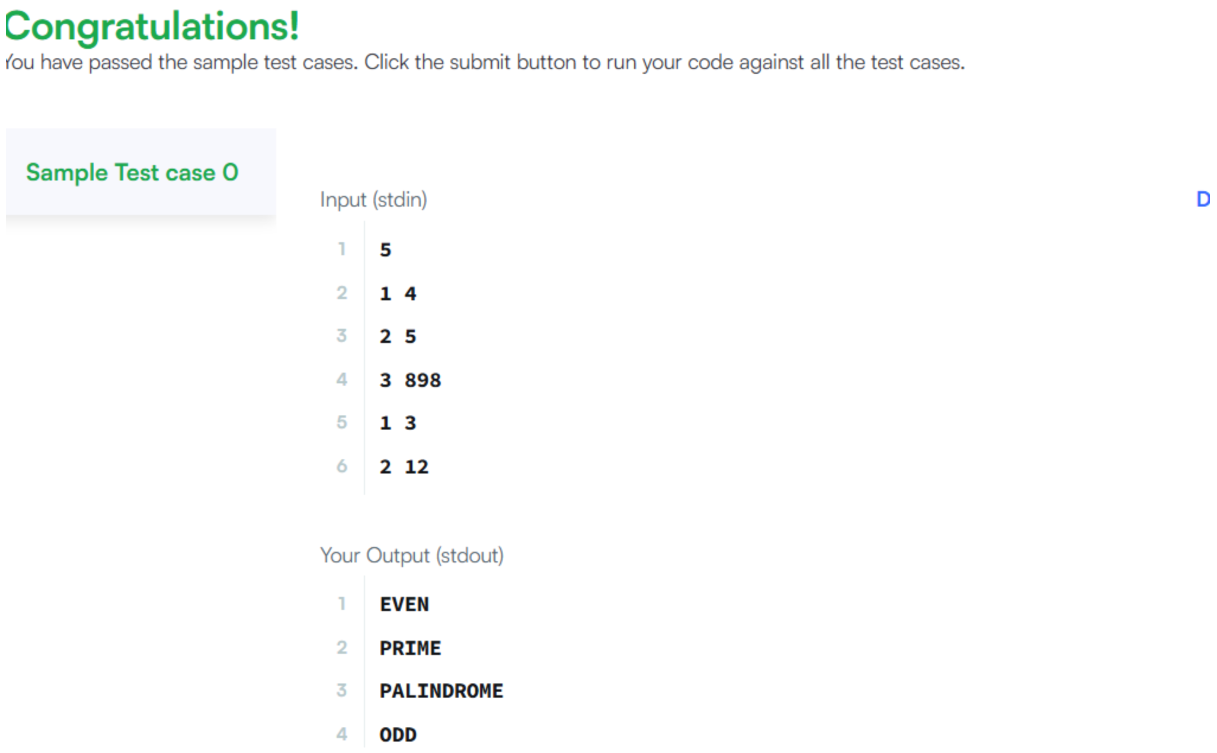
WEEK 2

```
import java.util.*;

interface PerformOperation {

 boolean check(int a);

}

class MyMath {

 public static boolean checker(PerformOperation p, int num) {

  return p.check(num);

 }


    // Logic to check ODD or EVEN

    public static PerformOperation isOdd() {

        return a -> a % 2 != 0;

    }


    public static PerformOperation isPrime() {

        return a -> {

            if (a <= 1) return false;

            for (int i = 2; i <= Math.sqrt(a); i++) {

                if (a % i == 0)

                    return false;

            }

            return true;

        };

    }


    public static PerformOperation isPalindrome() {

        return a -> {

            int original = a, reverse = 0;

            while (a > 0) {

                reverse = reverse * 10 + a % 10;
                a /= 10;
            }
            return original == reverse;
        };
    }
}
```



Congratulations!

You have passed the sample test cases. Click the submit button to run your code against all the test cases.

