WEEK 2

AIM : To use Java lambda expressions with a functional interface to check whether a number is odd or even, prime or composite, and a palindrome or not.

ALGORITHM :

1. Define a functional interface with a single method to perform a check on a number.
2. Create lambda expressions to check odd or even, prime or composite, and palindrome or not.
3. Use a method to apply the selected lambda expression to the given number.
4. Read the required input values.
5. Execute the appropriate lambda expression.
6. Display the result.
7.

PROGRAM :

```
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 alse;
         }
         return true;
      };
   }

   public static PerformOperation isPalindrome() {
      return a -> {
         int temp = a, rev = 0;
```

```
        while (temp > 0) {
            rev = rev * 10 + temp % 10;
            temp /= 10;
        }
        return rev == a;
    };
  }
}
```

Output :

Result :

The program successfully used lambda expressions and a functional interface to check whether numbers are odd or even, prime or composite, and palindrome or not, and displayed the correct output.

MIN – MAX PROBLEM :

PROGRAM :

```
import java.util.*;

class Result {
    public static void miniMaxSum(List<Integer> arr) {
        long sum = 0;
        int min = arr.get(0), max = arr.get(0);

        for (int x : arr) {
            sum += x;
            if (x < min) min = x;
            if (x > max) max = x;
        }

        System.out.println((sum - max) + " " + (sum - min));
    }
}

public class Solution {
    public static void main(String[] args) {
        List<Integer> arr = Arrays.asList(1, 2, 3, 4, 5);
        Result.miniMaxSum(arr);
    }
}
```

IS-PALINDROME PROBLEM :

PROGRM :

```java
public class practice{
   public static boolean ispalindrome(String name){
      int n = name.length();
      for(int i=0;i<n/2;i++){
         if(name.charAt(i) != name.charAt(n-i-1)){
            return false;
         }
      }
      return true;
   }
   public static void main(String[] args) {
      String name = "noop";
      System.out.println(ispalindrome(name));
   }
}
```

OUTPUT :

```
PS D:\java> javac practice.java
PS D:\java> java practice
false
PS D:\java>
```

ALL DIGIT COUNT :

PROGRAM :

```
class UserMainCode {
    public static int digitCount(int num) {

        int count = 0;

        while (num != 0) {
            count++;
            num = num / 10;
```

```
    }

    return count;
  }
}
```

OUTPUT :

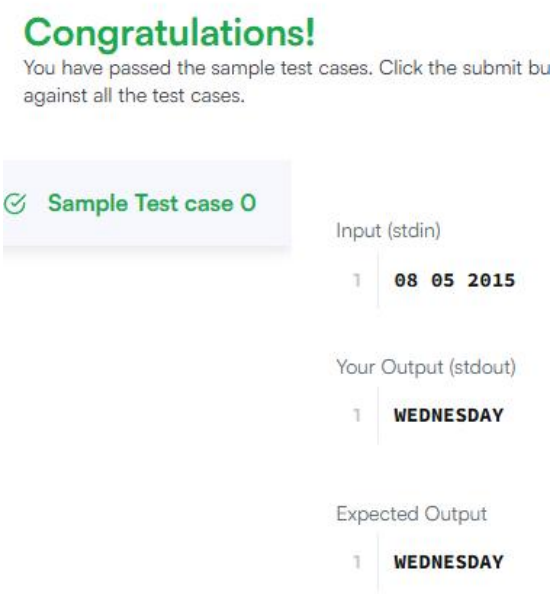
```
PS D:\java> javac practice.java
PS D:\java> java practice
4
```

JAVA DATE AND TIME

PROGRAM

```
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];
}

    }
```

OUTPUT :



HILL PATTERN :

PROGRAM :

```
public static int hillWeight(int N, int headWeight, int increment) {
    int total = 0;

    for (int i = 1; i <= N; i++) {
        int weightPerStar = headWeight + (i - 1) * increment;
        total += i * weightPerStar;
    }

    return total;
}
```

OUTPUT :

```
PS D:\java> javac practice.java
PS D:\java> java practice
90
```

SUM OF SUMS OF DIGIT

PROGRAM :

```
public class practice{
   public static int sumofdigit(int input){
      String num = String.valueOf(input);
      int total = 0;

      for(int i = 0;i<num.length();i++){
         int currentsum = 0;
         for(int j =i;j<num.length();j++){
            currentsum += num.charAt(i) - '0';
         }
         total += currentsum;
      }
      return total;
   }
   public static void main(String[] args){
      System.out.println(sumofdigit(3456));
   }
}
```

OUTPUT :

```
PS D:\java> javac practice.java
PS D:\java> java practice
40
```