Assignment-I

1. **Write a java program to display following messages.**

Hello World!
Hello Again
I like typing this.
This is fun.
Yay! Printing.
I'd much rather you 'not'.
I "said" do not touch this.

```
    public class A1Q1 {
    public static void main(String[] args) {
        System.out.println(“Hello World!”);
        System.out.println(“Hello Again”);
        System.out.println(“I like typing this.”);
        System.out.println(“This is fun.”);
        System.out.println(“Yay! Printing.”);
        System.out.println(“I'd much rather you 'not'.”);
        System.out.println(“I \”said\” do not touch this.”);
        }
    }
```

5. Write a java program that stores your Regd. No and year of admission into two variables, and displays their values on the screen.

My Regd. No is 191123142 and I have taken admission in B. Tech. In 2019.

```
public class A1Q5 {
    public static void main(String[] args) {
        int regd=191123142;
        int year=2019;
        System.out.println("My Regd. No is “+regd+ “and I have taken admission in B. Tech. 
In”+year+”.”);
        }
    }
```

6. Let we have two values 113, 2.71828. Then, declare two different variables to hold the given values. Write the java program to display the values of these two variables on the screen, one per line.

This is room # 113
e is close to 2.71828

```
public class A1Q6 {
    public static void main(String[] args) {
        double e=2.71828;
        int a=113;
        System.out.println("This is room # “+a);
        System.out.println(“e is close to “+e);
        }
    }
```

7. Write a java program to exchange the values of two variables of integer type A and B using third temporary variable C.

```
public class A1Q7 {
	public static void main(String[] args) {
		int a,b,c;
		a=12;
		b=15;
		System.out.println("Before swapping a="+a+"b="+b);
		c=a;
		a=b;
		b=c;
		System.out.println("After swapping a="+a+"b="+b);
	}
}
```

8. Write a java program to exchange the values of two variables of integer type A and B without using third temporary variable.

```
public class A1q8 {
	public static void main(String[] args) {
		int a=14;
		int b=17;
		System.out.println("Before swapping a="+a+"b="+b);
		a=a+b;
		b=a-b;
		a=a-b;
		System.out.println("After swapping a="+a+"b="+b);
	}
}
```

9. What do each of the following print?
a.System.out.println(2 + "bc");
b.System.out.println(2 + 3 + "bc");
c.System.out.println((2+3) + "bc");
d.System.out.println("bc" + (2+3));
e.System.out.println("bc" + 2 + 3);

```
2bc
5bc
5bc
bc5
bc23
```

10. What do each of the following print?
a. System.out.println('b');
b. System.out.println('b' + 'c');
c. System.out.println((char) ('a' + 4));

```
b
197
e
```

11. Suppose that a variable a is declared as int a = 2147483647 (or equivalently, Integer.MAX_VALUE). What do each of the following print?
a. System.out.println(a);
b. System.out.println(a+1);
c. System.out.println(2-a);
d. System.out.println(-2-a);

e. System.out.println(2*a);
f. System.out.println(4*a);

```
2147483647
-2147483648
-2147483645
2147483647
-2
-4
```

12. Suppose that a variable a is declared as double a = 3.14159. What do each of the following print?
a. System.out.println(a);
b. System.out.println(a+1);
c. System.out.println(8/(int) a);
d. System.out.println(8/a);
e. System.out.println((int) (8/a));

```
3.14159
4.14159
2
2.5464812403910124
2
```

13. Assume a string variable *ruler1* contains "1" initially i.e. *String ruler1="1"* . Write a java program to print the following output.

```
1
1 2 1
1 2 1 3 1 2 1
1 2 1 3 1 2 1 4 1 2 1 3 1 2 1
```

```
public class A1q13 {
    public static void main(String[] args) {
        String ruler1="1";
        System.out.println(ruler1);
        ruler1=ruler1+ 2 + ruler1;
        System.out.println(ruler1);
        ruler1=ruler1+ 3 + ruler1;
        System.out.println(ruler1);
        ruler1=ruler1+ 4 + ruler1;
        System.out.println(ruler1);
        }
}
```

Assignment-II

Using Keyboard Input

1. Write a program that reads a Celsius degree in a double value from the console, then converts it to Fahrenheit and displays the result. The formula for the conversion is as follows:

fahrenheit = (9 / 5) * celsius + 32

```
import java.util.Scanner;
public class A2Q1 {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.print("Enter a degree in Celsius: ");
        double celsius = input.nextDouble();
        double fahrenheit = (9.0 / 5) * celsius + 32;
        System.out.println(celsius + " Celsius is " + fahrenheit + " Fahrenheit");
    }
}
```

2. Write a program that reads in the radius and length of a cylinder and computes the area and volume using the following formulas:

area = radius * radius * pi
volume = area * length

```
import java.util.Scanner;
public class A2Q2 {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        final double PI = 3.14159265359;
        System.out.print("Enter the radius and length of a cylinder: ");
        double radius = input.nextDouble();
        double length = input.nextDouble();
        double area = radius * radius * PI;
        double volume = area * length;
        System.out.println("The area is " + area);
        System.out.println("The volume is " + volume);
    }
}
```

3. Write a program that reads a number in feet, converts it to meters, and displays the result. One foot is 0.305 meter.

```
import java.util.Scanner;
public class A2Q3 {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.print("Enter a value for feet: ");
        double feet = input.nextDouble();
        double meters = feet * 0.305;
        System.out.println(feet + " feet is " + meters + " meters");
    }
}
```

4. Write a program that reads an integer between 0 and 1000 and adds all the digits in the integer. For example, if an integer is 932, the sum of all its digits is 14. Hint: Use the % operator

to extract digits, and use the / operator to remove the extracted digit. For instance, 932 % 10 = 2 and 932 / 10 = 93.

```
import java.util.Scanner;
public class A2Q4 {
    public static void main(String[] args) {
    Scanner sc=new Scanner(System.in);
    int n, x, sum=0;
    System.out.println("Enter number between 0 and 1000");
    n=sc.nextInt();

    x=n % 10;
    sum=sum + x;
    n=n / 10;

    x=n % 10;
    sum=sum + x;
    n=n / 10;

    x=n % 10;
    sum=sum + x;
    n=n / 10;
    System.out.println("The sum of the digits is" + sum);
    }
}
```

5. Average acceleration is defined as the change of velocity divided by the time taken to make the change, as shown in the following formula:

$$a = \frac{v1 - v0}{t}$$

Write a program that prompts the user to enter the starting velocity v0 in meters/second, the ending velocity v1 in meters/second, and the time span t in seconds,and displays the average acceleration.

```
import java.util.Scanner;
public class A2Q5 {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.print("Enter v0, v1 and t: ");
        double v0 = input.nextDouble();
        double v1 = input.nextDouble();
        double t = input.nextDouble();
        double a = (v1 - v0) / t;
        System.out.println("The average acceleration is " + a);
    }
}
```

6. Body Mass Index (BMI) is a measure of health on weight. It can be calculated by taking your weight in kilograms and dividing by the square of your height in meters. Write a program that prompts the user to enter a weight in pounds and height in inches and displays the BMI. Note that one pound is 0.45359237 kilograms and one inch is 0.0254 meters.

```
import java.util.Scanner;
```

```
public class A2Q6{
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.print("Enter weight in pounds: ");
        double weight = input.nextDouble();
        System.out.print("Enter height in inches: ");
        double height = input.nextDouble();
        weight = weight * 0.45359237;
        height = height * 0.0254;
        double bmi = weight / Math.pow(height, 2);
        System.out.println("BMI is " + bmi);
    }
}
```

7. Write a program that prompts the user to enter the side of a hexagon and displays its area.

```
import java.util.Scanner;
public class A2Q7{
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.print("Enter the side: ");
        double side = input.nextDouble();
        double area = ((3 * Math.pow(3, 0.5)) / 2) * Math.pow(side, 2);
        System.out.println("The area of the hexagon is " + area);
    }
}
```

8. Write a program that displays the following table. Cast floating point numbers into integers.

a	**b**	**pow(a, b)**
1	**2**	**1**
2	**3**	**8**
3	**4**	**81**
4	**5**	**1024**
5	**6**	**15625**

```
public class A2Q8 {
    public static void main(String[] args) {
        float a, b;
        System.out.println("a        b        pow(a, b)");
        a = 1;
        b = 2;
        System.out.println((int)a + "        " + (int)b + "        " + (int)Math.pow(a, b));
        a++;
        b++;
        System.out.println((int)a + "        " + (int)b + "        " + (int)Math.pow(a, b));
        a++;
        b++;
        System.out.println((int)a + "        " + (int)b + "        " + (int)Math.pow(a, b));
        a++;
        b++;
        System.out.println((int)a + "        " + (int)b + "        " + (int)Math.pow(a, b));
        a++;
        b++;
        System.out.println((int)a + "        " + (int)b + "        " + (int)Math.pow(a, b));
    }
}
```

9. Write a program that prompts the user to enter two points (x1, y1) and (x2, y2) and displays their distance between them.

```
import java.util.Scanner;
public class A2Q9 {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.print("Enter x1 and y1: ");
        double x1 = input.nextDouble();
        double y1 = input.nextDouble();
        System.out.print("Enter x2 and y2: ");
        double x2 = input.nextDouble();
        double y2 = input.nextDouble();
        double distance = Math.pow(Math.pow(x2 - x1, 2) +Math.pow(y2 - y1, 2), 0.5);
        System.out.println("The distance between the two points is " + distance);
    }
}
```

10. Write a program that prompts the user to enter three points (x1, y1), (x2, y2), (x3, y3) of a triangle and displays its area.

```
import java.util.Scanner;
public class A2Q10 {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.print("Enter three points for a triangle: ");
        double x1 = input.nextDouble();
        double y1 = input.nextDouble();
        double x2 = input.nextDouble();
        double y2 = input.nextDouble();
        double x3 = input.nextDouble();
        double y3 = input.nextDouble();
        double side1 = Math.pow(Math.pow(x2 - x1, 2) + Math.pow(y2 - y1, 2), 0.5);
        double side2 = Math.pow(Math.pow(x3 - x2, 2) + Math.pow(y3 - y2, 2), 0.5);
        double side3 = Math.pow(Math.pow(x1 - x3, 2) + Math.pow(y1 - y3, 2), 0.5);
        double s = (side1 + side2 + side3) / 2;
        double area = Math.pow(s * (s - side1) * (s - side2) * (s - side3), 0.5);
        System.out.println("The area of the triangle is " + area);
    }
}
```

11. Write a program that reads in investment amount, annual interest rate, and number of years, and displays the future investment value using the following formula:
futureInvestmentValue =investmentAmount * (1 + monthlyInterestRate)^numberOfYears*12
For example, if you enter amount 1000, annual interest rate 3.25%, and number of years 1, the future investment value is 1032.98.

```
import java.util.Scanner;
public class A2Q11 {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.print("Enter investment amount: ");
        double amount = input.nextDouble();
        System.out.print("Enter annual interest rate in percentage: ");
        double monthlyInterestRate = input.nextDouble();
        monthlyInterestRate /= 1200;
        System.out.print("Enter number of years: ");
        int years = input.nextInt();
    double futureInvestmentValue =  amount * Math.pow(1 + monthlyInterestRate, years * 12);
        System.out.println("Accumulated value is $" + futureInvestmentValue);
    }
}
```

12. If you have N eggs, then you have N/12 dozen eggs, with N%12 eggs left over. (This is essentially the definition of the / and % operators for integers.) Write a java program that asks the user how many eggs she has and then tells the user how many dozen eggs she has and how many extra eggs are left over. A gross of eggs is equal to 144 eggs. Extend your program so that it will tell the user how many gross, how many dozen, and how many left over eggs she has. For example, if the user says that she has 1342 eggs, and then your program would respond with "Your number of eggs is 9 gross, 3 dozen, and 10."

```
import java.util.Scanner;
public class A2Q12 {
    public static void main(String[] args) {
        int num,dozen,gross;
        Scanner sc=new Scanner(System.in);
        System.out.println("How many eggs do you have?");
        num=sc.nextInt();
        gross=num/144;
        num=num%144;
        dozen=num/12;
        num=num%12;
       System.out.println("Your number of eggs "+gross+"gross,"+dozen+"dozens,"+"and"+num);
    }
}
```

13. Write a program that prompts the user to enter the minutes (e.g., 1 billion), and displays the number of years and days for the minutes. For simplicity, assume a year has 365 days.

```
import java.util.Scanner;
public class A2Q13 {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.print("Enter the number of minutes: ");
        int minutes = input.nextInt();
        int years = minutes / 525600;
        int days = (minutes % 525600) / 1440;
System.out.println(minutes + "minutes is approximately " + years + " years and " + days + " days");
    }
}
```

Using Command Line Argument

14. Write a java program that takes two positive integers as command-line arguments and prints true if either evenly divides the other.

```
public class A2Q14 {
	public static void main(String[] args) {
		int a,b;
		a=Integer.parseInt(args[0]);
		b=Integer.parseInt(args[1]);
		boolean ans=((a%b==0)||(b%a==0));
		System.out.println(ans);
	}
}
```

15. Write a java program that takes two int values a and b from the command line and prints a random integer between a and b.

```
public class A2Q15 {
	public static void main(String[] args) {
		int min;
		int max;
		min=Integer.parseInt(args[0]);
		max=Integer.parseInt(args[1]);
		int c=min+(int)(Math.random()*(max-min+1));
		System.out.println(“c=”+c);
	}
}
```

16. Write a java program that prints the sum of two random integers between 1 and 6 (such as you might get when rolling dice).

```
public class A2Q16 {
	public static void main(String[] args) {
		int a,b,res1,res2;
		a=Integer.parseInt(args[0]);
		b=Integer.parseInt(args[1]);
		res1=a+(int)(Math.random()*(a-b+1));
		res2=a+(int)(Math.random()*(a-b+1));
		int sum=res1+res2;
		System.out.println(sum);
	}}
```

17. Write a java program that takes three positive integers as command-line arguments and prints true if any one of them is greater than or equal to the sum of the other two and false otherwise. (Note: This computation tests whether the three numbers could be the lengths of the sides of some triangle.)

```
public class A2Q17 {
	public static void main(String[] args) {
		int a,b,c;
		a=Integer.parseInt(args[0]);
		b=Integer.parseInt(args[1]);
```

```
        c=Integer.parseInt(args[2]);
        boolean res=((a+b)>=c)||((b+c)>=a)||((a+c)>=b);
        System.out.println(res);
    }
}
```