Sample Test case 0

Input (stdin)

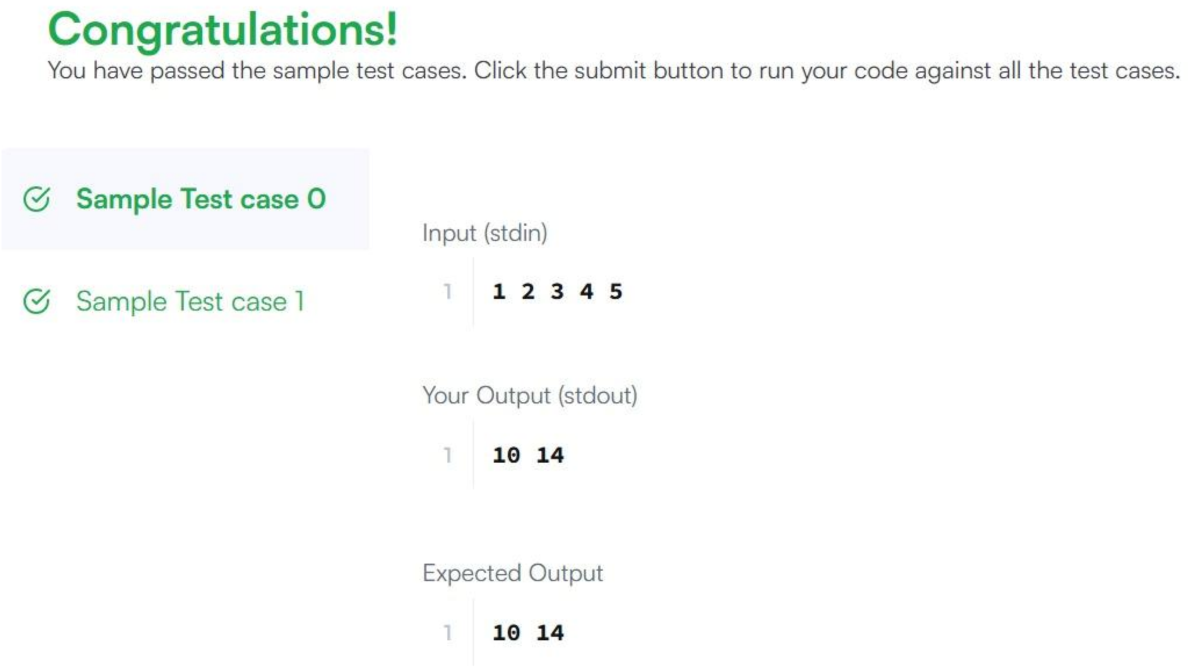
```
5
1 4
2 5
3 898
1 3
2 12
```

Your Output (stdout)

```
EVEN
PRIME
PALINDROME
ODD
```

TASK2.2:

```
import java.util.*;
public class Solution {
    public static void miniMaxSum(int[] arr) {
        long totalSum = 0;
        for (int i = 0; i < arr.length; i++) {
            totalSum += arr[i];
        }
        long minSum = totalSum - arr[0];
        long maxSum = totalSum - arr[0];
        for (int i = 1; i < arr.length; i++) {
            long sumExcluding = totalSum - arr[i];
            if (sumExcluding < minSum) {
                minSum = sumExcluding;
            }
            if (sumExcluding > maxSum) {
                maxSum = sumExcluding;
            }
        }
        System.out.println(minSum + " " + maxSum);
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int[] arr = new int[5];
        for (int i = 0; i < 5; i++) {
            arr[i] = sc.nextInt();
        }
        miniMaxSum(arr);
    }
}
```



Congratulations!

You have passed the sample test cases. Click the submit button to run your code against all the test cases.

