// By Ansuman Swain

```
                        CSE 2001: Data Structure & Algorithms
                             Programming Assignment-I
                                    (Java Primer)
Question-1:
Write a Java program that can take a positive integer greater than 2 as input and write out the number
of times one must repeatedly divide this number by 2 before getting a value less than 2.
CODE-
package assignment_1;
import java.util.*;
public class Q1
{

    public static void main(String[] args)
    {
        Scanner obj=new Scanner(System.in);
        System.out.print("Enter positive integer greater than 2 : ");
        int a=obj.nextInt();
        int count=0;
        while(a>2)
        {
            count++;
            a=a/2;
        }
        System.out.println("The number of times one must repeatedly divide this number"
                           +"by 2 before getting a value less than 2 is : "+count);
        obj.close();
    }

}
OUTPUT-:
Enter positive integer greater than 2 : 67
The number of times one must repeatedly divide this
numberby 2 before getting a value less than 2 is : 5
```

// By Ansuman Swain

```
Question-2:
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.
Metric: BMI = Weight (kg)(Height)^2 (m^2)
Write a java program by using conditional statement to show the category for a given BMI.

            Category            BMI
        Less than 18.5      Underweight
        18.5 to 24.9        Normal Weight
        25.0 to 29.9        Overweight
        30.0 or more        Obese
CODE-
package assignment_1;
import java.util.*;
public class Q2
{
    public static void main(String[] args)
    {
        Scanner obj=new Scanner (System.in);
        System.out.print("Enter the Weight in kg : ");
        double w=obj.nextDouble();
        System.out.print("Enter the Height in meters : ");
        double h=obj.nextDouble();
        double bmi=w/(Math.pow(h,2));
        String a="";

        if(bmi<18.5)
            a="Underweight";
        else if(bmi<24.9)
            a="Normal Weight";
        else if(bmi<29.9)
            a="Overweight";
        else
            a="Obese";
    System.out.println("The person is "+a);
    obj.close();
    }

}

OUTPUT-
Enter the Weight in kg : 96
Enter the Height in meters : 1.4
The person is Obese
```

// By Ansuman Swain

```
Question-3:
Write a java program to check whether a number is a spy number or not.
CODE-
package assignment_1;
import java.util.*;
public class Q3
{
        public static void main(String[] args)
    {
        Scanner obj=new Scanner(System.in);
        System.out.print("Enter a number : ");
        int n=obj.nextInt();
        int sum=0,product=1,t=n;
        while(n>0)
        {
            int r=n%10;
            sum+=r;
            product*=r;
            n/=10;
        }
        if(sum==product)
            System.out.print(t+" is a spy number");
        else
            System.out.print(t+" is not a spy number");
        obj.close();
    }

}

OUTPUT-
Enter a number : 1124
1124 is a spy number.
```

// By Ansuman Swain

Question-4:
Write a Java program that outputs all possible strings formed by using the characters 'c', 'a', 'r', 'b', 'o', and 'n' exactly once.

CODE-:

```java
package assignment_1;
public class Q4
{
    public static void main(String[] args)
    {
        char arr[]={'c','a','r','b','o','n'} ;
        for(int i=0;i<6;i++)
            for(int j=0;j<6;j++)
                for(int k=0;k<6;k++)
                    for(int l=0;l<6;l++)
                        for(int m=0;m<6;m++)
                            for(int n=0;n<6;n++)
                {
                    if(i!=j && i!=k && i!=l && i!=m && i!=n
                        && j!=k && j!=l && j!=m && j!=n
                        && k!=l && k!=m && k!=n
                        && l!=m && l!=n
                        && m!=n )
                    System.out.println(""+arr[i]+arr[j]+arr[k]+arr[l]+arr[m]+arr[n]);
                }
    }

}
```

// By Ansuman Swain

Question-5:
Write a java method to calculate the sum of digits of a given number until the number is a single digit.
The method signature is as follows.
public static int sum_Of_Digits(int n)

CODE-:

```
package assignment_1;
import java.util.*;
public class Q5
{
    public static void main(String[] args)
    {
        Scanner obj=new Scanner(System.in);
        System.out.print("Enter a number : ");
        int x=obj.nextInt();
        System.out.print(  "Sum of digits of "+x
                        + " until the number is a"
                        + " single digit is "+sum_Of_Digits(x));
        obj.close();
    }
    public static int sum_Of_Digits(int n)
    {
        int sum=0;
        while(n>0)
        {
            int r=n%10;
            sum+=r;
            n/=10;
        }
        if(sum>=10)
            return sum_Of_Digits(sum);
        else
            return sum;

    }
}
```