18. Write a java program that takes three double values x, y, and z as command-line arguments and prints true if the values are strictly ascending or descending (x < y < z or x > y > z), and false otherwise.

```
public class A2Q18 {
    public static void main(String[] args) {
        int x,y,z;
        x=Integer.parseInt(args[0]);
        y=Integer.parseInt(args[1]);
        z=Integer.parseInt(args[2]);
        boolean res=((x<y&&y<z)||(x>y&&y>z));
        System.out.println(res);
    }
}
```

19. Input the basic salary of an employee of an organization through the keyboard. His dearness allowance (DA) is 40% of basic salary, and house rent allowance (HRA) is 20% of basic salary. Write a java program to calculate his gross salary.

```
public class A2Q19 {
    public static void main(String[] args) {
        double basic,HRA,DA,gross;
        System.out.println("Enter the basic salary of the person");
        basic=Double.parseDouble(args[0]);
        DA=0.40*basic;
        HRA=0.20*basic;
        gross=basic+DA+HRA;
        System.out.println("The gross salary is: "+gross);
    }
}
```

20. Write a program that takes two int values m and d from the command line and prints true if day d of month m is between 3/20 and 6/20, false otherwise.

```
public class A2Q20 {
    public static void main(String[] args) {
        int m,d;
        m=Integer.parseInt(args[0]);
        d=Integer.parseInt(args[1]);
        boolean res=(m==3&&d>=20)||(m==4&&d<=30)||(m==5&&d<=31)||(m==6&&d<=20);
        System.out.println(res);
    }
}
```

21.Write a java program that takes a double value t from the command line and prints the value of sin (2t) + sin (3t).

```
public class A2Q21 {
    public static void main(String[] args) {
        double t,sum;
        t=Double.parseDouble(args[0]);
        t=Math.toRadians(t);            // convert degree to radian
        sum=Math.sin(2*t)+Math.sin(3*t);
        System.out.println(sum);
}
}
```

22. Write a java program that calculates the monthly payments you would have to make over a given number of years to pay off a loan at a given interest rate compounded continuously, taking the number of years t, the principal P, and the annual interest rate r as command-line arguments. The desired value is given by the formula Pert. Use Math.exp ().

```
public class A2Q22 {
    public static void main(String[] args) {
        double p,r,t;
        p=Double.parseDouble(args[0]);
        r=Double.parseDouble(args[1]);
        t=Double.parseDouble(args[2]);
        double monthly_payment=p*Math.exp(r*t);
        System.out.println(monthly_payment);
    }
}
```

23. Write a program that takes three int values from the command line and prints them in ascending order. Use Math.min() and Math.max().

```
public class A2Q23 {
    public static void main(String[] args) {
        int a,b,c;
        a=Integer.parseInt(args[0]);
        b=Integer.parseInt(args[1]);
        c=Integer.parseInt(args[2]);
        int max=Math.max(a,Math.max(b,c));
        int min=Math.min(a,Math.min(b,c));
        int median=(a+b+c)-max-min;
        System.out.println(min+" "+median+" "+max);
    }
}
```

24. Write a java program that prints five uniform random values between 0 and 1, their average value, and their minimum and maximum value. Use Math.random(), Math.min(), and Math.max().

```
public class A2Q24 {
    public static void main(String[] args) {
        double r1,r2,r3,r4,r5;
        r1=Math.random();
        r2=Math.random();
        r3=Math.random();
        r4=Math.random();
        r5=Math.random();
        System.out.println(r1+" "+r2+" "+r3+" "+r4+" "+r5);
        double sum=r1+r2+r3+r4+r5;
        double avg=sum/5;
        System.out.println("Average =”+avg);
        double max=Math.max(r1,Math.max(r2,Math.max(r3,Math.max(r4, r5))));
        double min=Math.min(r1,Math.min(r2,Math.min(r3,Math.min(r4, r5))));
        System.out.println("Max=”+max);
        System.out.println("Min=”+min);
    }
}
```

ASSIGNMENT-3

1. Write a java program to input the height of the person and check if the height of the person is greater than or equal to 6 feet then print the message “The person is tall”.

```
import java.util.Scanner;
public class A3Q1 {
    public static void main(String[] args) {
        Scanner sc=new Scanner(System.in);
        int h;
        System.out.println("Enter the height of the person");
        h=sc.nextInt();
        if(h>6)
            System.out.println("The person is tall");
    }
}
```

2. Write a java program to input the mark of a student and check if the student mark is greater than or equal to 40, then it generates the following message.
”Congratulation! You have passed the exam.”
Otherwise the output message is
”Sorry! You have failed the exam.”

```
import java.util.Scanner;
public class A3Q2 {

    public static void main(String[] args) {
        Scanner sc=new Scanner(System.in);
        int mark;
        System.out.println("Enter the mark of the student");
        mark=sc.nextInt();
        if(mark>=40)
         System.out.println("Congratulations! You have passed the exam");
        else
         System.out.println("sorry! you have failed the exam");
    }
}
```

3. Input an integer through the keyboard. Write a java program to find out whether it is an odd number or even number.

```
import java.util.Scanner;
public class A3Q3 {
    public static void main(String[] args) {
      Scanner sc=new Scanner(System.in);
      int n;
      System.out.println("Enter a number");
      n=sc.nextInt();
      if(n%2==0)
        System.out.println("Even number");
        else
            System.out.println("Odd number");
        }
}
```

4. Any character is entered through the keyboard, write a java program to determine whether the character entered is a capital letter, a small case letter, a digit or a special symbol. The following table shows the range of ASCII values for various characters.

Characters	**ASCII Values**
A – Z	**65 – 90**
a – z	**97 – 122**
0 – 9	**48 – 57**
special symbols	**0 - 47, 58 - 64, 91 - 96, 123 – 127**

```
import java.util.Scanner;
public class A3Q4 {
    public static void main(String[] args) {
        char c;
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter a character");
        c=sc.next().charAt(0);
        if(c>=65 && c<=90)
            System.out.println("The entered character is an uppercase");
        else if(c>=97 && c<=122)
            System.out.println("The entered character is a lower case");
        else if(c>=48 && c<=57)
            System.out.println("It is a digit");
        else if((c>=0 && c<=47) || (c>=58 && c<=64) || (c>=91 && c<=96) || (c>=123 &&
c<=127))
            System.out.println("It is a special Character");
    }
}
```

5. The two roots of a quadratic equation ax^2 + bx + c = 0 can be obtained using the following formula: b^2 - 4ac is called the discriminant of the quadratic equation. If it is positive, the equation has two real roots. If it is zero, the equation has one root. If it is negative, the equation has no real roots.

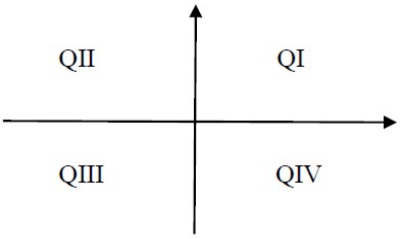
Write a program that prompts the user to enter values for a, b, and c and displays the result based on the discriminant. If the discriminant is positive, display two roots. If the discriminant is 0, display one root. Otherwise, display “The equation has no real roots”. Note that you can use Math.pow(x, 0.5) to compute 2x.

```
import java.util.Scanner;
public class A3Q5 {
	public static void main(String[] args) {
		Scanner input = new Scanner(System.in);
		System.out.print("Enter a, b, c: ");
		double a = input.nextDouble();
		double b = input.nextDouble();
		double c = input.nextDouble();
		double discriminant = Math.pow(b, 2) - 4 * a * c;
		System.out.print("The equation has ");
		if (discriminant > 0)
		{
			double root1 = (-b + Math.pow(discriminant, 0.5)) / (2 * a);
			double root2 = (-b - Math.pow(discriminant, 0.5)) / (2 * a);
			System.out.println("two roots " + root1 + " and " + root2);
		}
		else if (discriminant == 0)
		{
			double root1 = (-b + Math.pow(discriminant, 0.5)) / (2 * a);
			System.out.println("one root " + root1);
		}
		else
			System.out.println("no real roots");
	}
}
```

6. A linear equation can be solved using Cramer’s rule. Write a program that prompts the user to enter a, b, c, d, e, and f and displays the result. If ad - bc is 0, report that “The equation has no solution.”

```
import java.util.Scanner;
public class A3Q6{
	public static void main(String[] args) {
		Scanner input = new Scanner(System.in);
		System.out.print("Enter a, b, c, d, e, f: ");
		double a = input.nextDouble();
		double b = input.nextDouble();
		double c = input.nextDouble();
		double d = input.nextDouble();
		double e = input.nextDouble();
		double f = input.nextDouble();
		if (a * d - b * c == 0)
			System.out.println("The equation has no solution.");
		else
		{
			double x = (e * d - b * f) / (a * d - b * c);
			double y = (a * f - e * c) / (a * d - b * c);
			System.out.println("x is " + x + " and y is " + y);
		}
	}
}
```

7. Write a java program that takes the x – y coordinates of a point in the Cartesian plane and prints a message telling either an axis on which the point lies or the quadrant in which it is found.



```
import java.util.Scanner;
public class A3Q7 {
	public static void main(String[] args) {
		Scanner sc=new Scanner(System.in);
		double x,y;
		System.out.println("Enter the x and y coordinate respectively");
		x=sc.nextDouble();
		y=sc.nextDouble();
		if(x==0)
			System.out.println("("+x+","+y+")is on the y-axis");
		else if(y==0)
			System.out.println("("+x+","+y+") is on the x-axis");
		else if(x>0 && y>0)
			System.out.println("("+x+","+y+") is on the first quadrant");
		else if(x<0 && y>0)
			System.out.println("("+x+","+y+") is on the second quadrant");
		else if(x<0 && y<0)
			System.out.println("("+x+","+y+" is on the third quadrant");
		else if(x<0 && y>0)
			System.out.println("("+x+","+y+") is on the fourth quadrant");
		else
			System.out.println("("+x+","+y+") is at center");
	}
}
```

8. If the ages of Rahul, Ayush and Ajay are input through the keyboard, write a java program to determine the youngest of the three.

```
import java.util.Scanner;
public class A3Q8 {
	public static void main(String[] args) {
		Scanner sc=new Scanner(System.in);
		int rahul, ayush, ajay;
		System.out.println("Enter the ages of Rahul, Ayush, Ajay respectively");
		rahul=sc.nextInt();
		ayush=sc.nextInt();
		ajay=sc.nextInt();
		if(rahul<ayush && rahul<ajay)
			System.out.println("Youngest is Rahul");
		else if(ayush<rahul && ayush<ajay)
			System.out.println("The youngest is Ayush");
		else
```

```
                System.out.println("The youngest is Ajay");
    }
}
```

9. Write a program that randomly generates an integer between 1 and 12 and displays the English month name January, February, …, December for the number 1, 2, …, 12, accordingly.

```
public class A3Q9 {
    public static void main(String[] args) {
        int month = (int)((Math.random() * 12) + 1);
        switch (month)
        {
            case 1: System.out.println("January"); break;
            case 2: System.out.println("February"); break;
            case 3: System.out.println("March"); break;
            case 4: System.out.println("April"); break;
            case 5: System.out.println("May"); break;
            case 6: System.out.println("June"); break;
            case 7: System.out.println("July"); break;
            case 8: System.out.println("August"); break;
            case 9: System.out.println("September"); break;
            case 10: System.out.println("October"); break;
            case 11: System.out.println("November"); break;
            case 12: System.out.println("December");
        }
    }
}
```

10. Write a program that prompts the user to enter an integer for today's day of the week (Sunday is 0, Monday is 1, …, and Saturday is 6). Also prompt the user to enter the number of days after today for a future day and display the future day of the week.

```
import java.util.Scanner;
public class A3Q10 {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.print("Enter today's day: ");
        int day = input.nextInt();
        System.out.print("Enter the number of days elapsed since today: ");
        int daysElapsed = input.nextInt();
        int futureDay = (day + daysElapsed) % 7;
        System.out.print("Today is ");
        switch (day)
        {
            case 0: System.out.print("Sunday"); break;
            case 1: System.out.print("Monday"); break;
            case 2: System.out.print("Tuesday"); break;
            case 3: System.out.print("Wednesday"); break;
            case 4: System.out.print("Thursday"); break;
            case 5: System.out.print("Friday"); break;
            case 6: System.out.print("Saturday");
        }
        System.out.print(" and the future day is ");
        switch (futureDay)
        {
            case 0: System.out.println("Sunday"); break;
            case 1: System.out.println("Monday"); break;
            case 2: System.out.println("Tuesday"); break;
            case 3: System.out.println("Wednesday"); break;
            case 4: System.out.println("Thursday"); break;
            case 5: System.out.println("Friday"); break;
            case 6: System.out.println("Saturday");
        }
    }
}
```

11. The body mass index (BMI) is commonly used by health and nutrition professionals to estimate human body fat in populations. It is computed by taking the individual's weight (mass) in kilograms and dividing it by the square of their height in meters. i.e. $\text{BMI}=\text{weight(kg)} / (\text{height (m)})^2$ **. Then use some if statements to show the category for a given**

BMI. BMI	category
less than 18.5	underweight
18.5 to 24.9	normal weight
25.0 to 29.9	overweight
30.0 or more	obese

```
import java.util.Scanner;
public class A3Q11 {

    public static void main(String[] args) {
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter the weight of a person in kg and height in meter");
        double wt=sc.nextDouble();
        double ht=sc.nextDouble();
        double BMI=wt/(ht*ht);
        System.out.println("BMI= "+BMI);
        if(BMI<18.5)
            System.out.println("Under weight");
        else if(BMI>=18.5 && BMI<25)
            System.out.println("Normal weight");
        else if(BMI>=25 && BMI<30)
            System.out.println("Over weight");
        else
            System.out.println("Abese");
        }

    }
```

12. Write a program that prompts the user to enter three integers and display the integers in non-decreasing order.

```
import java.util.Scanner;
```

```
public class A3Q12 {
	public static void main(String[] args) {
		Scanner input = new Scanner(System.in);
		System.out.print("Enter three integers: ");
		int a = input.nextInt();
		int b = input.nextInt();
		int c = input.nextInt();
		int max=Math.max(a,Math.max(b,c));
		int min=Math.min(a,Math.min(b,c));
		int median=(a+b+c)-max-min;
		System.out.println(min + " " + median + " " + max);
	}
}
```

