

Assignment 2:

- 1) Write a program which creates an integer array and displays sum of its elements.

```
import java.util.Scanner;

public class ArraySum {
    public static int getSum(int[] array) {
        int sum = 0;
        for (int i = 0; i < array.length; i++) {
            sum += array[i];
        }
        return sum;
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        int[] array = new int[n];
        for (int i = 0; i < n; i++) {
            array[i] = sc.nextInt();
        }
        sc.close();
        System.out.println(getSum(array));
    }
}
```

- 2) Write a program which performs addition of elements which are stored in two arrays of type double.

```
import java.util.Scanner;

class DoubleArraySum{
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter length of first array: ");
        int arr1Length = sc.nextInt();
        System.out.println("Enter length of second array: ");
        int arr2Length = sc.nextInt();

        double arr1[] = new double[arr1Length];
        double arr2[] = new double[arr2Length];
        System.out.println("Enter elements of first array: ");
        for(int i = 0; i < arr1Length; i++){
            arr1[i] = sc.nextDouble();
        }
        System.out.println("Enter elements of second array: ");
        for(int i = 0; i < arr2Length; i++){
            arr2[i] = sc.nextDouble();
        }
    }
}
```

```

        sc.close();

        int minLen = Math.min(arr1Length, arr2Length);
        int maxLen = Math.max(arr1Length, arr2Length);
        int result[] = new int[maxLen];

        for(int i = 0; i<minLen;i++){
            result[i] = (int) (arr1[i] + arr2[i]);
        }
        for(int i = minLen; i<maxLen;i++){
            if(arr1Length > arr2Length){
                result[i] = (int) (arr1[i]);
            }
            else{
                result[i] = (int) (arr2[i]);
            }
        }

        System.out.println("The resultant sum array is: ");
        for(int i = 0; i<maxLen;i++){
            System.out.print(result[i] + " ");
        }
    }
}

```

3) Write a method that receives a name as parameter and prints on the console. "Hello, <name>!"

```

import java.util.*;
public class Greeting {

    public static void greetPerson(String name) {
        System.out.println("Hello, " + name + "!");
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the name: ");
        String name = sc.nextLine();
        greetPerson(name);
        sc.close();
    }
}

```

- 4) Create a method `GetMax(int a, int b, int c)`, that returns maximal of three numbers. Write a program that reads three numbers from the console and prints the biggest of them.

```
import java.io.Console;
public class MaxOfThree {

    public static int maxOfThree(int val1, int val2, int val3) {
        return (val1 > val2) ? ((val1 > val3) ? val1 : val3) : ((val2 > val3) ? val2 :
val3);
    }
    public static void main(String[] args){
        Console console = System.console();
        int val1 = Integer.valueOf(console.readLine());
        int val2 = Integer.valueOf(console.readLine());
        int val3 = Integer.valueOf(console.readLine());
        System.out.println(maxOfThree(val1, val2, val3));
    }
}
```

- 5) Write a method that prints the digits of a given decimal number in a reversed order.

```
import java.util.Scanner;
public class PrintReverse {

    public static void printReverse(int n){
        int reverse = 0;
        while(n>0){
            reverse = reverse*10 + n%10;
            n = n/10;
        }
        System.out.println(reverse);
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        printReverse(n);
        sc.close();
    }
}
```

- 6) Write a Boolean method `IsPrime(n)` that check whether a given integer number `n` is prime.

```

import java.util.Scanner;

public class CheckPrime {
    public static boolean isPrime(long n) {
        if(n<=1) return false;
        if(n == 2 || n == 3) return true;
        if(n%2 == 0 || n%3 == 0) return false;

        for(int i=5;i<=Math.sqrt(n);i+=6){
            if(n%i==0 || n%(i+2)==0){
                return false;
            }
        }
        return true;
    }

    public static void main(String args[]){
        Scanner sc = new Scanner(System.in);
        long n = sc.nextLong();
        sc.close();
        if(isPrime(n)){
            System.out.println("true");
        }
        else{
            System.out.println("false");
        }
    }
}

```

- 7) Write a method that calculates all prime numbers in given range and returns them as list of integers. Write a method to print a list of integers. Write a program that takes two integer numbers (each at a separate line) and prints all primes in their range, separated by a comma.

```

import java.util.ArrayList;
import java.util.Scanner;
public class PrimesBetweenRange {

    public static boolean isPrime(long n) {
        if(n<=1) return false;
        if(n == 2 || n == 3) return true;
        if(n%2 == 0 || n%3 == 0) return false;

        for(int i=5;i<=Math.sqrt(n);i+=6){
            if(n%i==0 || n%(i+2)==0){
                return false;
            }
        }
        return true;
    }
}

```

```

public static ArrayList<Integer> generatePrimeList(int init, int end ){
    ArrayList<Integer> primes = new ArrayList<>();

    for(int i = init; i<=end; i++){
        if(isPrime(i)){
            primes.add(i);
        }
    }
    return primes;
}

public static void printList(ArrayList<Integer> list){
    for(int i = 0; i<list.size(); i++){
        System.out.print(list.get(i) + ",");
    }
}

public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    int val1 = sc.nextInt();
    int val2 = sc.nextInt();
    ArrayList<Integer> primesList = generatePrimeList(val1, val2);
    sc.close();
    printList(primesList);
}
}