Input (stdin)

```
1 2 3 4 5
```

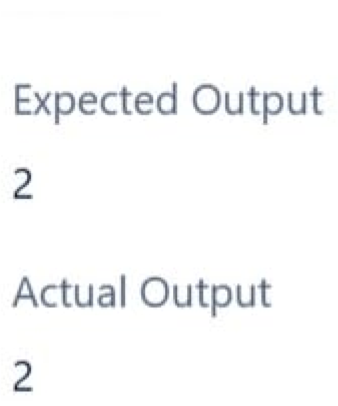
Your Output (stdout)

```
10 14
```

Expected Output

```
10 14
```

```
public int isPalindrome(String input1)
{
   String str = input1.toLowerCase();
   int left = 0;
   int right = str.length() - 1;
   while (left < right {
    if (str.charAt(left) != str.charAt(right))
      {
         return 1; // Not a palindrome
      }
      left++;
      right--;
   }return 2; // Palindrome
}
```



Expected Output

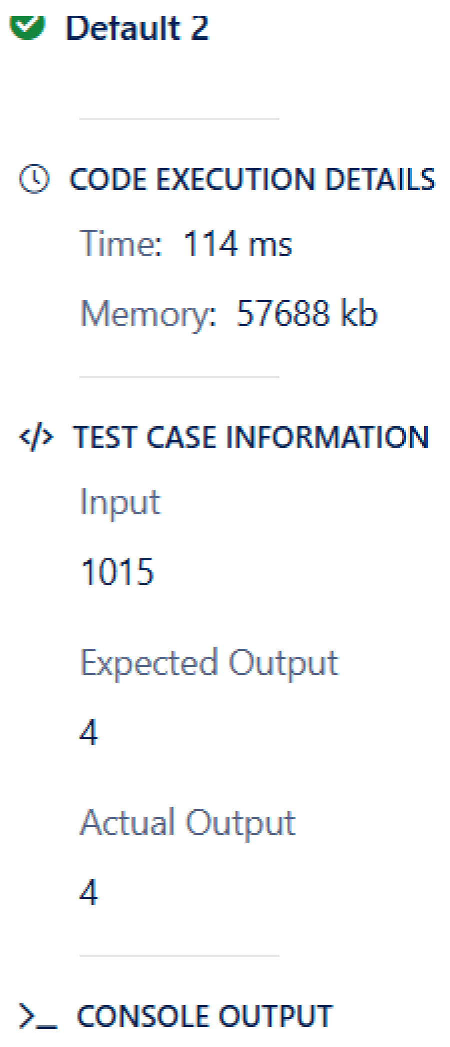
2

Actual Output

2

TASK2.3:

```
public int allDigitsCount(int input1)
{
   int count = 0;


   while (input1 ){
    count++;
       input1 = input
 return count;}
```