13. Write a program that prompts the user to enter the month and year and displays the number of days in the month. For example, if the user entered month 2 and year 2012, the program should display that February 2012 had 29 days. If the user entered month 3 and year 2015, the program should display that March 2015 had 31 days.

```
import java.util.Scanner;
public class A3Q13 {
	public static void main(String[] args) {
		Scanner input = new Scanner(System.in);
		System.out.print("Enter the month as integer: ");
		int month = input.nextInt();
		System.out.print("Enter the year as integer: ");
		int year = input.nextInt();
		boolean leapYear = (year % 4 == 0 && year % 100 != 0) || (year % 400 == 0);
		switch (month)
		{
			case 1: System.out.println(
					"January " + year + " had 31 days"); break;
			case 2: System.out.println("February " + year + " had" +
								((leapYear) ? " 29 days" : " 28 days")); break;
			case 3: System.out.println(
					"March " + year + " had 31 days"); break;
			case 4: System.out.println(
					"April " + year + " had 30 days"); break;
			case 5: System.out.println(
					"May " + year + " had 31 days"); break;
			case 6: System.out.println(
					"June " + year + " had 30 days"); break;
			case 7: System.out.println(
					"July " + year + " had 31 days"); break;
			case 8: System.out.println(
					"August " + year + " had 31 days"); break;
			case 9: System.out.println(
					"September " + year + " had 30 days"); break;
			case 10: System.out.println(
					"October " + year + " had 31 days"); break;
			case 11: System.out.println(
					"November " + year + " had 30 days"); break;
```

```
            case 12: System.out.println(
                        "December " + year + " had 31 days");
        }
    }
}
```

14. Write a program that plays the popular scissor-rock-paper game. (A scissor can cut a paper, a rock can knock a scissor, and a paper can wrap a rock.) The program randomly generates a number 0, 1, or 2 representing scissor, rock, and paper. The program prompts the user to enter a number 0, 1, or 2 and displays a message indicating whether the user or the computer wins, loses, or draws.

```
import java.util.Scanner;
public class A3Q14 {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        int computer = (int)(Math.random() * 3);
        System.out.print("scissor (0), rock (1), paper (2): ");
        int user = input.nextInt();
        System.out.print("The computer is ");
        switch (computer)
        {
            case 0: System.out.print("scissor."); break;
            case 1: System.out.print("rock."); break;
            case 2: System.out.print("paper.");
        }
        System.out.print(" You are ");
        switch (user)
        {
            case 0: System.out.print("scissor"); break;
            case 1: System.out.print("rock"); break;
            case 2: System.out.print("paper ");
        }
        if (computer == user)
            System.out.println(" too. It is a draw");
        else
        {
            boolean win = (user == 0 && computer == 2) || (user == 1 && computer == 0) ||
                          (user == 2 && computer == 1);
            if (win)
                System.out.println(". You won");
            else
                System.out.println(". You lose");
        }
    }
}
```

15. Write a program that prompts the user to enter a point (x, y) and checks whether the point is within the circle centered at (0, 0) with radius 10. For example, (4, 5) is inside the circle and (9, 9) is outside the circle.

```
import java.util.Scanner;
```

```
public class A3Q15 {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.print("Enter a point with two coordinates: ");
        double x = input.nextDouble();
        double y = input.nextDouble();
        double distance =  Math.pow(Math.pow(x, 2) + Math.pow(y, 2), 0.5;
        if (distance <= 10)
            System.out.println("Point lies in the circle");
        else
            System.out.println("Point does not lie in the circle");
    }
}
```

16. A University conducts a 100 mark exam for its student and grades them as follows. Assigns a grade based on the value of the marks. Write a java program to print the grade according to the mark secured by the student. [Use switch-case]

Mark Range	Letter Grade
>=90	O
>=80 AND <90	A
>=70 AND <80	B
>=60 AND <70	C
>=50 AND <60	D
>=40 AND <50	E
<40	F

```
import java.util.Scanner;
public class A3Q16 {
    public static void main(String[] args) {
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter the marks of a student out of 100 full mark ");
        int mark=sc.nextInt();
        switch(mark/10)
        {
            case 0: case 1: case 2: case 3:
                System.out.println("F”);
            case 4:
                System.out.println("E”);
            case 5:
                System.out.println("D”);
            case 6:
                System.out.println("C”);
            case 7:
                System.out.println("B”);
            case 8:
                System.out.println("A”);
            case 9: case 10:
                System.out.println("O”);
            default:
                System.out.println("Invalid Input”);
        }
    }
}
```

17. Write a program that prompts the user to enter an integer and determines whether it is divisible by 5 and 6, whether it is divisible by 5 or 6, and whether it is divisible by 5 or 6, but not both.

```
import java.util.Scanner;
public class A3Q17 {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.print("Enter an integer: ");
        int number = input.nextInt();
        if ((number % 5 == 0) && (number % 6 == 0))
            System.out.println("number divisible by 5 and 6");
        if ((number % 5 == 0) || (number % 6 == 0))
            System.out.println("number divisible by 5 or 6 ");
        if ((number % 5 == 0) ^ (number % 6 == 0))
            System.out.println("number divisible by 5 or 6 but not both”);
    }
}
```

18. Make a java program which displays an appropriate name for a person, using a combination of nested ifs and compound conditions. Ask the user for a gender, first name, last name and age. If the person is female and 20 or over, ask if she is married. If so, display "Mrs." in front of her name. If not, display "Ms." in front of her name. If the female is under 20, display her first and last name. If the person is male and 20 or over, display "Mr." in front of his name. Otherwise, display his first and last name. Note that asking a person if they are married should only be done if they are female and 20 or older, which means you will have a single if and else nested inside one of your if statements. Also, did you know that with an if statements (or else), the curly braces are optional when there is only one statement inside?

```
import java.util.Scanner;
public class A3Q18 {
    public static void main(String[] args) {
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter the Gender (M/F) ");
        char g=sc.next().charAt(0);
        System.out.println("Enter First Name");
        String fn=sc.next();
        System.out.println("Enter last Name");
        String ln=sc.next();
        System.out.println("Enter Age");
        int age=sc.nextInt();
        if(g=='F')
         {
            if(age>=20)
             {
               System.out.println("Are you married"+fn+"(y or n)?");
                 char m=sc.next().charAt(0);
                  if(m=='y')
                   System.out.println("Then I shall call you Mrs."+ fn+" "+ln);
                   else
```

```
                     System.out.println("Then I shall call you Ms."+ fn+" "+ln);
                 }
                else
                   {
                   System.out.println("Then I shall call you "+ fn+" "+ln);
                   }
          }
         else
         {
            if(g=='M')
              {
              if(age>=20)
              {
                System.out.println("Then I shall call you Mr."+ fn+" "+ln);
              }
              else
              {
                System.out.println("Then I shall call you "+ fn+" "+ln);
              }
          }
       }
     }
}
```

Assignment – IV

1. Write a java program to input a string message and display it 10 times in the following manner. Use a *while* loop. Let the string message be "Hello".

```
public class A4q1 {
	public static void main(String[]args)
	{
		Scanner sc=new Scanner(System.in);
		String msg;
		System.out.println("Enter the message");
		msg=sc.next();
		int i=1;
		while(i<=10)
		{
			if(i==1)
				System.out.println(i+"st "+msg);
			else if(i==2)
				System.out.println(i+"nd "+msg);
			else if(i==3)
				System.out.println(i+"rd "+msg);
			else
				System.out.println(i+"th "+msg);
			i++;
		}
	}
}
```

2. Rewrite the above java program in such a way that takes the number of lines to print as a command-line argument. You may assume that the argument is less than 1000. Hint: Use i % 10 and i % 100 to determine when to use st, nd, rd, or th for printing the ith Hello.

```
public class A4q2 {
	public static void main(String[] args) {
		int i=1,n;
		n=Integer.parseInt(args[0]);
		while(i<=n)
		{
			if(i%10==1&&i%100!=11)
				System.out.println(i+"st Hello");
			else if(i%10==2&&i%100!=12)
				System.out.println(i+"nd Hello");
			else if(i%10==3&&i%100!=13)
				System.out.println(i+"rd Hello");
			else
				System.out.println(i+"th Hello");
			i++;
		}
	}
}
```

3. Write a java program that gets an integer from the user. Count from 0 to that number. Use a *for* loop to do it.

```
Count to: 20
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
```

```
import java.util.*;
public class A4q3 {

    public static void main(String[] args) {
        int i,n;
        Scanner sc=new Scanner(System.in);
              System.out.println(“enter n”);
        n=sc.nextInt();
        for(i=0;i<=n;i++)
        {
            System.out.print(i+" ");
        }
    }
}
```

4. Write a java program that gets three integers from the user. Count from the first number to the second number in increments of the third number. Use a *for* loop to do it.

```
Count from: 4
Count to: 13
Count by: 3
4 7 10 13
```

```
import java.util.*;
public class A4q4 {
    public static void main(String[] args) {
        int cf,ct,cb,i;
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter the countfrom");
        cf=sc.nextInt();
        System.out.println("Enter the countto");
        ct=sc.nextInt();
        System.out.println("Enter the countby");
        cb=sc.nextInt();
        for(i=cf;i<=ct;i+=cb)
        {
            System.out.println(i+" ");
        }
    }
}
```

5. Write a java program that uses a *for* loop. With the loop, make the variable x go from -2 to 2, counting by 0.5. (This means that x can't be an int.)

```
public class A4q5 {
    public static void main(String[] args) {
        double x
        for(x=-2;x<=2;x+=0.5)
        {
            System.out.println(x);
        }
    }
}
```

6. Write a java program that, using one for loop and one if statement, prints the integers from 1,000 to 2,000 with five integers per line. Hint: Use the % operation.

```
public class A4q6 {

    public static void main(String[] args) {
        int i,ctr=0;
        for(i=1000;i<=2000;i++)
        {
            System.out.print(i+" ");
            ctr++;
            if(ctr%5==0)
                System.out.println();
        }
    }
}
```

7. Write a java program that takes an integer N as a command-line argument, uses Math.random() to print N uniform random values between 0 and 1, and then prints their average value.

```
public class A4q7 {

    public static void main(String[] args) {
        int n,i;
        double r,s=0,avg=0;
        n=Integer.parseInt(args[0]);
        for(i=1;i<=n;i++)
        {
            r=Math.random();
            System.out.println("Random no."+i+"="+r);
            s+=r;
        }
        System.out.println("Sum of the random numbers="+s);
        avg=s/n;
        System.out.println("average="+avg);
    }
}
```

8. Write a java program to print the following output using loop.

```
1
121
1213121
121312141213121
1213121412131215121312141213121
```

```
public class A4q8 {

    public static void main(String[] args) {
        int i;
        String p="";
        for(i=1;i<=5;i++)
        {
            p+=i+p;
            System.out.println(p);
        }
    }
}
```

9. If we list all the natural numbers below 10 that are multiples of 3 or 5, we get 3, 5, 6 and 9. The sum of these multiples is 23. Write a java program to find the sum of all the multiples of 3 or 5 below 1000.

```
public class A4q9 {
    public static void main(String[] args) {
        int i,sum=0;
        for(i=1;i<1000;i++)
        {
            if(i%3==0||i%5==0)
                sum+=i;
        }
        System.out.println("Sum of multiples of 3&5 below 1000 is= "+sum);
    }
}
```

10. Write a java program to print the multiplication table of a number entered by the user.

```
import java.util.*;
public class A4q10 {

    public static void main(String[] args) {
        int i,r=0;
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter a number whose multiplication table you want to find: ");
        int num=sc.nextInt();
        for(i=1;i<=10;i++)
        {
            r=1;
            r=num*i;
            System.out.println(num+"x"+i+"="+r);
        }
    }
}
```

11. Write a java program to find the difference between the sum of the squares of the first one hundred natural numbers and the square of the sum.
The sum of the squares of the first ten natural numbers is,
12 + 22 + ... + 102 = 385
The square of the sum of the first ten natural numbers is,
(1 + 2 + ... + 10)2 = 552 = 3025
Hence the difference between the sum of the squares of the first ten natural numbers and the square of the sum is 3025 − 385 = 2640.

```
public class A4q11 {

    public static void main(String[] args) {
        int i,sum1=0,sum2=0,diff=0,s=0;
        for(i=1;i<=10;i++)
        {
            sum1+=i*i;
            sum2+=i;
        }
        s=sum2*sum2;
        diff=s-sum1;
        System.out.println("The sum of squares of the integers="+sum1);
        System.out.println("The square of the sum of the integers="+s);
        System.out.println("The difference of sum1 and sum2 is= "+diff);
    }
}
```

12. Write a java program called FunctionGrowth that prints a table of the values log N, N, N log N, N2, N3, and 2N for N = 16, 32, 64, ..., 2048. Use tabs (\t characters) to line up columns.

```
public class A4q12 {
    public static void main(String[] args) {
        int N;
        double s=0,t=0,p=0;
        System.out.println("log N,          N,    NlogN,          N2,       N3,    2N ");
        for(N=16;N<=2048;N=N*2)
        {
System.out.println(Math.log(N)+"\t"+N+"\t"+N*Math.log(N)+"\t"+Math.pow(N,2)+"\t"+
Math.pow(N, 3)+"\t"+Math.pow(2, N));
        }   }
}
```

13. An integer n is divisible by 9 if the sum of its digits is divisible by 9. Write a java program to display each digit, starting with the rightmost digit.

```
import java.util.*;
public class A4q13 {
    public static void main(String[] args) {
        Scanner sc=new Scanner(System.in);
        int n,r=0,s=0;
        System.out.println("Enter the number");
        n=sc.nextInt();
        int n1=n;
        while(n>0)
        {
            r=n%10;
            s+=r;
            n/=10;
            System.out.println(r+" ");
        }
        if(s%9==0)
            System.out.println(n1+" is divisible by 9");
        else
            System.out.println(n1+" is not divisible by 9");
    }
}
```

14. Write a java program to print largest power of two less than or equal to N.

```
import java.util.*;
public class A4q14 {
    public static void main(String[] args) {
        Scanner sc=new Scanner(System.in);
        int n;
        double s=0,res=0;
        System.out.println("Enter the value of n");
        n=sc.nextInt();
        int x=0;
        int v=1;;
        while(v<=n)
        {
             x=v;
            v=v*2;
        }
        System.out.println(x);
    }
}
```

15. Write a java program to print the below given pattern using while loop as well as for loop in two different programs.

```
* * * * *
* * * * *
* * * * *
* * * * *
```

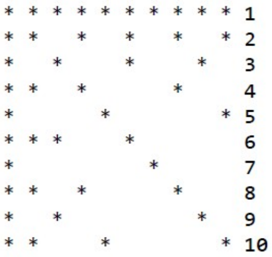
```
public class A4q15{
    public static void main(String[] args)
    {
        int i=1,j;
        while(i<=4)
        {
            j=1;
            while(j<=4)
            {
                System.out.print("* ");
                j++;
            }
            System.out.println();
            i++;
        }
    }
}
```

