ASSIGNMENT 02

1) Write a Java program to find all prime numbers between given pair of range.

```
import java.util.Scanner;

public class PrimeRange {
    public static void main(String[] args) {
        int num1, num2;
        try (Scanner in = new Scanner(System.in)) {
            System.out.println("Enter first number");
            num1 = in.nextInt();
            System.out.println("Enter second number");
            num2 = in.nextInt();
        }
        while (num1 < num2) {
            boolean flag = false;

            for(int i = 2; i <= num1/2; ++i) {
                // condition for nonprime number
                if(num1 % i == 0) {
                    flag = true;
                    break;
                }
            }

            if (!flag && num1 != 0 && num1 != 1)
                System.out.print(num1 + " ");

            ++num1;
        }
    }
}
```

OUTPUT:

```
Enter any number:
121
Number is Palindrome
```

2) Write a Java program to find all Armstrong numbers between a given pair of range.

```
import java.util.*;

class AngRange {
    public static void main(String[] args) {
        int num1, num2;
        try (Scanner sc = new Scanner(System.in)) {
            System.out.println("Enter the first number ::");
            num1 = sc.nextInt();
            System.out.println("Enter the second number ::");
            num2 = sc.nextInt();
        }
        for(int number = num1 + 1; number < num2; ++number) {
            int digits = 0;
            int result = 0;
            int originalNumber = number;

            // number of digits calculation
            while (originalNumber != 0) {
              originalNumber /= 10;
              ++digits;
            }

            originalNumber = number;

            // result contains sum of nth power of its digits
            while (originalNumber != 0) {
              int remainder = originalNumber % 10;
              result += Math.pow(remainder, digits);
              originalNumber /= 10;
            }

            if (result == number) {
              System.out.print(number + " ");
            }
        }
    }
}
```

OUTPUT:

```
Enter the number
153
It is a Angstrome
```

3) Write a Java program to find all Palindrome numbers within a given pair of range.

```
import java.util.Scanner;

public class Pali {
    public static void main(String args[]) {
        int x, number, start, end, y, temp = 0;
        Scanner in = new Scanner(System.in);
        System.out.println("Enter The Range : ");
        start = in.nextInt();
        end = in.nextInt();
        for (number = start; number <= end; number++) {
            temp=0;
            y = number;
            while (number != 0) {
                x = number % 10;
                temp = temp * 10 + x;
                number = number / 10;
            }
            if (temp == y)
                System.out.print(y + ", ");
        }
    }
}
```

OUTPUT:

```
Enter The Range:
10 20
11
```

4) Write a Java program to Fibonacci series up to a given range.

```
import java.util.*;
class Fibo{
    public static void main(String args[]){
        int n;
        try (Scanner in = new Scanner(System.in)) {
            System.out.println("Enter the last number");
            n = in.nextInt();
        }
        System.out.print("0, 1, ");
        int n1 = 0, n2 = 1;
        for(int i = 2; i<=n-1; i++, n1++,n2++){
            int sum = n1 + n2;
            System.out.print(sum+", ");
        }
    }
}
```

OUTPUT:

```
Enter the last number
13
0, 1, 1, 3, 5, 7, 9, 11, 13,
```

5) Write a Java program to check if the square root of reverse of a number is equal to reverse of that original number.

```
import java.util.Scanner;

public class rev {
    public static void main(String[] args) {
        int n;
        Scanner in = new Scanner(System.in);
        System.out.println("Enter the number");
        n = in.nextInt();
        int t = n;
        int t1, t2, rev_n = 0, rev_sq = 0;
        while(n!=0){
            t1 = n%10;
            rev_n = rev_n * 10 + t1;
            n = n / 10;
        }
        int sq_of_n = t * t;

        while(sq_of_n != 0){
            t2 = sq_of_n % 10;
            rev_sq = rev_sq * 10 + t2;
            sq_of_n = sq_of_n/10;
        }

        int sqr_rev_sqn = (int) Math.sqrt(rev_sq);

        if(sqr_rev_sqn == rev_n){
            System.out.println("Yes");
        }
        else{
            System.out.println("No");
        }
    }
}
```

OUTPUT:

```
Enter the number
12
Yes
```

6) Write a Java program to display the multiplication table up to a given number.

```
import java.util.*;
class MultiTable{
    public static void main(String args[]){
        int n;
        Scanner in = new Scanner(System.in);
        System.out.println("Enter the last number");
        n = in.nextInt();
        int r;
        System.out.println("");
        System.out.println("Multiplication  Table of 1 is: ");
        for(int j = 1; j<=n; j++){
            for(int i = 1; i<=10; i++){
                r = j * i;
                System.out.println(j+" X "+i+" = "+r);
            }
            System.out.println("");
            System.out.println("Multiplication  Table of "+(j+1)+"is: ");
        }
    }
}
```

OUTPUT:

```
Enter the last number
3
```

```
Multiplication  Table of 1 is:
1 X 1 = 1
1 X 2 = 2
1 X 3 = 3
1 X 4 = 4
1 X 5 = 5
1 X 6 = 6
1 X 7 = 7
1 X 8 = 8
1 X 9 = 9
1 X 10 = 10

Multiplication  Table of 2is:
2 X 1 = 2
2 X 2 = 4
2 X 3 = 6
2 X 4 = 8
2 X 5 = 10
2 X 6 = 12
2 X 7 = 14
2 X 8 = 16
2 X 9 = 18
2 X 10 = 20

Multiplication  Table of 3is:
3 X 1 = 3
3 X 2 = 6
3 X 3 = 9
3 X 4 = 12
3 X 5 = 15
3 X 6 = 18
3 X 7 = 21
3 X 8 = 24
3 X 9 = 27
3 X 10 = 30
```

NAME – PRIYANSHU MALLICK
SIC – 21BCSF11
ROLL. NO – 30
SEC – B, GROUP – B2
BRANCH - CSE