Default 2

CODE EXECUTION DETAILS

Time: 114 ms

Memory: 57688 kb

TEST CASE INFORMATION

Input

```
1015
```

Expected Output

```
4
```

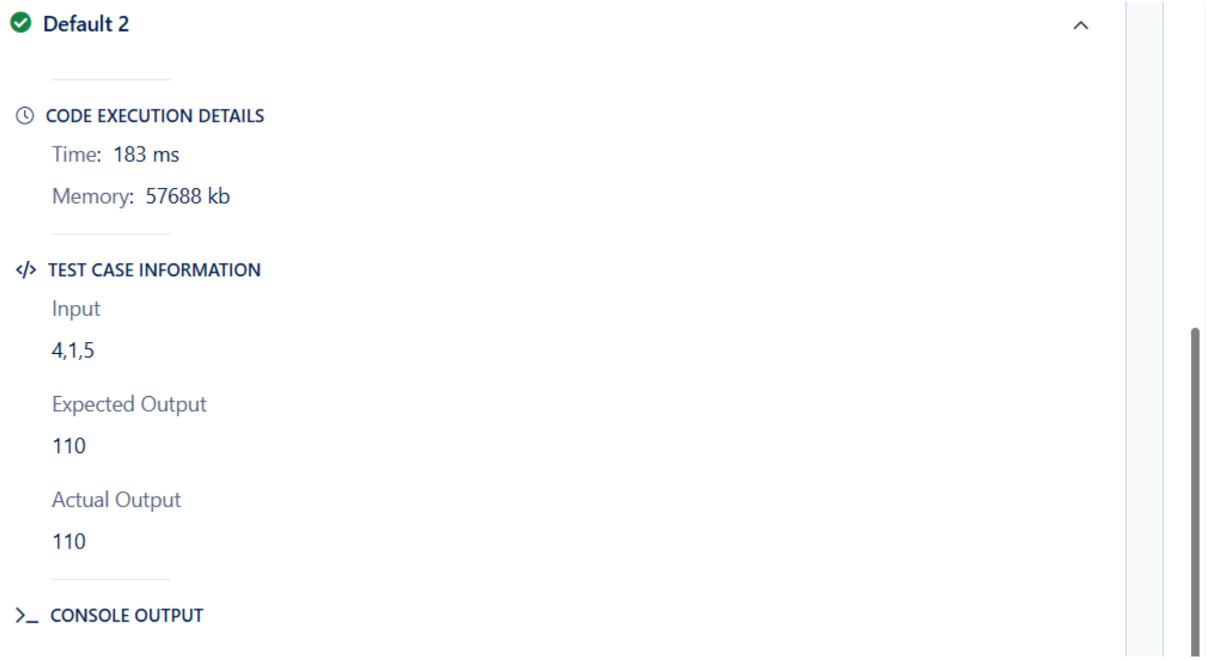
Actual Output

```
4
```

CONSOLE OUTPUT

TASK2.4:

```
public int totalHillWeight(int input1, int input2, int input3) {
   int totalWeight = 0;
 for (int level = 1; level <= input1; level++) {
      int stars = level;
      int weightPerStar = input2 + (level - 1) * input3;
      totalWeight += stars * weightPerStar;
   }

   return totalWeight;
}
```