16. Write the java programs to print the following four patterns using for loop using four different programs.

```
 (a)              (b)              (c)              (d)
 *                1                1                1
 * *              1 2              2 2              2 3
 * * *            1 2 3            3 3 3            4 5 6
 * * * *          1 2 3 4          4 4 4 4          7 8 9 10
 * * * * *        1 2 3 4 5        5 5 5 5 5        11 12 13 14 15
```

```
public class A4q16a {

    public static void main(String[] args) {
        int i,j;
        for(i=1;i<=5;i++)
        {
            for(j=1;j<=i;j++)
            {
                System.out.print("* ");
            }
            System.out.println();
        }

    }

}

public class A4q16b {

    public static void main(String[] args) {
        int i,j;
        for(i=1;i<=5;i++)
        {
            for(j=1;j<=i;j++)
            {
                System.out.print(j+" ");
            }
            System.out.println();
        }
    }
}

public class A4q16c {

    public static void main(String[] args) {
        int i,j;
        for(i=1;i<=5;i++)
        {
            for(j=1;j<=i;j++)
            {
                System.out.print(i+" ");
            }
            System.out.println();
        }
    }
}
public class A4q16d {

    public static void main(String[] args) {
        int i,j,k=1;
        for(i=1;i<=5;i++)
        {
            for(j=1;j<=i;j++)
            {
                System.out.print(k);
                k++;
            }
            System.out.println();
        }
    }
}
```

17. Write a java program to print the following pattern using nested loops.



```
* * * * * * * * * * 1
* *   *   *   *   * 2
*   *     *     *   3
* *   *       *     4
*       *         * 5
* * *     *         6
*           *       7
* *   *       *     8
*   *           *   9
* *     *         * 10
```

```
import java.util.Scanner;
public class A4q17 {

    public static void main(String[] args) {
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter the value of n");
        int n=sc.nextInt();
        for(int i=1;i<=n;i++)
        {
            for(int j=1;j<=n;j++)
            {
                if((i%j==0)||(j%i==0))
                    System.out.print("* ");
                else
                    System.out.print("  ");
            }
            System.out.println(i);
        }
    }
}
```

Assignment-V

1. Write a java program that takes the value of N through keyboard and prints a table of the power of 2 that are less than or equal to 2 N .

```
import java.util.*;
public class A5q1
{
	public static void main(String[] args)
	{
		Scanner sc=new Scanner(System.in);
		int n,i;
		System.out.println("Enter the value of n:");
		n=sc.nextInt();
		for(i=0;i<=n;i++)
		{
			double r=Math.pow(2,i);
			System.out.println("2^"+i+"="+r);
		}
	}
}
```

2. Given a set of n numbers. Write a java program that to count the number of students that passed the examination. A pass is awarded for all marks of 40 and above.

```
import java.util.*;
public class A5q2
{
	public static void main(String[] args)
	{
		Scanner sc=new Scanner(System.in);
		int i,n,c=0;
		System.out.println("Enter the number of numbers u want to enter");
		n=sc.nextInt();
		for(i=1;i<=n;i++)
		{
			System.out.println("Enter the mark");
			int num=sc.nextInt();
			if ( num >=40)
				c++;
		}
		System.out.println("The pass students are"+c);

	}
}
```

3. Write a java program that displays all the numbers from 100 to 1,000, ten per line, that are divisible by 5 and 6. Numbers are separated by exactly one space.

```
public class A5q3
{
	public static void main(String[] args)
	{
		int count =0;
		for (int i=100; i<=1000; i++)
		{
			if (i % 5 ==0 && i % 6 ==0)
```

```
                count++;
            if (count % 11 ==0)
                System.out.println();
            else
                System.out.print(i + " ");
        }
    }
}
```

4. Write a java program that reads an unspecified number of integers, determines how many positive and negative values have been read, and computes the total and average of the input values (not counting zeros). Your program ends with the input 0. Display the average as a floating-point number.

```
import java.util.*;
public class A5q4
{
    public static void main(String[] args)
    {
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter the number (input ends if it is 0");
        int n=sc.nextInt();
        int sum = 0, c=0, pos=0, neg=0;
        if ( n ==0)
        {
            System.out.println("No number are entered except 0");
        }
        else
        {
            while (n !=0)
                {
              c++;
              sum = sum+n;
              if (n>0)
                  pos++;
              else   neg++;
              System.out.println("Enter the number (input ends if it is 0");
              n=sc.nextInt();
             }
                 double avg = sum/c;
                 System.out.println("The number of positives is"+ pos);
                 System.out.println("The number of negatives is"+neg);
                 System.out.println("The total is"+ sum);
                 System.out.println("The average is"+ avg);
        }
    }
}
```

5. Given a set of n numbers. Write a java program that adds these numbers and returns the resultant sum and compute the average. Assume n is greater than or equal to zero.

```
import java.util.*;
public class A5q5
{
    public static void main(String[] args)
    {
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter how many numbers you want to enter");
        int n=sc.nextInt();
        int sum = 0, m;
        for (int i = 0;  i <= n; i++)
         {
               System.out.println("Enter the number");
                 m=sc.nextInt();
                 sum = sum+m;
         }
          double avg = sum/n;
          System.out.println("The total is"+ sum);
          System.out.println("The average is"+ avg);
    }
}
```

6. Write a java program to compute the harmonic mean.

```
import java.util.*;
public class A5q6
{
    public static void main(String[] args)
    {
        Scanner sc=new Scanner(System.in);
        int n,a;
        double s=0,H=0;
        System.out.println("Enter the value of n");
        n=sc.nextInt();
        for(int i=1;i<=n;i++)
        {
            System.out.println("Enter the value of a["+i+"]");
            a=sc.nextInt();
            s+=(1/a);
        }
        H=n/s;
        System.out.println("The harmonic mean is= "+H);
    }
}
```

7. Write a java program to compute the sum of the first n terms (n>=1) of the series. S=1-3+5-7+9- ....

```
import java.util.*;
public class A5q7
{
    public static void main(String[] args)
    {
        Scanner sc=new Scanner(System.in);
        int i,sign=1,term=0,sum=0,m=1;
        System.out.println("Enter the range");
        int n=sc.nextInt();
        for(i=1;i<=n;i++)
        {
            term=1;
            term=sign*m;
            sum+=term;
            m+=2;
            sign*=-1;
        }
        System.out.println("The sum is= "+sum);
    }
}
```

8. Input a number n, write a java program to compute n factorial (written as n!) where n>=0.

```
import java.util.*;
public class A5q8
{
    public static void main(String[] args)
    {
        Scanner sc=new Scanner(System.in);
        int n,i,fact=1;
        System.out.println("enter the no whose factorial u want to find");
        n=sc.nextInt();
        for(i=1;i<=n;i++)
        {
            fact*=i;
        }
        System.out.println("Factorial of "+n+ " is = "+fact);
    }
}
```

9. For a given x and a given n, write a java program to compute x n /n!.

```
import java.util.*;
public class A5q9
{
    public static void main(String[] args)
    {
        Scanner sc=new Scanner(System.in);
        int x,n,fact=1;
        System.out.println("Enter the value of x");
        x=sc.nextInt();
        System.out.println("Enter the value of n");
        n=sc.nextInt();
        for(int i=1;i<=n;i++)
        {
            fact*=n;
        }
        double term=((Math.pow(x, n))/n);
        System.out.println(term);}
    }
}
```

10. Write a java program to evaluate the function sin(x) as defined by the infinite series expansion. sin (x) = x- x 3 /3! + x 5 /5! - x 7 /7! +... The acceptable error for computation is 10 -6 .

```
import java.util.Scanner;
public class A5q10
{
    public static void main(String[] args)
    {
        Scanner sc=new Scanner(System.in);
        double x;
        System.out.println("Enter the value of x in radians");
        x=sc.nextDouble();
        double tsin=x;
        double term=x;
        double error=0.000001;
        int i=1;
        while(Math.abs(term)>error)
        {
            i+=2;
            term=-term*(x*x)/(i*(i-1));
            tsin+=term;
        }
        System.out.println("The value of sin("+x+") = "+tsin);
    }
}
```

11. Write a java program to evaluate the function cos(x) as defined by the infinite series expansion. cos (x) =1- x 2 /2! + x 4 /4! - x 6 /6! +.... The acceptable error for computation is 10 - 6 .

```
import java.util.*;
public class A5q11
{
    public static void main(String[] args)
    {
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter the value of x in radians");
        double x=sc.nextDouble();
        double error=0.000001;
        double tcos=1;
        double term=1;
        int i=0;
        while(Math.abs(term)>error)
        {
            i+=2;
            term=-term*(x*x)/(i*(i-1));
            tcos+=term;
        }
        System.out.println("cos("+x+") = "+tcos);
    }
}
```

12. Assume that x is a positive variable of type double. Write a code fragment that uses the Taylor series expansion to set the value of sum to e x = 1 + x + x 2 /2! + x 3 /3! + ...

```
import java.util.*;
public class A5q12
{
    public static void main(String[] args)
    {
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter the value of x in radians");
        double x=sc.nextDouble();
        double sum = 0.0;
        double term = 1.0;
        for (int i = 1; sum != sum + term; i++)
        {
            sum = sum + term;
            term = term * x / i;
        }
        System.out.println("e("+x+") = "+sum);
    }
}
```

13. Write a java program to generate and print the first n terms of the Fibonacci sequence where n>=1.The first few terms are: 0, 1, 1, 2, 3, 5, 8, 13, ...
Example: If n=1, it will display as: Fibonacci Series is: 0
If n=2, it will display as: Fibonacci Series is: 0, 1
If n=3, it will display as: Fibonacci Series is: 0, 1, 1 ....
If n=10, it will display as: Fibonacci Series is: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34

```
import java.util.*;
public class A5Q13
{
    public static void main(String[] args)
    {
        Scanner sc=new Scanner(System.in);
        int n, t1 = 0, t2 = 1;
        System.out.println("Enter number of terms ");
        n=sc.nextInt();
        System.out.println("Fibonacci series");
        for (int i = 1; i <= n; ++i)
        {
            System.out.print(t1 + " ");
            int sum = t1 + t2;
            t1 = t2;
            t2 = sum;
        }
    }
}
```

14. Write a java program to generate and print the first n terms of the Fibonacci numbers using an efficient algorithm.

```
import java.util.Scanner;
public class A5q14
{
    public static void main (String args[])
    {
        Scanner sc=new Scanner(System.in);
        int a=0;
        int b=1;
        int i=2;
        System.out.println("Enter the range");
        int n=sc.nextInt();
        while(n>i)
        {
            System.out.println(a+"\n"+b);
            a=a+b;
            b=a+b;
            i+=2;
        }
        if(i%n==0)
            System.out.println(a+"\n"+b);
        else
            System.out.println(a);
    }
}
```

15. Suppose you save $100 each month into a savings account with the annual interest rate 5%. So, the monthly interest rate is 0.05 / 12 = 0.00417.

After the first month, the value in the account becomes

100 * (1 + 0.00417) = 100.417

After the second month, the value in the account becomes

(100 + 100.417) * (1 + 0.00417) = 201.252

After the third month, the value in the account becomes

(100 + 201.252) * (1 + 0.00417) = 302.507

and so on.

Write a java program that prompts the user to enter an amount (e.g., 100), the annual interest rate (e.g., 5), and the number of months (e.g., 6) and displays the amount in the savings account after the given month.

```
import java.util.Scanner;
public class A5q15
{
    public static void main(String[] args)
    {
        Scanner input = new Scanner(System.in);
        System.out.print("Enter an amount (e.g., 100): ");
        double amount = input.nextDouble();
        System.out.print("Enter the annual interest rate (e.g., 5): ");
        double annualInterestRate = input.nextDouble();
        System.out.print("Enter the number of months (e.g., 6): ");
        int months = input.nextInt();

        double monthlyInterestRate = annualInterestRate / 1200;
        double compoundValue = 0; // Accumulates compount value
        for (int m = 1; m <= months; m++)
        {
            compoundValue = (amount + compoundValue) * (1 + monthlyInterestRate);
        }
        System.out.println("Amount in savings account after " + months + " months" +
                                                        compoundValue);
    }
}
```

16. Write a java program that accepts a positive integer n and reverses the order of its digits.

```
import java.util.Scanner;
public class A5q16
{
    public static void main(String[] args)
    {
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter a number");
        int n=sc.nextInt();
        int rev=0;
        while(n>0)
        {
            int rem=n%10;
            rev=(rev*10)+rem;
            n/=10;
        }
        System.out.println("Reverse is = "+rev);
    }
}
```

17. Write a java program to compute the square root of a number using Newton’s method.

```
import java.util.*;
public class A5q17
{
    public static void main(String[] args)
    {
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter the number whos square root is to be found");
        int n=sc.nextInt();
        double g1,g2;
        g2=n/2;
        do
        {
            g1=g2;
            g2=(g1+(n/g1))/2.0;
        } while (Math.abs(g1-g2)>0.000001);
        double root=g2;
        System.out.println("The root of "+ n+" is ="+root);
    }
}
```

18. Using Newton’s method, write a java program that takes integers N and k as command-line arguments and prints the k th root of N.

```
import java.util.*;
public class A5q18
{
    public static void main(String[] args)
    {
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter the number to find the given root");
        int n=sc.nextInt();
        System.out.println("Enter the root");
        int k=sc.nextInt();
        double g1,g2;
        g2=n/2;
        do
        {
            g1=g2;
            g2=((k-1)*g1+n/Math.pow(g1,k-1))/k;
        }while(Math.abs(g1-g2)>0.000001);
        System.out.println("The square root of "+n+" is = "+g2);
    }
}
```

19. Write a java program that puts the binary representation of a positive integer N into a String s.

```
import java.util.*;
public class A5q19
{
	public static void main(String[] args)
	{
		Scanner sc=new Scanner(System.in);
		System.out.println("Enter a number");
		int n=sc.nextInt();
		String str="";
		while(n>0)
		{
			int rem=n%2;
			str=rem+str;
			n=n/2;
		}
		System.out.println(str);
	}
}
```

20. Write a java program that reads an integer and displays all its smallest factors in increasing order. For example, if the input integer is 120, the output should be as follows: 2, 2, 2, 3, 5.

```
import java.util.Scanner;
public class A5q20
{
	public static void main(String[] args)
	{
		Scanner input = new Scanner(System.in);
		System.out.print("Enter an integer: ");
		int number = input.nextInt();
		int index = 2;
		while (number / index != 1)
		{
			if (number % index  == 0)
			{
				System.out.print(index + ", ");
				number /= index;
			}
			else
				index++;
		}
		System.out.println(number + ".");
	}
}
```

