WEEK 2

AIM :

To implement and use lambda expressions in Java by creating methods that return lambda expressions to check whether a given number is odd or even**,** prime or composite, and palindrome or not**,** using a functional interface.

ALGORITHM :

- Create a functional interface `PerformOperation` with a single abstract method `boolean check(int a)`.

- Create a class `MyMath` that contains:

  - A method `checker()` to apply a given lambda operation on a number.
  - A method `isOdd()` that returns a lambda expression to check if a number is odd.
  - A method `isPrime()` that returns a lambda expression to check if a number is prime.
  - A method `isPalindrome()` that returns a lambda expression to check if a number is palindrome.

- In the `main()` method:

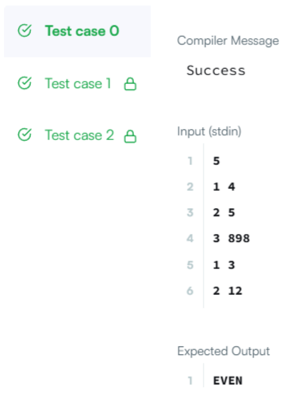
  - Read the number of test cases.
  - For each test case, read the operation type and the number.
  - Based on the operation type:
    - Call `isOdd()`, `isPrime()`, or `isPalindrome()` to obtain the corresponding lambda expression.
    - Pass the lambda expression and number to the `checker()` method.
  - Print the appropriate result.

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 false;
```

```
            }
            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 :



```
Compiler Message
Success

Input (stdin)
5
1 4
2 5
3 898
1 3
2 12

Expected Output
EVEN
```

Result :

It correctly used lambda expressions and a functional interface to determine whether the given numbers are odd or even, prime or composite, and palindrome or not, and displayed the appropriate output for each test case.

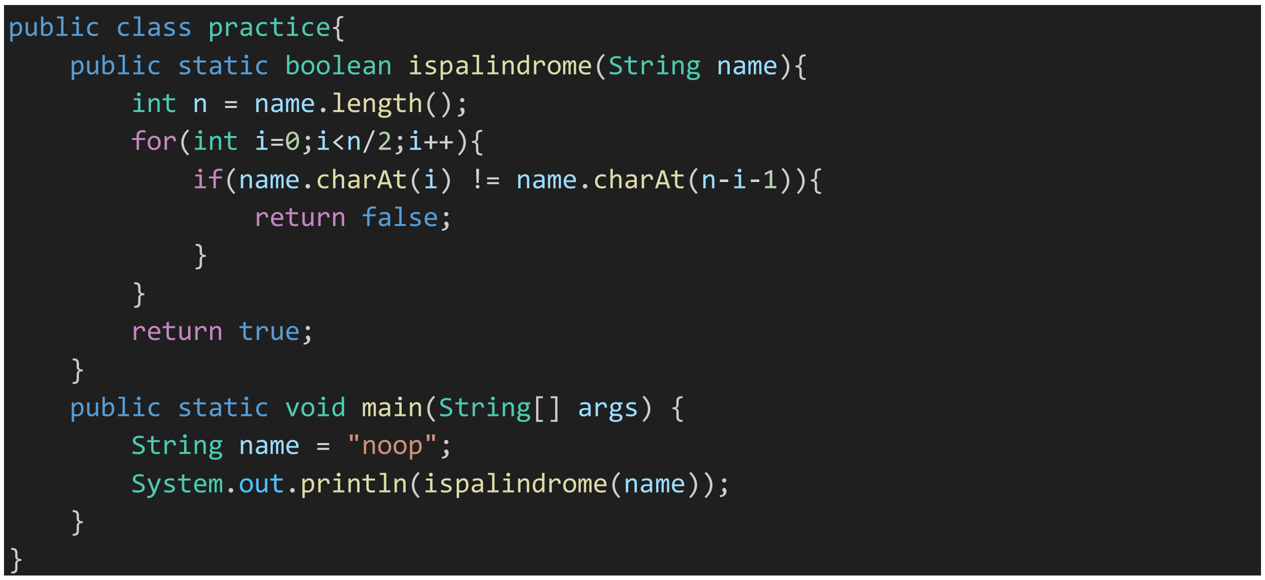
MIN – MAX PROBLEM :