Default 2

CODE EXECUTION DETAILS

Time: 183 ms

Memory: 57688 kb

TEST CASE INFORMATION

Input

```
4,1,5
```

Expected Output

```
110
```

Actual Output

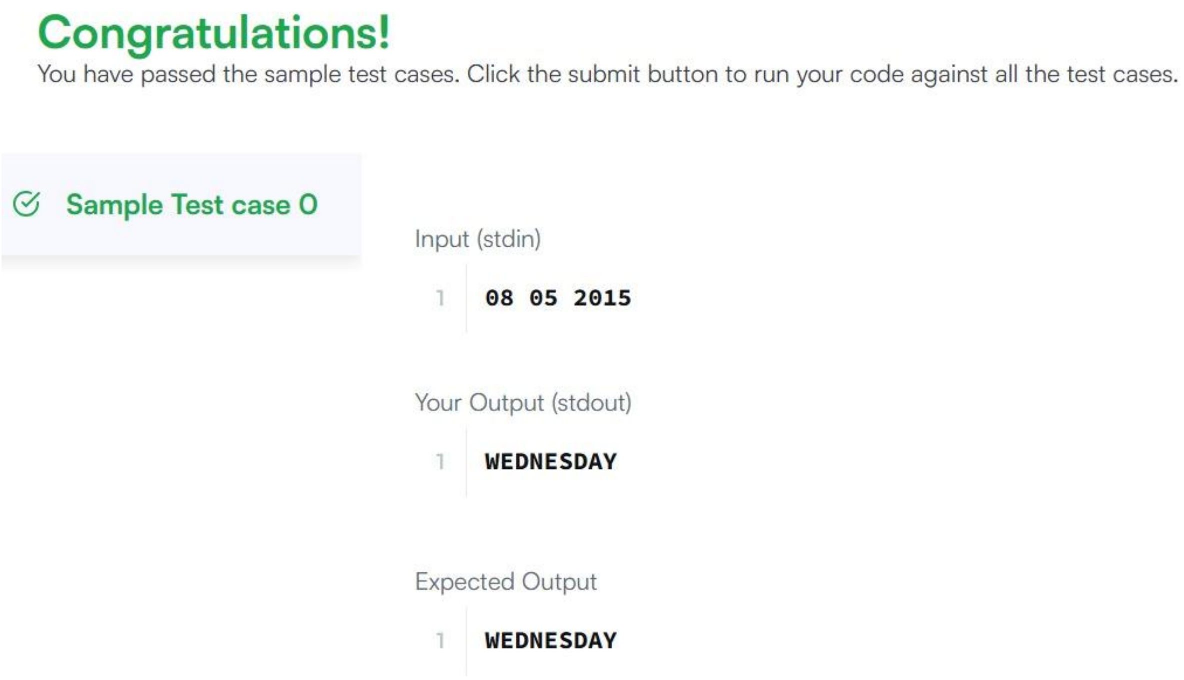
```
110
```

CONSOLE OUTPUT

TASK2.5:

```
import java.io.*;

import java.math.*;

import java.security.*;

import java.text.*;

import java.util.*;

import java.util.concurrent.*;

import java.util.function.*;

import java.util.regex.*;

import java.util.stream.*;

import static java.util.stream.Collectors.joining;

import static java.util.stream.Collectors.toList;


class Result {


    public static String findDay(int month, int day, int year) {


        Calendar cal = Calendar.getInstance();
        cal.set(year, month - 1, day);

        int dayOfWeek = cal.get(Calendar.DAY_OF_WEEK);

        String[] days = {
            "SUNDAY", "MONDAY", "TUESDAY",
            "WEDNESDAY", "THURSDAY", "FRIDAY", "SATURDAY"
        };

        return days[dayOfWeek - 1];
    }

    public static void main(String[] args) throws Exception {

        Scanner sc = new Scanner(System.in);

        int month = sc.nextInt();
        int day = sc.nextInt();
        int year = sc.nextInt();

        String result = findDay(month, day, year);

        System.out.println(result);

        sc.close();
    }
```

```
 }
```



Congratulations!

You have passed the sample test cases. Click the submit button to run your code against all the test cases.

Input (stdin)

```
08 05 2015
```

Your Output (stdout)

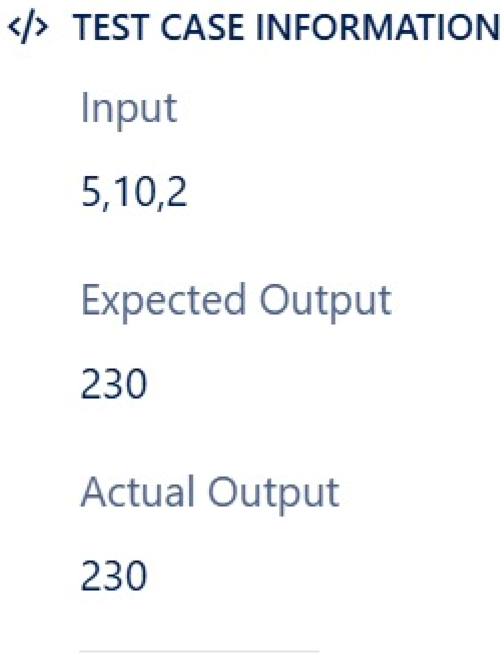
```
WEDNESDAY
```

Expected Output

```
WEDNESDAY
```

TASK2.6:

```
public int sumOfSumsOfDigits(int input1) {
int total = 0;
 int run = 0;

 char[] arr = String.valueOf(input1).toCharArray();

 for (int i = 0; i < arr.length; i++) {
   run = run + (arr[i] - '0');
   total = total + run;
 }

 return total;
}
```



</> TEST CASE INFORMATION

Input

5,10,2

Expected Output

230

Actual Output

230