21. Write a java program GCD that finds the greatest common divisor (gcd) of two integers using Euclid's algorithm, which is an iterative computation based on the following observation: if x is greater than y, then if y divides x, the gcd of x and y is y; otherwise, the gcd of x and y is the same as the gcd of x % y and y.

```
import java.util.*;
public class A5q21
```

```
{
    public static void main(String[] args)
    {
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter a number");
        int x=sc.nextInt();
        System.out.println("Enter the second number");
        int y=sc.nextInt();
        while(y>0)
        {
            int rem=x%y;
            x=y;
            y=rem;
        }
        System.out.println("GCD = "+x);
    }
}
```

22. Write a java program to check a number n is prime or not. The number to be inputted through keyboard.

```
import java.util.*;
public class A5q22
{
    public static void main(String[] args)
    {
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter a number");
        int n=sc.nextInt();
        int flag=0;
        for(int i=2;i<n;i++)
        {
            if(n%i==0)
            {
                flag=1;
                break;
            }
        }
        if(flag==0)
            System.out.println(n+" is prime");
        else
            System.out.println(n+" is not prime");
    }
}
```

23. Write a java program called PrimeCounter that takes a commandline argument N and finds the number of primes less than or equal to N.

```
public class A5q23
{
    public static void main(String[] args)
    {
        System.out.println("Enter the range");
        int n=Integer.parseInt(args[0]);
        int flag;
        for(int i=2;i<=n;i++)
        {
            flag=0;
            for(int j=2;j<=i;j++)
            {
                if(i%j==0)
                flag++;
            }
            if(flag==1)
                System.out.println(i+" is prime");
            else
                System.out.println(i+" is not prime");
        }
    }
}
```

24. Write a java program that takes a command-line argument N and prints out all integers less than or equal to N that can be expressed as the sum of two cubes in two different ways. In other words, find distinct positive integers a, b, c, and d such that $a^3 + b^3 = c^3 + d^3$. Use four nested for loops.

```
public class A5Q24
{
    public static void main(String[] args)
    {
        int n = Integer.parseInt(args[0]);
           for (int a = 1; a <= n; a++)
        {
               int a3 = a*a*a;
               if (a3 > n) break;
               for (int b = a; b <= n; b++)
            {
                 int b3 = b*b*b;
                 if (a3 + b3 > n) break;
                 for (int c = a + 1; c <= n; c++)
              {
                   int c3 = c*c*c;
                   if (c3 > a3 + b3) break;
                   for (int d = c; d <= n; d++)
                 {
                     int d3 = d*d*d;
                     if (c3 + d3 > a3 + b3) break;
                     if (c3 + d3 == a3 + b3)
                 {
                       System.out.print((a3+b3) + " = ");
                       System.out.print(a + "^3 + " + b + "^3 = ");
                       System.out.print(c + "^3 + " + d + "^3");
                       System.out.println();
                     }
                   }
                 }
               }
        }   }
```

Assignment-6

1. **Design a Simple Calculator using methods in java containing the following functionalities, namely, with Addition, Subtraction, Multiplication, Remainder, Division and Square Root. The signature of the methods are given below.**
   public static int additionSimple(int x, int y)
   two inputs, x and y. Return the result of adding x to y.
   public static int subtractionSimple(int x, int y)
   two inputs, x and y. Return the result of subtracting x from y i.e y-x.
   public static int multiplicationSimple(int x, int y)
   two inputs, x and y. Return the result of multiplying x to y i.e. x*y.
   public static double divisionSimple(int x, int y)
   two inputs, x and y. Return the result of dividing y by x. Please check whether x is zero before dividing.
   public static int remainderSimple(int n, int m)
   Please make sure that remainderSimple() takes two inputs, namely, a number (int) n and a number (int) m. the method should return the remainder of n divided by m.
   public static double squareRootSimple(int n)
   Takes one input, namely a number n, and returns the square root of the number. The return should be double. Please kindly make sure that the number n is positive.

```
import java.util.Scanner;
public class A6Q1
{
       public static void main(String[] args)
       {
              Scanner input = new Scanner(System.in);
              int a,b;
              System.out.println("Enter the numbers");
              a = input.nextLine();
              b = input.nextLine();

              int add=additionSimple(a,b);
              int sub=substractionSimple(a,b);
              int mul=multiplicationSimple(a,b);
              double div= divisionSimple(a,b);
              int rem= remainderSimple(a,b);
              double sqr=squareRootSimple(a);
              System.out.println(add+ “ “+sub+” “ + mul+” “ + div+ “ “+rem+” “+ sqr);
       }
       public static int additionSimple( int a, int b)
       {   return (a+b);    }
       public static int substractionSimple(int a, int b)
       {  return (a-b);    }
       public static int multiplicationSimple(int a, int b)
       {  return (a*b);    }
       public static double divisionSimple(int a, int b)
       {  return (a/b);    }
       public static int remainderSimple(int a, int b)
       {  return (a%b);    }
       public static double squareRootSimple(int a)
```

```
          {  return (Math.sqrt(a));    }
      }
```

2. **A pentagonal number is defined as n(3n–1)/2 for n = 1, 2, . . ., and so on. Therefore, the first few numbers are 1, 5, 12, 22, ... Write a method with the following header that returns a pentagonal number: public static int getPentagonalNumber(int n)**
   Write a java program that uses this method to display the first 100 pentagonal numbers with 10 numbers on each line.

```
public class A6Q2
{
        public static void main(String[] args)
        {
                int NUMBER_PER_LINE = 10; // Display 10 number per line
                System.out.println("The first 100 pentagonal numbers, ten per line: ");
                for (int i = 1; i <= 100; i++)
                {
                        if (i % NUMBER_PER_LINE == 0)
                                System.out.println(“   ”+ );
                        else
                                System.out.print("   "+  getPentagonalNumber(i));
                }
        }
        public static int getPentagonalNumber(int n)
        {
                        return (n * (3 * n - 1)) / 2;
        }
}
```

3. **Write the methods with the following headers**
   // Return the reversal of an integer, i.e., reverse(456) returns 654
   public static int reverse(int number)
   // Return true if number is a palindrome
   public static boolean isPalindrome(int number)
   Use the reverse method to implement isPalindrome. A number is a palindrome if its reversal is the same as itself. Write a java program that prompts the user to enter an integer and reports whether the integer is a palindrome.

```
public class A6Q3
{
        public static void main(String[] args)
        {
                Scanner input = new Scanner(System.in);
                System.out.print("Enter an integer: ");
                int number = input.nextInt();
                boolean p = isPalindrome (number);
                if (p)
                        System.out.println(" is palindrome");
                else
                        System.out.println(" is not a palindrome.");
        }
        public static boolean isPalindrome(int number)
        {
                if ( number == reverse(number))
                return true
        else  return false;
     }
    public static int reverse(int number)
    {
        int d, r=0;
        while (number > 0)
        {
            d = number %10;
            r = r*10 + d;
            number = number/10;
        }
        return r;
    }
}
```

4. **Write a method that returns the number of days in a year using the following header:**
   public static int numberOfDaysInAYear(int year)
   Write a java program that displays the number of days in year from 2000 to 2020.

```
public class A6Q4
{
    public static void main(String[] args)
    {
        for (int year = 2000; year <= 2020; year++)
        {
            System.out.println(year + "         " + numberOfDaysInAYear(year));
        }
    }
    public static int numberOfDaysInAYear(int year)
    {
        if (year % 400 == 0 || (year % 4 == 0 && year % 100 != 0))
                return 366;
        else
                return 365;
    }
}
```

5. **Write a method to test whether a number is prime or not using the following header.**
   public boolean isPrime(int number)
   Write a java program by using the above method to find the number of prime numbers less than 10000.

```
public class A6Q5
{
    public static void main(String[] args)
    {
        int c = 0;
        for (int i = 1; i < 10000; i++)
        {
            if (isPrime(i))
                System.out.println( i );
        }
    }
```

```
        public static boolean isPrime(int n)
        {
            for (int i = 2;  i <= n/2; i++)
            {
                if (n % i == 0)
                {
                    return false;
                }
            }
            return true;
        }
    }
```

6. **A palindromic prime is a prime number and also palindromic.
   For example, 131 is a prime and also a palindromic prime, as are 313 and 757. Write a java program using method that displays the first 100 palindromic prime numbers. Display 10 numbers per line, separated by exactly one space, as follows:
   2 3 5 7 11 101 131 151 181 191
   313 353 373 383 727 757 787 797 919 929**

```
    public class A6Q6
    {
        public static void main(String[] args)
        {
            int n =100;
            int NUMBERS_PER_LINE = 10;
            int count = 0;
            int no = 2;
            while (count < n)
            {
                if (isPalindrome(no)  && isPrime(no))
                {
                    System.out.print(no);
                    if ( count %  NUMBERS_PER_LINE == 0)
                         System.out.println();
                    count++;
                }
                no++;
            }
        }
        public static boolean isPrime(int n)
        {
            if (n==2)   return true;
            for (int i = 2;  i <= n/2; i++)
            {
                if (n % i == 0)
                {
                    return false;
                }
            }
            return true;
        }
        public static boolean isPalindrome(int n)
        {
            int d, r=0, m=n;
            while (n > 0)
            {
                d = n %10;
                r = r*10 + d;
                n = n/10;
            }
            if ( m == r))
                return true
            else  return false;
        }
    }
```

7. **win primes are a pair of prime numbers that differ by 2. For example, 3 and 5 are twin primes, 5 and 7 are twin primes, and 11 and 13 are twin primes. Write a java program using method to find all twin primes less than 1,000.**

```
public class A6Q7
{
        public static void main(String[] args)
        {
            for (int i = 1; i < 1000; i++)
            {
                if (isPrime(i) && isPrime(i+2))
                    System.out.println( i +"    " + (i+2));
            }
        }
        public static boolean isPrime(int n)
        {
            for (int i = 2;  i <= n/2; i++)
            {
                if (n % i == 0)
                {
                    return false;
                }
            }
            return true;
        }
}
```

8. **A regular polygon is an n-sided polygon in which all sides are of the same length and all angles have the same degree (i.e., the polygon is both equilateral and equiangular). The formula for computing the area of a regular polygon is**

   Area =$(n \times s^{2})/(4 \times \tan (\pi/n))$

   Write a method that returns the area of a regular polygon using the following header:

   public static double area(int n, double side)Write a java main method that prompts the user to enter the number of sides and the side of a regular polygon and displays its area. Here is a sample run:

   Enter the number of sides: 5
   Enter the side: 6.5
   The area of the polygon is 72.69017017488385

```
import java.util.Scanner;
```

```
public class A6Q8
{
    public static void main(String[] args)
    {
        Scanner input = new Scanner(System.in);
        System.out.print("Enter the number of sides: ");
        int n = input.nextInt();
        System.out.print("Enter the side: ");
        double side = input.nextDouble();
        System.out.println("The area of the polygon is " + area (n, side));
    }
    public static double area(int n, double side)
    {
        return (n * Math.pow(side, 2) / (4 * Math.tan(Math.PI / n)));
    }
}
```

9. **Given the two fractions a/b and c/d, write a java program using method that will compute their sum in terms of the smallest common denominator.**

```
import java.util.Scanner;
public class A6Q9
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter two fractional numbers (a/b) and (c/d)”);
        int a = sc.nextInt();
        int b = sc.nextInt();
        int c = sc.nextInt();
        int d = sc.nextInt();
        scd ( a, b, c, d);
    }
    public static void scd (int a, int b, int c, int d)
    {
        int x = lcm (b , d);
        a = a* (x / b);
        c = c * (x / d);
        int sum = a + c;
        System.out.print(“sum = ” + sum);
    }
    public static int lcm ( int b, int d)
    {
        for (int i = 1; i <= Math.abs (b) && Math.abs (d); i++)
        {
            if ( b % i ==0 && d % i ==0)
                    gcd = i;
        }
        int x = (b * d) / gcd;
        return x;
    }
}
```

10. Write a method that prints characters using the following header:
public static void printChars(char ch1, char ch2, int numberPerLine)
This method prints the characters between ch1 and ch2 with the specified numbers per line. Write a java program that prints ten characters per line from 1 to Z. Characters are separated by exactly one space.

```
public class A6Q10
{
      public static void main(String[] args)
      {
            int NUMBER_OF_CHARS_PER_LINE= 10;
            char ch1 = '1',  ch2 = 'Z';
            System.out.println("Characters per 1 to Z");
            printChars(ch1, ch2, NUMBER_OF_CHARS_PER_LINE);
            System.out.println();
      }
      public static void printChars(char ch1, char ch2, int num)
      {
            int count =1;
            for (char ch = ch1; ch <= ch2; ch++)
            {
                  count++;
                  if (count % num == 0)
                        System.out.println(ch);
                  else
                        System.out.print(ch + " ");
            }
      }
}
```

11. Write a method that finds the number of occurrences of a specified character in a string using the following header:
public static int count(String str, char a)
For example, count ("Welcome", 'e') returns 2. Write a java program that prompts the user to enter a string followed by a character and displays the number of occurrences of the character in the string.

```
import java.util.Scanner;
public class A6Q11
{
      public static void main(String[] args)
      {
            Scanner input = new Scanner(System.in);
            System.out.print ("Enter a string”);
            String string = input.nextLine();
            System.out.println ( "The number of occurrences = " + count (string, ‘e’));
      }
      public static int count(String string, char a)
      {
            int count = 0;
            for (int i = 0; i < string.length(); i++)
            {
                if (a == string.charAt(i))
                    count++;
            }
            return count;
        }
    }
```

12. Write java method called count accepts a string as input and returns the number of vowels in it.

```
public class A6Q12
{
    public static void main(String[] args)
    {
        Scanner input = new Scanner(System.in);
        System.out.print("Enter a string: ");
        String string = input.nextLine();
        string = string.toLower();
        System.out.println( "The number of vowels in the string " +count (string));
    }
    public static int count(String s)
    {
        int v = 0;
        for (int i = 0; i < s.length(); i++)
        {
            if (s.charAt (i) == ‘a’ || s.charAt (i) == ‘e’ || s.charAt (i) == ‘i’ ||  s.charAt (i)
            == ‘o’ || s.charAt (i) == ‘u’ ||)
                v++; // Increment the number of letters
        }
        return v;
    }
}
```

13. Write a java method to check a string is palindrome or not.

```
import java.util.Scanner;
public class A6Q13
{
    public static void main(String[] args)
    {
        Scanner input = new Scanner(System.in);
        System.out.print("Enter string: ");
        String s = input.nextLine();
        String rev = reverse (s);
        if (s. equals(rev))
            System.out.println("Palindrome");
        else
            System.out.println("not a palindrome");
    }
    public static String reverse (String s)
    {
        String rev;
        for ( int i = s.length()-1; i>= 0; i--)
            rev = rev + s.charAt(i);
        return rev;
    }
}
```