```

8) Write a program that can calculate the area of four different geometry figures - triangle, square, rectangle and circle. On the first line you will get the figure type. Next you will get parameters for the chosen figure, each on a different line:

- Triangle - side and height
- Square - side
- Rectangle - width and height
- Circle - radius

```

import java.util.Scanner;

public class ShapeArea {
    public static void main(String args[]){
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter the shape name: ");
        System.out.println("Triangle, Square, Rectangle, Circle");
        String shape=sc.next();
        switch (shape){

            case "Traingle":
                int s=sc.nextInt();
                int h=sc.nextInt();
                System.out.println("Area of Triangle is "+0.5*s*h);
                break;

```

```

    case "Square":
        int l=sc.nextInt();
        System.out.println("Area of Square is "+l*l);
        break;

    case "Rectangle":
        int a=sc.nextInt();
        int b=sc.nextInt();
        System.out.println("Area of Rectangle is "+a*b);
        break;

    case "Circle":
        double r=sc.nextInt();
        System.out.println("Area of Circle is "+3.14*r*r);
        break;

    default:
        System.out.println("Invalid option");
    }
    sc.close();
}
}

```

9) Write a method which accepts two integer arrays and returns an array of unique elements.

```

import java.util.ArrayList;
import java.util.Scanner;
public class UniqueElements {

    public static ArrayList<Integer> uniqElements(int[] array1, int[] array2) {

        ArrayList<Integer> result = new ArrayList<Integer>();

        for (int i = 0; i < array1.length; i++) {
            if (!result.contains(array1[i])) {
                result.add(array1[i]);
            }
        }
        for (int i = 0; i < array2.length; i++) {
            if (!result.contains(array2[i])) {
                result.add(array2[i]);
            }
        }
        return result;
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        int[] array1 = new int[n];
    }
}

```

```

    for (int i = 0; i < n; i++) {
        array1[i] = sc.nextInt();
    }
    int m = sc.nextInt();
    int[] array2 = new int[m];
    for (int i = 0; i < m; i++) {
        array2[i] = sc.nextInt();
    }
    sc.close();

    ArrayList<Integer> result = uniqElements(array1, array2);
    for (int i = 0; i < result.size(); i++) {
        System.out.print(result.get(i) + " ");
    }
}
}

```

10) Predict the output of the given below:

Output:

0	1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18	19
20	21	22	23	24	25	26	27	28	29
30	31	32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47	48	49

11) Write a method public static boolean isRowMagic(int[][] a) that checks if the array is row-magic (*this means that every row has the same row sum*).

```

import java.util.Scanner;
public class RowMagic {

    private static boolean rowMagic(int[][] arr1,int row,int col) {
        int sum = 0;
        int res[] = new int[row];
        for(int i= 0;i< row;i++){
            for(int j=0;j<col;j++){
                sum+=arr1[i][j];
            }
            res[i]=sum;
        }
        for(int i=0;i<row-1;i++){
            if(res[i]!=res[i+1]){
                return false;
            }
        }
        return true;
    }
}

```

```

public static void main(String[] args) {
    Scanner sc=new Scanner(System.in);
    int row = sc.nextInt();
    int col = sc.nextInt();
    int arr[][]=new int[row][col];
    for(int i=0;i<row;i++){
        for(int j=0;j<col;j++){
            arr[i][j]=sc.nextInt();
        }
    }
    System.out.println(rowMagic(arr,row,col));
    sc.close();
}
}

```

12) Write a method public static boolean isMagic(int[][] a) that checks if the array is a magic square. This means that it must be square, and that all row sums, all column sums, and the two diagonal-sums must all be equal.

```

import java.util.Scanner;

public class AllMagicMatrix {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter No of Row's and Column of Matrix:");
        int n = sc.nextInt();
        int m = sc.nextInt();
        int[][] arr = new int[n][m];
        for (int i = 0; i < n; i++) {
            for (int j = 0; j < m; j++) {
                arr[i][j] = sc.nextInt();
            }
        }
        sc.close();
        System.out.println(isMagicMatrix(arr, n, m));
    }

    public static boolean isMagicMatrix(int[][] arr, int row, int col) {

        if(row != col)    return false;
        int rowSum = 0;
        int colSum = 0;
        int diagSum = 0;
        int antiDiagSum = 0;
        for (int i = 0; i < row; i++) {
            for (int j = 0; j < col; j++) {
                rowSum += arr[i][j];
                colSum += arr[j][i];
            }
        }
    }
}

```



```
    for (int i = 0; i < row; i++) {  
        diagSum += arr[i][i];  
        antiDiagSum += arr[i][col - 1 - i];  
    }  
    int bothDiagnolSum = diagSum + antiDiagSum;  
  
    return ( rowSum == colSum && rowSum == bothDiagnolSum);  
}  
  
}
```