OUTPUT:-

```
Enter a number : 9294
Sum of digits of 9294 until the number is a single digit is 6
```

// By Ansuman Swain

Question-6:
Write a Java method, isOdd( ), that takes an int i and returns true if and only if i is odd. Your method can't use the multiplication, modulus, or division operators, The method signature is as follows.
public static boolean isOdd(int n)

CODE-:

```
package assignment_1;
import java.util.*;
public class Q6
{
    public static void main(String[] args)
    {
        Scanner obj=new Scanner(System.in);
        System.out.print("Enter a number : ");
        int x=obj.nextInt();
        System.out.println(x+" is odd : "+isOdd(x));
        obj.close();
    }
    public static boolean isOdd(int n)
    {
        int b=n&1;
        if(b==0)
            return false;
        else
            return true;

    }
}
```

OUTPUT:-

```
Enter a number : 37
37 is odd : true
```

// By Ansuman Swain

Question-7:
Write a java program to find the maximum and minimum and how many times they both occur in an array of n elements. Find out the positions where the maximum first occurs and the minimum last occurs.

CODE-:

```
package assignment_1;
import java.util.*;
public class Q7
{
    public static void main(String[] args)
    {
        Scanner obj=new Scanner(System.in);
        System.out.print("Enter number of elements of Array: ");
        int n=obj.nextInt();
        int arr[]=new int[n];
        System.out.print("Enter elements of the array: ");
        for (int i=0;i<n;i++)
            arr[i]=obj.nextInt();
        int[] mm=MaxMin(arr);
        int[] maxop=Maxocc(arr,mm[0]);
        int[] minop=Minocc(arr,mm[1]);
        System.out.println("Maximum element of Array is "+mm[0]+" and occurs "+maxop[0] +
                           (maxop[0]>1?" times": " time"));
        System.out.println("Minimum element of Array is "+mm[1]+" and occurs "+minop[0]+
                           (minop[0]>1?" times": " time"));
        System.out.println("First occurrence of maximum element is at position "+maxop[1]);
        System.out.println("Last occurrence of minimum element is at position "+minop[1]);
        obj.close();
    }
    public static int[] MaxMin(int []a)
    {
        int max=a[0];
        int min=a[0];
        for(int i=0;i<a.length;i++)
        {
         if(a[i]>max)
            max=a[i];
         if(a[i]<min)
            min=a[i];
        }
        int[] m={max,min};
        return m;
    }
    public static int[] Maxocc(int []a,int max)
    {
        int mc=0,index=0;
        for(int i=a.length-1;i>=0;i--)
            if(max==a[i])
            {
                mc++;
                index=i;
            }
        int[] aa={mc,index+1};
        return aa;
    }
    public static int[] Minocc(int []a,int min)
    {
        int mc=0,index=0;
        for(int i=0;i<a.length;i++)
            if(min==a[i])
            {
                mc++;
                index=i;
            }
        int[] aa={mc,index+1};
        return aa;
    }
}
```

OUTPUT:-

```
Enter number of elements of Array: 5
Enter elements of the array: 12 14 12 14 13
Maximum element of Array is 14 and occurs 2 times
Minimum element of Array is 12 and occurs 2 times
First occurrence of maximum element is at position 2
Last occurrence of minimum element is at position 3
```

// By Ansuman Swain

Question-8:
Write a java program to print M-by-N array in the tabular format. Also, display the sum of elements of the array.

CODE-:

```
package assignment_1;
import java.util.*;
public class Q8
{
    public static void main(String[] args)
    {
        Scanner obj=new Scanner(System.in);
        System.out.print("Enter number of Row and Columns of 2D-Array:");
        int m=obj.nextInt();
        int n=obj.nextInt();
        int arr[][]=new int[m][n];
        int sum=0;
        System.out.print("Enter elements of 2D-Array:");
        for (int i=0;i<m;i++)
            for(int j=0;j<n;j++)
            arr[i][j]=obj.nextInt();
        System.out.println("The elements of 2D array are: ");
        for (int i=0;i<m;i++)
        {
            for(int j=0;j<n;j++)
            {
                System.out.print(arr[i][j]+" ");
                sum+=arr[i][j];
            }
            System.out.println();
        }
        System.out.print("The sum of elements of the 2D-Array is "+sum);
        obj.close();
    }

}
```

OUTPUT:-

```
Enter number of Row and Columns of 2D-Array:3 3
Enter elements of 2D-Array:1 2 3 2 3 4 3 4 5
The elements of 2D array are:
1 2 3
2 3 4
3 4 5
```

// By Ansuman Swain

Question-9:
Write a method that sums all the numbers in the major diagonal in an n * n matrix of double values using the following header:
public static double sumMajorDiagonal(double[][] m)
Write a java program that reads a 4-by-4 matrix and displays the sum of all its elements on the major diagonal.