14. **Some websites impose certain rules for passwords. Write a method that checks whether a string is a valid password. Suppose the password rules are as follows:**
    A password must have at least eight characters.
    A password consists of only letters and digits.
    A password must contain at least two digits.
    Write a program that prompts the user to enter a password and displays Valid Password if the rules are followed or Invalid Password otherwise.

```
  import java.util.Scanner;
  public class A6Q14
  {
        public static void main(String[] args)
        {
              Scanner input = new Scanner(System.in);
              System.out.print("Enter a password: ");
              String password = input.nextLine();
              if (isValidPassword(password))
                    System.out.println("Valid Password");
              else
                    System.out.println("Invalid Password");
        }
        public static boolean isValidPassword (String password)
        {
              int LENGTH_OF_VALID_PWD = 8;
              int MIN_NUMBER_OF_DIGITS = 2;
              boolean validPassword =  isLengthValid(password, LENGTH_OF_VALID_PWD)
              && isOnlyLettersAndDigits(password)
              && hasNDigits(password, MIN_NUMBER_OF_DIGITS);

              return validPassword;
        }
        public static boolean isLengthValid(String password, int validLength)
        {
              return password.length() >= validLength;
        }
        public static boolean isOnlyLettersAndDigits(String password)
        {
              for (int i = 0; i < password.length(); i++)
              {
                    if (!Character.isLetterOrDigit(password.charAt(i)))
                    {
                          return false;
                    }
              }
              return true;
        }
        public static boolean hasNDigits(String password, int n)
        {
              int numberOfDigits = 0;
              for (int i = 0; i < password.length(); i++)
        {
            if (Character.isDigit(password.charAt(i)))
            {
                numberOfDigits++;
            }
            if (numberOfDigits >= n)
            {
                return true;
            }
        }
        return false;
    }
}
```

15. Write a java program to calculate the area of triangle, square, circle, rectangle by using method overloading.

```
import java.util.Scanner;
    public class A6Q15
    {
        public static void main(String[] args)
        {
            Scanner sc = new Scanner(System.in);
            System.out.print("Enter the values”);
            int a = sc.nextInt();
            int b = sc.nextInt();
            int c = sc.nextInt();

            double circle = area ( a);
            double triangle = area (a, b, c);
            int square = area (a);
            int rectangle = area (a,b);
            System.out.println(circle+ “   ”+ square+ “   ”+ rectangle+ “   ” + triangle);
        }
        public static int area (int a, int b)
        {
            return (a*b);
        }
        public static int area (int a)
        {
            return (a*a);
        }
        public static double area (int a, int b, int c)
        {
            int s = (a+b+c)/2;
            double ans = Math.sqrt (s* (s-a)* (s-b) * (s-c));
            return ans;
        }
        public static double area (int a)
        {
            return (Math.PI * a * a);
        }
    }
```

Programming Assignment-VII
(Single-Dimensional Arrays)

1. Write a java program to create an array of size N and store the random values in it and find the sum and average.

```
import java.util.Scanner;
public class A7Q1
{
	public static void main(String[] args)
	{
		Scanner input = new Scanner(System.in);
		System.out.println(“ enter size of array”);
		int N= input.nextInt();
		int[] numbers = new int[N];
		int sum =0, i;
		double avg;
		System.out.print("Enter ten numbers: ");
		for (int i = 0; i < N; i++)
		{
			numbers[i] = input.nextInt();
			sum = sum + numbers[i];
		}
		avg = sum / N;
		System.out.println (“ sum = ” + sum + “  Average = ” + avg);
	}
}
```

2. Write a java program using an array that reads the integers between 1 and 100 and counts the occurrences of each. Assume the input ends with 0. Here is a sample run of the program:
Enter the integers between 1 and 100: 2 5 6 5 4 3 23 43 2 0
2 occurs 2 times
3 occurs 1 time
4 occurs 1 time
5 occurs 2 times
6 occurs 1 time
23 occurs 1 time
43 occurs 1 time
Note that if a number occurs more than one time, the plural word “times” is used in the output.

```
import java.util.Scanner;
public class A7Q2
{
	public static void main(String[] args)
	{
		int[] counts = new int[100];
		System.out.print("Enter the integers between 1 and 100: ");
		count(counts);
		for (int i = 0; i < counts.length; i++)
		{
		  if (counts[i] > 0)
```

```
            System.out.println((i + 1) + " occurs " + counts[i] +" time" + (counts[i] > 1 ? "s" : ""));
        }
    }
    public static void count(int[] counts)
    {
        Scanner input = new Scanner(System.in);
        int num;
        do
        {
            num = input.nextInt();
            if (num >= 1 && num <= 100)
                counts[num - 1]++;
        } while (num != 0);
    }
}
```

3. Input 10 integers from the keyboard into an array. The number to be searched is entered through the keyboard by the user. Write a java program to find if the number to be searched is present in the array and if it is present, display the number of times it appears in the array.

```
import java.util.*;
public class A7Q3
{
    public static void main(String[] args)
    {
        int a[] = { 1, 6, 3, 4, 5, 6, 7, 8, 6, 10};
        int key = 6, count = 0;
        for (int i = 0; i < a.length; i++)
        {
            if (key == array[i])
                count++;
        }
        System.out.println (“ The number is found” + count + “ times ”);
    }
}
```

4. Write a method that finds the smallest element in an array of double values using the following header:
public static double min(double[] array)
Write a java program that prompts the user to enter ten numbers, invokes this method to return the minimum value, and displays the minimum value. Here is a sample run of the program:
Enter ten numbers: 1.9 2.5 3.7 2 1.5 6 3 4 5 2
The minimum number is: 1.5

```
import java.util.Scanner;
public class A7Q4
{
    public static void main(String[] args)
    {
        Scanner input = new Scanner(System.in);
        double[] numbers = new double[10];
        System.out.print("Enter ten numbers: ");
        for (int i = 0; i < numbers.length; i++)
        {
            numbers[i] = input.nextDouble();
        }
        System.out.println("The minimum number is: " + min(numbers));
    }
    public static double min(double[] array)
    {
        double min = array[0];
        for (double i: array)
        {
            if (i < min)
                min = i;
        }
        return min;
    }
}
```

5. Write a java program to find the second largest value in an array of n elements.

```
import java.util.Arrays;
public class A7Q5
{
    public static void main(String args[])
    {
        int array[] = {10, 20, 25, 63, 96, 57};
        int size = array.length;
        Arrays.sort(array);
        System.out.println("sorted Array ::"+Arrays.toString(array));
        int res = array[size-2];
        System.out.println("2nd largest element is ::"+res);
    }
}
```

6. Write a java program that implements the array reversal algorithm suggested in Note 1.
Note 1: There is a simpler algorithm for array reversal that starts out with two indices, i=0 and j=n-1. With each iteration i is increased and j is decreased for i<j.

```
public class A7Q6
{
    public static void main(String args[])
    {
        int array[] = {10, 20, 25, 63, 96, 57};
        int size = array.length;
        int i =0, j = i-1;
        while (i < j)
        {
            int temp = array [i];
            array [i] = array [j];
            array [j] = temp;
```

```
                i++;
                j- -;
            }
            System.out.println (“ Reversed array”);
            for (int i = 0; i < array.length; i++)
                    System.out.println(array[i] + “   ”);
        }
}
```

7. Write a java program to convert a decimal integer to its corresponding octal representation.

```
import java.util.Scanner;
public class A7Q7
{
    public static void main(String args[])
    {
        int decimal, i=1, j;
        int oct[] = new int[100];
        Scanner sc = new Scanner(System.in);
        System.out.print("Input a Decimal Number: ");
       decimal= sc.nextInt();
       while(decimal != 0)
       {
         oct[i++] = decimal % 8;
         decimal = decimal / 8;
       }
       System.out.print("Octal number is: ");
       for(j=i-1; j>0; j--)
             System.out.print(oct_num[j]);
     }
}
```

8. Design and develop a menu driven java program for the following array operations.
a. Create an array of N integers
b. Display the array elements
c. Insert an element at specific position
d. Delete an element at a given position
e. Exit

```
public class A7Q8
{
      public static void main(String args[])
      {
            int a[]=new int [10];
            int choice;
            do
            {
              System.out.println("\n\n--------Menu-----------\n");
              System.out.println("1.Create\n");
              System.out.println("2.Display\n");
              System.out.println("3.Insert\n");
              System.out.println("4.Delete\n");
              System.out.println("5.Exit\n");

              System.out.println("\nEnter your choice:\t");
              choice = sc.nextInt();
              switch(choice)
              {
                  case 1:     a[] = create();  break;
                  case 2:     display(a);  break;
                  case 3:     a[] = insert(a);   break;
                  case 4:     a[] = del(a);   break;
                  case 5:     System.exit (0);   break;
                  default:
                              System.out.println("\nInvalid choice:\n");
              }
            }while(choice!=5);
        }
        public static int[] create()
        {
          Scanner sc = new Scanner (System.in);
          int a[] = new int [5];
          System.out.println("Enter the elements for the array:");
          for(i=0 ; i<4 ; i++)
                a[i] = sc. NextInt();
      return a;
        }
        public static void display (int a[])
        {
      int i;
      System.out.println("The array elements are:");
      for(i=0 ; i<4 ; i++)
           System.out.print (a[i] + “   ”);
        }
        public static int[] insert (int a[])
        {
         Scanner sc = new Scanner (System.in);
         System.out.println("Enter the position for the new element:");
         int pos = sc.nextInt();
         System.out.println("Enter the element to be inserted :");
         int val = sc.nextInt();
         for (int i=a.length ; i>=pos ; i--)
         {
              a[i+1]=a[i];
         }
         a[pos]=val;
         return a;
        }
        public static int[] del (int a[])
        {
      Scanner sc = new Scanner (System.in);
      System.out.println("Enter the position of the element to be deleted:");
      int pos = sc.nextInt();
      int val=a[pos], n = a.length;
```

```
         for(int i=pos ; i<n-1 ; i++)
         {
                a[i]=a[i+1];
         }
         n=n-1;
         System.out.println("The deleted element is =" + val);
        return a;
}
}
```

9. You can compute the standard deviation with the following formula; you have to store the individual numbers using an array, so that they can be used after the mean is obtained. Your program should contain the following methods:

/ Compute the deviation of double values */**
public static double deviation(double[] x)
/ Compute the mean of an array of double values */**
public static double mean(double[] x)
Write a java program that prompts the user to enter ten numbers and displays the mean and standard deviation, as shown in the following sample run:

```
Enter ten numbers: 1.9 2.5 3.7 2 1 6 3 4 5 2
The mean is 3.11
The standard deviation is 1.55738
```

```
import java.util.Scanner;
public class A7Q9
{
       public static void main(String[] args)
       {
              Scanner input = new Scanner(System.in);
              double[] numbers = new double[10];
              System.out.print("Enter ten numbers: ");
              for (int i = 0; i < numbers.length; i++)
                     numbers[i] = input.nextDouble();
              System.out.println("The mean is " + mean(numbers));
              System.out.println("The standard deviation is " + deviation(numbers));
       }
       public static double deviation(double[] x)
       {
              double deviation = 0;
              double mean = mean(x);
              for (double e = 0; e < x.length; e++)
              {
                     deviation += Math.pow(e - mean, 2);
              }
              return (Math.sqrt(deviation / (x.length - 1)));
       }
       public static double mean(double[] x)
       {
              double mean = 0;
              for (double e = 0; e < x.length; e++)
              {
                     mean += e;
```

```
        }
        return (mean / x.length);
    }
}
```

10. Write a method that returns a new array by eliminating the duplicate values in the array using the following method header:
public static int[] eliminateDuplicates(int[] list)
Write a java program that reads in ten integers, invokes the method, and displays the result. Here is the sample run of the program:
Enter ten numbers: 1 2 3 2 1 6 3 4 5 2
The distinct numbers are: 1 2 3 6 4 5

```
import java.util.Scanner;
public class A7Q10
{
    public static void main(String[] args)
    {
        Scanner input = new Scanner(System.in);
        int[] numbers = new int[10];
        System.out.print("Enter ten numbers: ");
        for (int i = 0; i < numbers.length; i++)
            numbers[i] = input.nextInt();

        eliminateDuplicates(numbers);
    }
    public static void eliminateDuplicates(int arr[])
    {
        if (n==0 || n==1)
            return n;
        int[] temp = new int[n];
        int j = 0;
        for (int i=0; i<n-1; i++)
        {
            if (arr[i]!=arr[i+1])
            {
                temp[j++] = arr[i];
            }
        }
        temp[j++] = arr[n-1];
        for (int i=0; i<j; i++)
        {
           System.out.print (temp[i]+ “  ”);
        }
    }
}
```

11. Write a sort method that uses the bubble-sort algorithm. The bubble sort algorithm makes several passes through the array. On each pass, successive neighbouring pairs are compared. If a pair is not in order, its values are swapped; otherwise, the values remain unchanged. The

technique is called a bubble sort or sinking sort because the smaller values gradually “bubble” their way to the top and the larger values “sink” to the bottom. Write a java program that reads in ten double numbers, invokes the method, and displays the sorted numbers.

```
import java.util.Scanner;
public class A7Q11
{
    public static void main(String[] args)
    {
        Scanner input = new Scanner(System.in);
        int[] numbers = new int[10];
        System.out.print("Enter ten numbers: ");
        for (int i = 0; i < numbers.length; i++)
            numbers[i] = input.nextInt();

        bubbleSort(numbers);
    }
    public static void bubbleSort(double[] list)
    {
        int n = list.length;
        int temp =0;
        for (int i = 0; i < n; i++)
        {
            for (int j = 1; j < n-i ; j++)
            {
                if (list[j-1] < list[j])
                {
                    temp = list[j-1];
                    list[j-1] = list[j];
                    list[j] = temp;
                }
            }
        }
        for (int i = 0; i < list.length; i++)
            System.out.print (list[i] +“   ”);
    }
}
```

12. The selection-sort method repeatedly finds the smallest number in the current array and swaps it with the first. Write a java program that reads in ten integer values, invoke the method, and displays the sorted elements.

```
import java.util.Scanner;
public class A7Q12
{
    public static void main(String[] args)
    {
        Scanner input = new Scanner(System.in);
        System.out.print("Enter list: ");
        int n = input.nextInt();
        int[] list = new int[n];
        for (int i = 0; i < list.length; i++)
```

```
            list[i] = input.nextInt();
        selectionSort(list);
    }
    public static void selectionSort(int[] array)
    {
        for (int i = 0; i < array.length-1; i++)
        {
            int index = i;
            for (int j = i + 1; j < array.length; j++)
            {
                if (array[j] < array[index])
                {
                    minIndex = j;
                }
            }
            int temp = array[index];
            array[Index] = array[i];
            array[i] = temp;
        }
        for (int i = 0; i < array.length; i++)
            System.out.print (array[i] +“   ”);
    }
}
```