PROGRAM :

```
import java.util.*;

class Result {

    public static void miniMaxSum(List<Integer> arr) {

        int min = arr.get(0);
        int max = arr.get(0);
        long sum = 0;

        for (int i = 0; i < arr.size(); i++) {
            int val = arr.get(i);
            sum += val;

            if (val < min) min = val;
            if (val > max) max = val;
        }

        long minSum = sum - max;
        long maxSum = sum - min;

        System.out.println(minSum + " " + maxSum);
    }
}

public class Solution {

    public static void main(String[] args) {

        List<Integer> arr = Arrays.asList(1, 2, 3, 4, 5);
        Result.miniMaxSum(arr);
    }
}
```

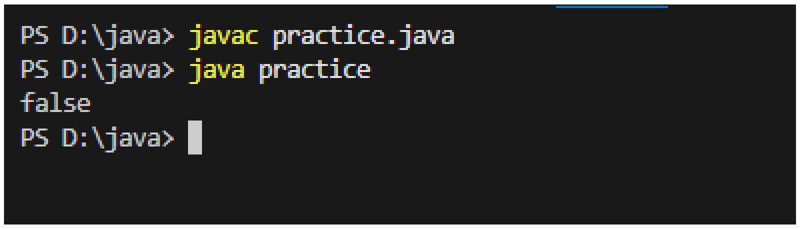
IS-PALINDROME PROBLEM :

PROGRM :



```
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 :



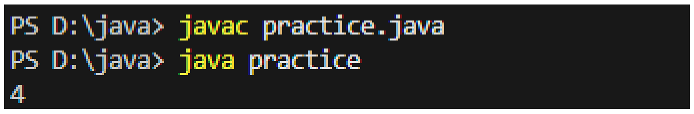
```
PS D:\java> javac practice.java
PS D:\java> java practice
false
PS D:\java> 
```

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 :



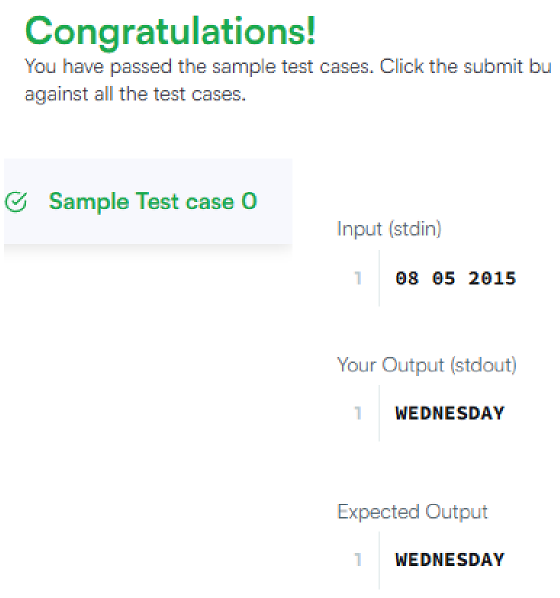
```
PS D:\java> javac practice.java
PS D:\java> java practice
4
```

JAVA DATE AND TIME

PROGRAM

```java
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 :


Congratulations!
You have passed the sample test cases. Click the submit bu
against all the test cases.

Sample Test case 0

Input (stdin)

```
08 05 2015
```

Your Output (stdout)

```
WEDNESDAY
```

Expected Output

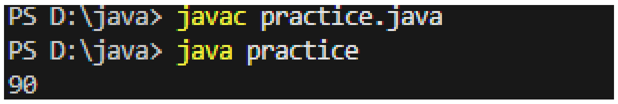
```
WEDNESDAY
```

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 :



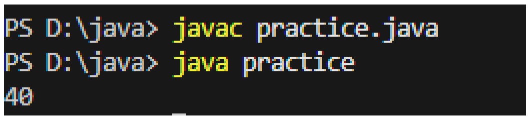
```
PS D:\java> javac practice.java
PS D:\java> java practice
90
```

SUM OF SUMS OF DIGIT

PROGRAM :

```java
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
```