CODE-:

```
package assignment_1;
import java.util.*;
public class Q9
{
    public static void main(String[] args)
    {
        Scanner obj=new Scanner(System.in);
        double arr[][]=new double[4][4];
        System.out.println("Enter a 4-by-4 matrix row by row: ");
        for (int i=0;i<4;i++)
            for(int j=0;j<4;j++)
            arr[i][j]=obj.nextDouble();

        System.out.print("Sum of the elements in the major diagonal is "+sumMajorDiagonal(arr));
        obj.close();
    }
    public static double sumMajorDiagonal(double[][] m)
    {
        double sum=0;
        for (int i=0;i<4;i++)
            for(int j=0;j<4;j++)
                if (i==j)
                    sum+=m[i][j];

        return sum ;
    }
}
```

OUTPUT:-

```
Enter a 4-by-4 matrix row by row:
1 2 3 4.0
5 6.5 7 8
9 10 11 12
13 14 15 16
Sum of the elements in the major diagonal is 34.5
```

// By Ansuman Swain

Question-10:
Write a method that returns the sum of all the elements in a specified column in a matrix using the following header:
public static double sumColumn(double[][] m, int columnIndex)
Write a java program that reads a 3-by-4 matrix and displays the sum of each column.

CODE-:

```
package assignment_1;
import java.util.*;
public class Q10
{
    public static void main(String[] args)
    {
        Scanner obj=new Scanner(System.in);
        double arr[][]=new double[4][4];
        System.out.println("Enter a 3-by-4 matrix row by row: ");
        for (int i=0;i<3;i++)
            for(int j=0;j<4;j++)
            arr[i][j]=obj.nextDouble();
        for(int k=0;k<4;k++)
        System.out.println("Sum of the elements at column " +k+"  is  "+sumColumn(arr,k));
        obj.close();
    }
    public static double sumColumn(double[][] m, int columnIndex)
    {
        double sum=0;
        for (int i=0;i<4;i++)
            for(int j=0;j<4;j++)
                if (j==columnIndex)
                    sum+=m[i][j];

        return sum ;
    }
}
```

OUTPUT:-

```
Enter a 3-by-4 matrix row by row:
1.5 2 3 4
5.5 6 7 8
9.5 1 3 1
Sum of the elements at column 0  is  16.5
Sum of the elements at column 1  is  9.0
Sum of the elements at column 2  is  13.0
Sum of the elements at column 3  is  13.0
```

// By Ansuman Swain

Home Assignment

Question-1:
Write a Java program that takes as input three integers, a, b, and c, from the Java console and determines if they can be used in a correct arithmetic formula (in the given order), like “a + b = c,” “a = b − c,” or “a*b = c.”

CODE-:

```
package assignment_1;
import java.util.*;
public class HW1
{
    public static void main(String[] args) {
        Scanner obj = new Scanner(System.in);
        System.out.print("enter 1st integer : ");
        int a = obj.nextInt();
        System.out.print("enter 2nd integer : ");
        int b = obj.nextInt();
        System.out.print("enter 3rd integer : ");
        int c = obj.nextInt();
        if( a+b==c && b-c ==a && a*b==c )
        System.out.println("the numbers can be used in a correct arithmetic formula");
        else
        System.out.println("the numbers cannot be used in correct arithmetic formula ");
        obj.close();
    }
}
```

OUTPUT:-

```
enter 1st integer : 1
enter 2nd integer : 2
enter 3rd integer : 1
the numbers cannot be used in correct arithmetic formula
```

// By Ansuman Swain

Home Assignment

Question-2:
Write a Java program that takes all the lines input to standard input and writes them to standard output in reverse order. That is, each line is output in the correct order, but the ordering of the lines is reversed.

CODE-:

```
package assignment_1;
import java.util.*;
public class HW2
{
    public static void main(String[] args)
    {
        Scanner obj=new Scanner(System.in);
        System.out.print("Enter the number of lines for input : ");
        int n=obj.nextInt();
        obj.nextLine();
        String[] s=new String[n];
        System.out.println("Enter lines : ");
        for(int i=0;i<n;i++)
            s[i]=obj.nextLine();
        System.out.println("Lines in reverse ordered are : ");
        for(int j=n-1;j>=0;j--)
            System.out.println(s[j]);
        obj.close();
    }
}
```

OUTPUT:-

```
Enter the number of lines for input : 5
Enter lines :
        Ansuman Swain
        soa
        Iter
        BHubaneswar odisha
        Khandagiri
Lines in reverse ordered are :
        Khandagiri
        BHubaneswar odisha
        Iter
        soa
        Ansuman Swain
```

// By Ansuman Swain

Home Assignment

Question-3:
Write a Java program that takes two arrays a and b of length n storing int values, and returns the dot product of a and b. That is, it returns an array c of length n such that c[i] = a[i] · b[i], for i = 0, . . . , n − 1.