13. Write the following method that returns true if the list is already sorted in increasing order.

public static boolean isSorted(int[] list)

Write a java program that prompts the user to enter a list and displays whether the list is sorted or not. Here is a sample run. Note that the first number in the input indicates the number of the elements in the list. This number is not part of the list. Here is the sample run:

```
Enter list:  8 10 1 5 16 61 9 11 1
The list is not sorted
Enter list: 10 1 1 3 4 4 5 7 9 11 21
The list is already sorted
```

```
import java.util.Scanner;
public class A7Q13
{
    public static void main(String[] args)
    {
        Scanner input = new Scanner(System.in);
        System.out.print("Enter list: ");
        int n = input.nextInt();
        int[] list = new int[n];
        for (int i = 0; i < list.length; i++)
            list[i] = input.nextInt();

        System.out.println(
            "The list is " + (isSorted(list) ? "already " : "not ") + "sorted");
    }
    public static boolean isSorted(int[] list)
    {
        for (int i = 0; i < list.length - 1; i++)
        {
            if (list[i] > list[i + 1])
                return false;
        }
        return true;
    }
}
```

14. Write a java program that randomly generates an array of 100 integers and a key. Estimate the execution time of invoking the linearSearch method. Sort the array and estimate the execution time of invoking the binarySearch method. You can use the following code template to obtain the execution time:
long startTime = System.currentTimeMillis();
perform the task;
....
....
long endTime = System.currentTimeMillis();
long executionTime = endTime - startTime;

```
import java.util.*;
public class A7Q14
{
    public static void main(String[] args)
    {
        int num;
        int[] numbers = new int[10000];
        for (int i = 0; i < numbers.length; i++)
        {
            numbers[i] = getRandomInt();
        }
        int key = getRandomInt();

        int searchResult;
        long startTime = System.currentTimeMillis();
        searchResult = linearSearch(numbers, key);
        long endTime = System.currentTimeMillis();
        long executionTime = endTime - startTime;
        System.out.println(
            "Execution time of invoking the linearSearch in milliseconds: "
            + executionTime);

        // Sort array
        Arrays.sort(numbers);

        // Estimate the execution time of invoking the linearSearch
        startTime = System.currentTimeMillis();
        searchResult = binarySearch(numbers, key);
        endTime = System.currentTimeMillis();
        executionTime = endTime - startTime;
        System.out.println(
            "Execution time of invoking the binarySearch in milliseconds: "
            + executionTime + " ");
    }

    public static int getRandomInt()
    {
        return 1 + (int)(Math.random() * 100000);
    }

    /** linearSearch */
    public static int linearSearch(int[] array, int key)
    {
        for (int i = 0; i < array.length; i++)
        {
            if (key == array[i])
                return i;
        }
        return -1;
    }

    /** binarySearch */
    public static int binarySearch(int[] array, int key)
    {
        int low = 0;
        int high = array.length - 1;

        while (high >= low)
        {
            int mid = (low + high) / 2;
            if (key < mid)
                high = mid - 1;
            else if (key == mid)
                return mid;
            else
                low = mid + 1;
        }
        return -low -1;
    }
}
```

15. Write the following method that partitions the list using the first element, called a pivot.
public static int partition(int[] list)
After the partition, the elements in the list are rearranged so that all the elements before the Pivot are less than or equal to the pivot and the elements after the pivot are greater than the pivot. The method returns the index where the pivot is located in the new list. For example, suppose the list is {5, 2, 9, 3, 6, 8}. After the partition, the list becomes {3, 2, 5, 9, 6, 8}. Implement the method in a way that takes at most list.length comparisons. Write a java program that prompts the user to enter a list and displays the list after the partition.
Here is a sample run. Note that the first number in the input indicates the number of the elements in the list. This number is not part of the list.
Enter list: 8 10 1 5 16 61 9 11 1
After the partition, the list is 9 1 5 1 10 61 11 16

```
import java.util.Scanner;
public class A7Q15
{
    public static void main(String[] args)
    {
        Scanner input = new Scanner(System.in);
        int n = input.nextInt();
        System.out.print("Enter list:");
        int[] list1 = new int[n];
        for (int i = 0; i < list1.length; i++)
        {
             list1[i] = input.nextInt();
        }
        partition(list1);
    }
     public static void partition(int[] list)
    {
        int partition = list[0];
        int partitionLocation = 0;
        int hi = list.length-1;
        while(partitionLocation < hi)
        {
          if(partition>list[partitionLocation+1])
          {
            list[partitionLocation] = list[partitionLocation+1];
            list[partitionLocation+1] = partition;
            partitionLocation++;
          }
          else
          {
            int temp = list[hi];
            list[hi]=list[partitionLocation+1];
            list[partitionLocation+1] = temp;
            hi- -;
          }
       }
        System.out.print("After the partition, the list is ");
        for (int i = 0; i < list.length; i++)
        {
            System.out.print(list[i] + " ");
        }
  }
}
```

Programming Assignment-VIII
(Multidimensional Arrays)

1. Write a java program to print M-by-N array in the tabular format.

```
import java.util.Scanner;
public class A8Q1
{
    public static void main(String[] args)
    {
        Scanner input = new Scanner(System.in);
        int M = input.nextInt();
        int N = input.nextInt();
        int a[][] = new int[M][N];

        for (int i = 0; i < M; i++)
            for (int j = 0; j < N; j++)
                a[i][j] = input.nextInt();

        for (int i = 0; i < M; i++)
        {
            for (int j = 0; j < N; j++)
            {
            System.out.print( a[i][j] + "   ");
            }
        System.out.println ();
        }
    }
}
```

2. Write a java program that reads a 3-by-4 matrix and displays the sum of each column & row.

```
import java.util.Scanner;
public class A8Q2
{
    public static void main(String[] args)
    {
        Scanner input = new Scanner(System.in);
        int ROW = 3, COL = 4;
        int m[][] = new int[ROW][COL];

        for (int i = 0; i < ROW; i++)
            for (int j = 0; j < COL; j++)
                m[i][j] = input.nextInt();

        for (int i = 0; i < COL; i++)
        {
            int sum = 0;
            for (int j = 0; j < ROW; j++)
                sum += m[j][i];
        }
        System.out.println (“column sum = ”+sum);
        for (int i = 0; i < ROW; i++)
        {
            int sum = 0;
            for (int j = 0; j < COL; j++)
                sum += m[i][j];
        }
        System.out.println (“row sum = ”+sum);
    }
}
```

3. Write a java program that reads a 4-by-4 matrix and displays the sum of all its elements on the diagonals.

```
import java.util.Scanner;
public class A8Q3
{
    public static void main(String[] args)
    {
        Scanner input = new Scanner(System.in);
        int ROW = 4, COL = 4, sum = 0;
        int m[][] = new int[ROW][COL];
        for (int i = 0; i < ROW; i++)
            for (int j = 0; j < COL; j++)
                m[i][j] = input.nextInt();
        System.out.println ();
        for (int i = 0; i < ROW; i++)
        {
            for (int j= 0; j < COL; j++)
            {
                if( i ==j)
                    sum += m[i][j];
            }
        }
        System.out.println (“sum of 1st diagonal is” + sum);
        System.out.println ();
        for (int i = 0; i < ROW; i++)
        {
            for (int j= 0; j < COL; j++)
            {
                if( (i +j) = = (ROW-1))
                    sum += m[i][j];
            }
        }
        System.out.println (“sum of 2nd diagonal is” + sum);
    }
}
```

4. Write a java program to create a two-dimensional array b[][] that is a copy of an existing two-dimensional array a[][], under each of the following assumptions:

a. a[][] is square

b. a[][] is rectangular

c. a[][] may be ragged

```
import java.util.*;
public class A8Q4
{
    public static void main(String[] args)
    {
        Scanner in=new Scanner(System.in);
        System.out.println (“Enter row and column”);
        int m=in.nextInt();
        int n=in.nextInt();
        int a[][]=new int[m][n];
        int b[][]=new int[m][n];
        for(int i=0;i<m;i++)
        {
            for(int j=0;j<n;j++)
            {
                a[i][j]=in.nextInt();
            }
        }
        if(m==n)
        {
            System.out.println("square matrix is");
            for(int i=0;i<m;i++)
            {
                for(int j=0;j<n;j++)
                {
                    b[i][j]=a[i][j];
                    System.out.print(b[i][j]+ “   ”);
                }
                System.out.println();
            }
        }
        if(m!=n)
        {
            System.out.println("rectangle matrix");
            for(int i=0;i<m;i++)
            {
                for(int j=0;j<n;j++)
                {
                    b[i][j]=a[i][j];
                    System.out.print(b[i][j]+ “   ”);
                }
                System.out.println();
            }
        }
        int c[][] = {{1,2,3,4},{5,6,7},{8,9},{10}};
        int p=c.length;
        int d[][]=new int[p][];
        for (int i = 0; i < p; i++)
        {
            d[i] = new int[p];
```

```
                for (int j = 0; j < c[i].length; j++)
                {
                        d[i][j] = c[i][j];
                        System.out.print( d[i][j] +" ");
                }
                System.out.println ();
            }
        }
}
```

5. Suppose a teacher with M students and N Marks of each student is maintained in an (M+1)-by (N+1) array, reserving the last column for each student's average mark and the last row for average test mark. Write a java program to compute the average mark for each student (average values of each row) and calculate the average test mark (average values of each column).

```
Import java.util.Scanner;
public class A8Q5
{
        public static void main(String[] args)
        {
                Scanner input = new Scanner(System.in);
                int M = input.nextInt();
                int N = input.nextInt();
                int a[][] = new int[M+1][N+1];

                for (int i = 0; i < M; i++)
                        for (int j = 0; j < N; j++)
                                a[i][j] = input.nextInt();

                //Compute row averages
                for (int i = 0; i < M; i++)
                {
                        int sum =0;
                        for (int j = 0; j < N; j++)
                                sum += a[i][j];
                        a[i][N] = (int) Math.round(sum/N);
                        System.out.println (a[i][N]);
                }

                //Compute column averages
                for (int j = 0; j < N; j++)
                {
                        int sum =0;
                        for (int i = 0; i < M; i++)
                                sum += a[i][j];
                        a[M][j] = (int) Math.round(sum/M);
                        System.out.println (a[M][j]);
                }
        }
}
```

6. Write a java program that prompts the user to enter two 3 * 3 matrices and displays their sum.

```
import java.util.Scanner;
public class A8Q6
{
	public static void main(String[] args)
	{
		Scanner input = new Scanner(System.in);
		int a[][] = new int[3][3];
		int b[][] = new int[3][3];
		int c[][] = new int[3][3];

		for (int i = 0; i < M; i++)
			for (int j = 0; j < N; j++)
				a[i][j] = input.nextInt();

		for (int i = 0; i < M; i++)
			for (int j = 0; j < N; j++)
				b[i][j] = input.nextInt();

		for (int i = 0; i < 3; i++)
		{
			for (int j = 0; j < 3; j++)
			{
				c[i][j] = a[i][j]+b[i][j];
			}
		}
		for (int i = 0; i < 3; i++)
		{
			for (int j = 0; j < 3; j++)
			{
				System.out.print (c[i][j] + "  ");
			}
			System.out.println();
		}
	}
}
```

7. Write a java program to transpose a square two-dimensional array in place without creating a second array.

```
import java.util.*;
public class A8Q7
{
	public static void main(String args[])
	{
		Scanner s=new Scanner(System.in);
		System.out.println("\nGive the no. of rows and column needed:\n");
		int row=s.nextInt();
		int col=s.nextInt();
		int a[][]=new int[row][col];
```

```
        System.out.print("enter a no:");
        for(int i = 0; i < row; i++)
        {
            for (int j = 0; j < col; j++)
            {
              a[i][j]=s.nextInt();
             }
         }
        System.out.print("Original matrix");
        for(int i = 0; i < row; i++)
        {
            for (int j = 0; j < col; j++)
            {
              System.out.print(a[i][j]+"\t");
            }
            System.out.println();
        }
        System.out.print("Transpose is");
        for(int i = 0; i < col; i++)
        {
            for (int j = 0; j < row; j++)
            {
              System.out.print(a[j][i]+"\t");
            }
            System.out.println();
        }
    }
}
```

8. Write a java program that takes an integer N from the command line and creates an N-by-N boolean array a[][] such that a[i][j] is true if i and j are relatively prime (have no common factors), and false otherwise.

```
import java.util.*;
public class A8Q8
{
    public static void main(String args[])
    {
        Scanner s=new Scanner(System.in);
        System.out.println("Enter N");
        int N = s.nextInt();
        boolean a[][] = new boolean[N+1][N+1];
        for(int i = 1; i <=N; i++)
        {
            for(int j = 1; j <=N; j++)
            {
                int x = i > j ? i:j;
                int y = i > j ? j:i;
                int val = gcd(x, y);
                if (val ==1)
                    a[i][j] = true;
                else
```

```
                  a[i][j] = false;
                System.out.print(a[i][j] + " ");
            }
            System.out.println();
        }
    }
    public static int gcd(int a, int b)
    {
        int gcd1=0;
        for (int i=1; i<= a && i <= b;i++)
           if (a%i==0 && b%i ==0)
              gcd1=i;
        return gcd1;
    }
}
```

9. Write a java program that prompts the user to enter two 3 * 3 matrices and displays their product.

```
Import java.util.Scanner;
public class A8Q9
{
    public static void main(String[] args)
    {
        Scanner input = new Scanner(System.in);
        int a[][] = new int[3][3];
        int b[][] = new int[3][3];
        int c[][] = new int[3][3];
        for (int i = 0; i < 3; i++)
            for (int j = 0; j < 3; j++)
                a[i][j] = input.nextInt();
        for (int i = 0; i < 3; i++)
            for (int j = 0; j < 3; j++)
                b[i][j] = input.nextInt();

        for (int i = 0; i < 3; i++)
        {
            for (int j = 0; j < 3; j++)
            {
                int c[i][j]=0;
                for (int k = 0; k < 3; k++)
                    c[i][j] += a[i][k]*b[k][j];
            }
        }
        System.out.print("Multiplication is ");
        for(int i = 0; i < 3; i++)
        {
            for (int j = 0; j < 3; j++)
            {
              System.out.print(c[i][j]+"\t");
            }
            System.out.print("\n");
        }
    }
}
```

