="+divide(m, n));
                    break;
                case 3: /* add modulus() call */
                    System.out.println("Enter m");
                    int a = sc.nextInt();
                    System.out.println("Enter n");
                    int b = sc.nextInt();
                    System.out.println(a+"%"+b+"="+modulus(a, b));
                    break;
                case 4: /* add countDigits() call */
                    System.out.println("Enter number n");
                    int c = sc.nextInt();
                    System.out.println("n: "+c+" - count = "+countDigits(c));
                    break;
                case 5: /* add position() call */
                    System.out.println("Enter n");
                    int d = sc.nextInt();
                    System.out.println("Enter digit");
                    int digit = sc.nextInt();
                    System.out.println("Position: "+position(d, digit));
                    break;
                case 6: /* add extractOddDigits() call */
                    System.out.println("Enter n");
                    int e = sc.nextInt();
                    if(e<0)
                    {
                        System.out.println("oddDigits = Error input!!");
                    }
                    else
                    {
                        System.out.println("oddDigits = "+extractOddDigits(e));
                    }
                    break;
                case 7: System.out.println("Program terminating...");
            }
        } while (choice < 7);
    }
    /* add method code here */
    //1
    public static void mulTest()
    {
        Scanner sc = new Scanner(System.in);

        int i = 0;
        int correct = 0;
        int total = 0;
```

```
74            int ans = 0;
75
76            while(i<5)
77            {
78                int rand1 = ThreadLocalRandom.current().nextInt(1, 9 + 1);
79                int rand2 = ThreadLocalRandom.current().nextInt(1, 9 + 1);
80                System.out.println("How much is "+rand1+" times "+rand2+"? ");
81
82                 ans = sc.nextInt();
83                 total = rand1*rand2;
84
85                if(ans==total)
86                {
87                    correct++;
88                }
89                i++;
90            }
91            System.out.println(correct + " answers out of 5 are correct.\n");
92        }
93
94        //2
95        public static int divide(int m, int n) {
96
97            int q = 0;
98
99            while(m>=n)
100           {
101               m = m - n;
102               q++;
103           }
104           return q;
105       }
106
107       //3
108       public static int modulus(int m, int n)
109       {
110           int q = 0;
111
112           if(m<n)
113           {
114               return m;
115           }
116           else
117           {
118               while(m>=n)
119               {
120                   m = m - n;
121               }
122           }
123           return m;
124       }
125
126       //4
127       public static int countDigits(int n)
128       {
129           int count = 0;
130
131           if(n<0)
132           {
133               System.out.println("Negative number error!");
134           }
135           else
136           {
137               while(n!=0)
138               {
139                   n = n/10;
140                   count++;
141               }
142           }
143           return count;
144       }
145
146       //5
```

```java
    public static int position(int n, int digit)
    {

        int pos = 0;

        while(n!=0)
        {
            if(n%10 == digit)
            {
                pos++;
                break;
            }
            else
            {
                n = n/10;
                pos++;
            }

        }
        if(n==0)
        {
            pos = -1;
            System.out.println("Error input!!");
        }
        return pos;
    }

    //6
    public static long extractOddDigits(long n) {

        long newNum = 0;
        int odd = 0;
        long temp = n;
        int count = 0;
        int p = 0;

        while(temp!=0)
        {
            temp = temp/10;
            count++;
        }
        count = count-1;
        p = (int) Math.pow(10, count);

        while (n != 0)
        {

            if (((n / p) % 2) == 1)
            {
                newNum = newNum * 10;
                newNum = newNum + (n / p);
                n = n % p;
                p = p / 10;
                odd++;
            }
            else {
                n = n % p;
                p = p / 10;
            }

        }
        if (odd == 0)
        {
            newNum = -1;
        }
        return newNum;
    }
}
```