CODE-:

```
package assignment_1;
import java.util.*;
public class HW3
{
    public static void main(String[] args)
    {
        Scanner obj=new Scanner(System.in);
        System.out.print("Enter the length of the array : ");
        int n=obj.nextInt();
        int a[]=new int[n];
        int b[]=new int[n];
        int c[]=new int[n];
        System.out.print("Enter elements of array a : ");
        for (int i=0;i<n;i++)
            a[i]=obj.nextInt();
        System.out.print("Enter elements of array b : ");
        for (int j=0;j<n;j++)
            b[j]=obj.nextInt();
        System.out.println("Array c of having dot product is : ");
        for (int k=0;k<n;k++)
        {
            c[k]=a[k]*b[k];
            System.out.print(c[k]+" ");
        }
        obj.close();
    }
}
```

OUTPUT:-

```
Enter the length of the array : 5
Enter elements of array a : 1 2 3 4 5
Enter elements of array b : 7 6 5 2 3
Array c of having dot product is :
7 12 15 8 15
```

// By Ansuman Swain

Home Assignment

Question-4:
Write a method to add two matrices. The header of the method is as follows:
public static double[][] addMatrix(double[][] a, double[][] b)
In order to be added, the two matrices must have the same dimensions and the same or compatible types of elements.

CODE-:

```
package assignment_1;
import java.util.*;
public class HW4
{
    public static void main(String[] args)
    {
        Scanner obj=new Scanner(System.in);
        System.out.println("Enter length of row and coloumn : ");
        int m=obj.nextInt();
        int n=obj.nextInt();
        double x[][]=new double[m][n];
        double y[][]=new double[m][n];
        double z[][]=new double[m][n];
        System.out.println("Enter element of first element : ");
        for (int i=0;i<m;i++)
            for(int j=0;j<n;j++)
                x[i][j]=obj.nextDouble();
        System.out.println("Enter element of second element : ");
        for (int i=0;i<m;i++)
            for(int j=0;j<n;j++)
                y[i][j]=obj.nextDouble();
        System.out.println("The addition of two array is : ");
        z=addMatrix(x,y);
        for (int i=0;i<m;i++)
        {
            for(int j=0;j<n;j++)
              System.out.print(z[i][j]+" ");
            System.out.println();
        }
        obj.close();
    }
    public static double[][] addMatrix(double[][] a, double[][] b)
    {
        int row=a.length;
        int colomn=a[0].length;
        double sum[][]=new double[row][colomn];
        for (int i=0;i<row;i++)
             for(int j=0;j<colomn;j++)
               sum[i][j]=a[i][j]+b[i][j];

        return sum;
    }
}
```

OUTPUT:-

```
Enter length of row and coloumn :
3 3
Enter element of first element :
1 2 3
3 4 5
6 7 9
Enter element of second element :
6 5 4
3 2 1
9 8 7
The addition of two array is :
7.0 7.0 7.0
6.0 6.0 6.0
15.0 15.0 16.0
```

// By Ansuman Swain

Home Assignment

Question-5:
Write a java program that randomly fills in 0s and 1s into a 4-by-4 matrix, prints the matrix, and finds the first row and column with the most 1s.

CODE-:

```
package assignment_1;
public class HW5
{
    public static void main(String[] args)
    {
        int[][] arr = new int[4][4];
        for (int i = 0; i < 4; i++)
        {
            for (int j = 0; j < 4; j++)
                arr[i][j] = (int)(Math.random( ) * 2);
        }
        for (int i = 0; i < arr.length; i++)
        {
            for (int j = 0; j < arr[i].length; j++)
                System.out.print(arr[i][j]+" ");

            System.out.println( );
        }
        System.out.println("The largest row index : " + largestRow(arr));
        System.out.println("The largest column index : " + largestColumn(arr));

    }
    public static int largestRow(int[][] m) {
        int Maximum = 0;
        int temp = 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 > temp) {
                temp = count;
                Maximum = i;
            }
        }
        return Maximum;
    }
    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;
    }
}
```

OUTPUT:-

```
0 1 1 1
0 1 1 0
0 0 0 1
1 0 0 1
The largest row index : 0
The largest column index : 3
```