10. Write a java program that randomly fills in 0s and 1s into a 4-by-4 matrix, prints the matrix, and finds the largest row and column indices with the most 1s.

```
public class A8Q10
{
    public static void main(String[] args)
    {
        int matrix[][] = new int[4][4];
        for (int i = 0; i < matrix.length; i++)
        {
            for (int j = 0; j < matrix[i].length; j++)
            {
                matrix[i][j] = (int)(Math.random() * 2);
            }
        }
        // Displays the matrix
        for (int i = 0; i < matrix.length; i++)
        {
            for (int j = 0; j < matrix[i].length; j++)
            {
                System.out.print(matrix[i][j]);
            }
            System.out.println();
        }
        // Find and display the largest row and column indices with the most 1s
        System.out.println("The largest row index: " + largestRow(matrix));
        System.out.println("The largest row column: " + largestColumn(matrix));
    }
    public static int largestRow(int m[][])
    {
        int maxRowIndex = 0;
        int max = 0;
        for (int i = 0; i < m.length; i++)
        {
            int count = 0;
            for (int j = 0; j < m[i].length; j++)
            {
                if (m[i][j] == 1)
                    count++;
            }
            if (count > max)
            {
                max = count;
                maxRowIndex = i;
            }
        }
        return maxRowIndex;
    }
    public static int largestColumn(int m[][])
    {
        int maxColumnIndex = 0;
        int max = 0;
        for (int col = 0; col < m[0].length; col++)
        {
            int count = 0;
            for (int row = 0; row < m.length; row++)
            {
                if (m[row][col] == 1)
                    count++;
            }
            if (count > max)
            {
                max = count;
                maxColumnIndex = col;
            }
        }
        return maxColumnIndex;
    }
}
```

Assignment-IX
(Classes and Objects)

1. Design a class Student with instance variables name, roll, mark and instance methods setData(), display(). Write a Java program to create three objects of Student class to input details of three different students and display the details. Enclose main() method inside another class StudentDetails. (Use the setter method setData() to input details.)

```
class Student
{
      String name;
      int roll, mark;

      void setData()
      {
            Scanner sc=new Scanner (System.in);
            name=sc.nextLine();
            roll=sc.nextInt();
            mark=sc.nextInt();
      }

      void display()
      {
            System.out.println(name+“    ”+ roll + “   ”+ mark);
      }
}
public class StudentDetails
{
      public static void main(String[] args)
      {
            Student s1 = new Student();
            Student s2 = new Student();
            Student s3 = new Student();

            System.out.println("Enter 1st student details: name, roll, mark:");
            s1.setData();
            System.out.println("Enter 2nd student details: name, roll, mark:");
            s2.setData();
            System.out.println("Enter 3rd student details: name, roll, mark:");
            s3.setData();

            System.out.println("The student details are \n\n");
            s1.display();
            s2.display();
            s3.display();
      }
}
```

2. Design a class named Rectangle to represent a rectangle. The class contains:
Two double data fields named width and height that specify the width and height of the rectangle. The default values are 1 for both width and height.
A no-argument constructor that creates a default rectangle.

A constructor that creates a rectangle with the specified width and height.
A method named getArea() that returns the area of this rectangle.
A method named getPerimeter() that returns the perimeter.
Write a java program that creates two Rectangle objects—one with width 4 and height 40 and the other with width 3.5 and height 35.9. Display the width, height, area, and perimeter of each rectangle in this order.

```
class Rectangle
{
    double width;
    double height;

    Rectangle()
    {
        width = 1;
        height = 1;
    }

    Rectangle(double newWidth, double newHeight)
    {
        width = newWidth;
        height = newHeight;
    }

    double getArea()
    {
        return width * height;
    }

    double getPerimeter()
    {
        return 2 * (width + height);
    }
}
public class A9Q2
{
    public static void main(String[] args)
    {
        Rectangle rectangle1 = new Rectangle(4, 40);
        Rectangle rectangle2 = new Rectangle(3.5, 35.9);

        System.out.println("\n Rectangle 1");
        System.out.println("-------------");
        System.out.println("Width:     " + rectangle1.width);
        System.out.println("Height:    " + rectangle1.height);
        System.out.println("Area:      " + rectangle1.getArea());
        System.out.println("Perimeter: " + rectangle1.getPerimeter());

        System.out.println("\n Rectangle 2");
        System.out.println("-------------");
        System.out.println("Width:     " + rectangle2.width);
        System.out.println("Height:    " + rectangle2.height);
        System.out.println("Area:      " + rectangle2.getArea());
        System.out.println("Perimeter: " + rectangle2.getPerimeter());
    }
}
```

3. Design a class named QuadraticEquation for a quadratic equation ax 2 + bx + c = 0. The class contains:
Private data fields a, b, and c that represent three coefficients.
A constructor for the arguments for a, b, and c.
Three getter methods for a, b, and c.
A method named getDiscriminant() that returns the discriminant. The methods named getRoot1() and getRoot2() for returning two roots of the equation. These methods are useful only if the discriminant is non-negative. Let these methods return 0 if the discriminant is negative. Write a java program that prompts the user to enter values for a, b, and c and displays the result based on the discriminant. If the discriminant is positive, display the two roots. If the discriminant is 0, display the one root. Otherwise, display “The equation has no roots.”

```
import java.util.Scanner;
class QuadraticEquation
{
    private double a, b, c;
    QuadraticEquation(double a, double b, double c)
    {
        this.a = a;
        this.b = b;
        this.c = c;
    }
    public double getA()
    {
        return a;
    }
    public double getB()
    {
        return b;
    }
    public double getC()
    {
        return c;
    }
    public double getDiscriminant()
    {
        return Math.pow(b, 2) - 4 * a * c;
    }
    public double getRoot1()
    {
        return getDiscriminant() < 0 ? 0 :
            ((-b) + Math.sqrt(Math.pow(b, 2) - 4 * a * c)) / (2 * a);
    }
    public double getRoot2()
    {
        return getDiscriminant() < 0 ? 0 :
            ((-b) - Math.sqrt(Math.pow(b, 2) - 4 * a * c)) / (2 * a);
    }
}
public class A9Q3
{
    public static void main(String[] args)
    {
        Scanner input = new Scanner(System.in);
        System.out.print("Enter a, b, c: ");
        double a = input.nextDouble();
        double b = input.nextDouble();
        double c = input.nextDouble();

        QuadraticEquation quadraticEquation = new QuadraticEquation(a, b, c);
        System.out.print("The equation has ");
        if (quadraticEquation.getDiscriminant() < 0)
            System.out.println("no real roots");
        else if (quadraticEquation.getDiscriminant() > 0)
        {
            System.out.println("two roots " + quadraticEquation.getRoot1() +
                " and " + quadraticEquation.getRoot2());
        }
        else
        {
            System.out.println("one root " + (quadraticEquation.getRoot1() > 0 ?
                quadraticEquation.getRoot1() : quadraticEquation.getRoot2()));
        }
    }
}
```

4. Design a class named Account that contains:

A private int data field named id for the account (default 0).

A private double data field named balance for the account (default 0).

A private double data field named annualInterestRate that stores the current interest rate (default 0). Assume all accounts have the same interest rate.

A private data field named dateCreated that stores the date when the account was created.

A no-argument constructor that creates a default account.

A constructor that creates an account with the specified id and initial balance.

The accessor and mutator methods for id, balance, and annualInterestRate.

The accessor method for dateCreated.

A method named getMonthlyInterestRate() that returns the monthly interest rate.

A method named getMonthlyInterest() that returns the monthly interest.

A method named withdraw that withdraws a specified amount from the account.

A method named deposit that deposits a specified amount to the account.

(Hint: The method getMonthlyInterest() is to return monthly interest, not the interest rate. Monthly interest is balance * monthlyInterestRate. monthlyInterestRate is annualInterestRate / 12. Note that annualInterestRate is a percentage, e.g., like 4.5%. You need to divide it by 100.)

Write a java program that creates an Account object with an account ID of 1122, a balance of Rs. 20,000, and an annual interest rate of 4.5%. Use the withdraw method to withdraw Rs. 2,500, use the deposit method to deposit Rs. 3,000, and print the balance, the monthly interest, and the date when this account was created.

```
class Account
{
    private int id;
    private double balance;
    private static double annualInterestRate;
    private Date dateCreated;

    Account()
    {
        id = 0;
        balance = 0;
        annualInterestRate = 0;
        dateCreated = new Date();
    }

    Account(int newId, double newBalance)
    {
        id = newId;
        balance = newBalance;
        dateCreated = new Date();
    }

    public void setId(int newId)
    {
        id = newId;
    }

    public void setBalance(double newBalance)
    {
        balance = newBalance;
    }

    public void setAnnualInterestRate(double newAnnualInterestRate)
    {
        annualInterestRate = newAnnualInterestRate;
    }

    public int getId()
    {
        return id;
    }

    public double getBalance()
    {
        return balance;
    }


    public double getAnnualInterestRate()
    {
        return annualInterestRate;
    }

    public String getDateCreated()
    {
        return dateCreated.toString();
    }

    public double getMonthlyInterestRate()
    {
        return annualInterestRate / 12;
    }

    public double getMonthlyInterest()
    {
        return balance * (getMonthlyInterestRate() / 100);
    }

    public void withdraw(double amount)
    {
        balance -= amount;
    }

    public void deposit(double amount)
    {
        balance += amount;
    }
}
public class A9Q4
{
    public static void main(String[] args)
    {
        Account account = new Account(1122, 20000);
        account.setAnnualInterestRate(4.5);
        account.withdraw(2500);
        account.deposit(3000);
        System.out.println("\n          Account Statement");
        System.out.println("------------------------------------------");
        System.out.println("Account ID: " + account.getId());
        System.out.println("Date created: " + account.getDateCreated());
        System.out.printf("Balance: $%.2f\n", account.getBalance());
        System.out.printf("Monthly interest: $%.2f\n", account.getMonthlyInterest());
    }
}
```

5. Design a class named Fan to represent a fan. The class contains:
Three constants named SLOW, MEDIUM, and FAST with the values 1, 2, and 3 to denote the fan speed.
A private int data field named speed that specifies the speed of the fan (the default is SLOW).
A private boolean data field named on that specifies whether the fan is on (the default is false).
A private double data field named radius that specifies the radius of the fan (the default is 5).
A string data field named color that specifies the color of the fan (the default is blue).

The accessor and mutator methods for all four data fields.
A no-argument constructor that creates a default fan.
A method named toString() that returns a string description for the fan. If the fan is on, the method returns the fan speed, color, and radius in one combined string. If the fan is not on, the method returns the fan color and radius along with the string "fan is off" in one combined string.
Write a java program that creates two Fan objects. Assign maximum speed, radius 10, color yellow, and turn it on to the first object. Assign medium speed, radius 5, color blue, and turn it off to the second object. Display the objects by invoking their toString method.

```
class Fan
{
    final static int SLOW = 1;
    final static int MEDIUM = 2;
    final static int FAST = 3;
    private int speed;
    private boolean on;
    private double radius;
    String color;

    Fan()
    {
        speed = SLOW;
        on = false;
        radius = 5;
        color = "blue";
    }

    public void setSpeed(int newSpeed)
    {
        speed = newSpeed;
    }

    public void turnOn()
    {
            on = true;
    }

    public void turnOff()
    {
            on = false;
    }

    public void setColor(String newColor)
    {
        color = newColor;
    }

    public void setRadius(double newRadius)
    {
        radius = newRadius;
    }
    public String getSpeed()
    {
        String s = "";
        switch (speed)
        {
            case SLOW: s = "SLOW"; break;
            case MEDIUM: s = "MEDIUM"; break;
            case FAST: s = "FAST";
        }
        return s;
    }

    public boolean isOn()
    {
        return on;
    }

    public double getRadius()
    {
        return radius;
    }

    public String getColor()
    {
        return color;
    }

    public String toString()
    {
        if (on == true)
        {
            return "\nFan speed: " + getSpeed() + ", color: " + color + ", radius: "
                                                                   + radius + "\n";
        }
        else
        {
            return "\nFan color: " + color + ", radius: " + radius +"\n fan is off\n";
        }
    }
}
public class A9Q5
{
    public static void main(String[] args)
    {
        final int SLOW = 1;             // Fan speed slow
        final int MEDIUM = 2;           // Fan speed medium
        final int FAST = 3;             // Fan speed fast

        Fan fan1 = new Fan();
        Fan fan2 = new Fan();
        fan1.setSpeed(FAST);
        fan1.setRadius(10);
        fan1.setColor("yellow");
        fan1.turnOn();

        fan2.setSpeed(MEDIUM);
        fan2.setRadius(5);
        fan2.setColor("blue");
        fan2.turnOff();

        System.out.println(fan1.toString());
        System.out.println(fan2.toString());
    }
}
```

6. Design a class named StopWatch. The class contains:
Private data fields startTime and endTime with getter methods.
A no-argument constructor that initializes startTime with the current time.
A method named start() that resets the startTime to the current time.
A method named stop() that sets the endTime to the current time.
A method named getElapsedTime() that returns the elapsed time for the stopwatch in milliseconds.
Write a java program that measures the execution time of sorting 100 numbers using bubble sort.

```
class StopWatch
{
    private long startTime, endTime;

    StopWatch()
    {
        startTime = System.currentTimeMillis();
    }

    public void start()
    {
        startTime = System.currentTimeMillis();
    }

    public void stop()
    {
        endTime = System.currentTimeMillis();
    }

    public long getElapsedTime()
    {
        return getEndTime() - getStratTime();
    }

    public long getStratTime()
    {
        return startTime;
    }

    public long getEndTime()
    {
        return endTime;
    }
}
public class A9Q6
{
    public static void main(String[] args)
    {
        StopWatch stopWatch = new StopWatch();

        int[] randomArray = getArray();

        stopWatch.start();

        bubbleSort(randomArray);

        stopWatch.stop();

        System.out.println("The execution time of sorting 100,000 " +
            "numbers using bubble sort: " + stopWatch.getElapsedTime() +
            " milliseconds");
    }
    public static int[] getArray()
    {
        int[] array = new int[100000];
        for (int i = 0; i < array.length; i++)
        {
            array[i] = (int)(Math.random() * 100000);
        }
        return array;
    }
    public static void bubbleSort(int[] array)
    {
        int n = array.length;
        int temp =0;
        for (int i = 0; i < n; i++)
        {
            for (int j = 1; j < n-i ; j++)
            {
                if (array[j-1] < array[j])
                {
                    temp = array[j-1];
                    array[j-1] = array[j];
                    array[j] = temp;
                }
            }
        }
        for (int i = 0; i < n; i++)
            System.out.print (array[i] +“   ”); } }
```