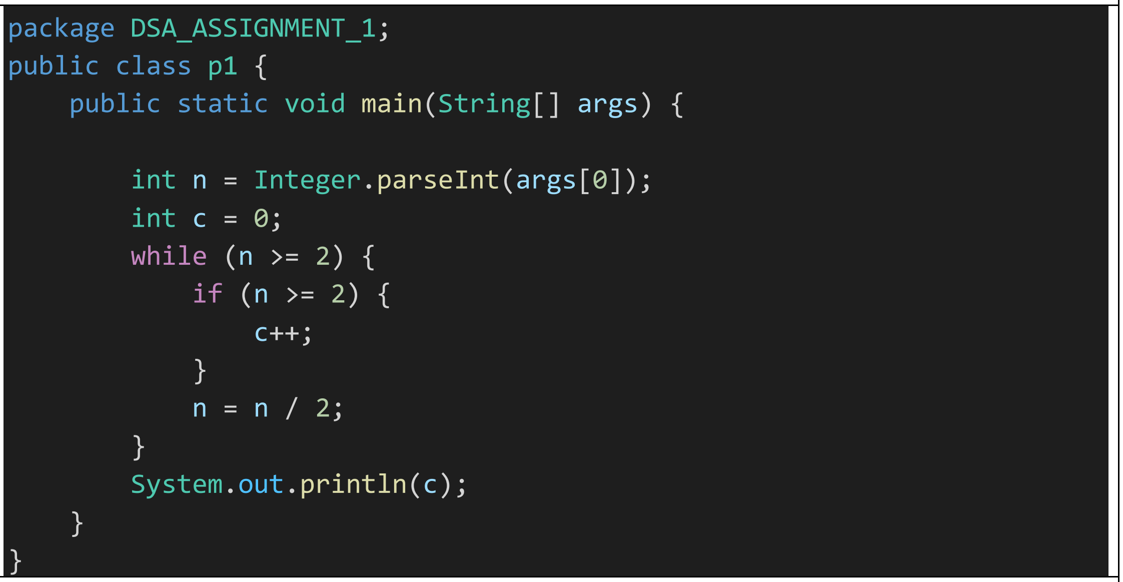
1.

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.



```java
package DSA_ASSIGNMENT_1;
public class p1 {
    public static void main(String[] args) {

        int n = Integer.parseInt(args[0]);
        int c = 0;
        while (n >= 2) {
            if (n >= 2) {
                c++;
            }
            n = n / 2;
        }
        System.out.println(c);
    }
}
```

Output.

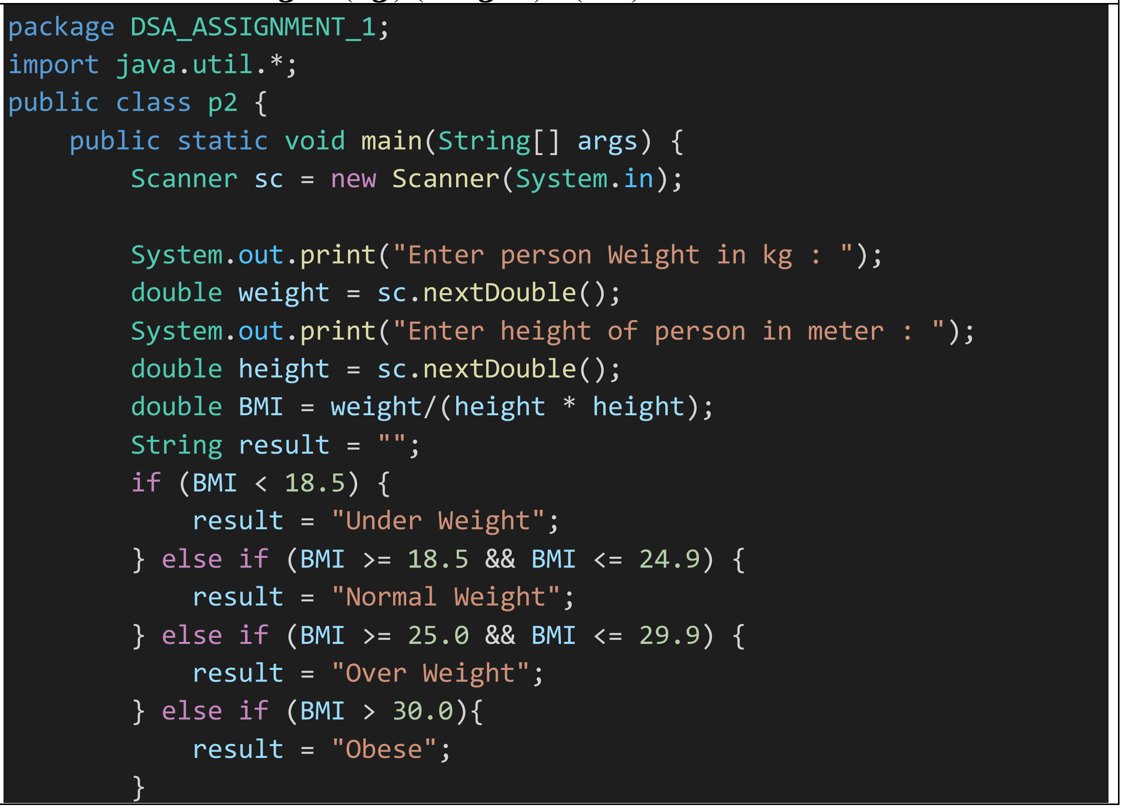
The number of times one must repeatedly divide this number by 2 before getting a value less than 2 is 6

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 = $\boldsymbol{Weight\ (kg)/(Height)2\ (m2)}$

Code.



```java
package DSA_ASSIGNMENT_1;
import java.util.*;
public class p2 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print("Enter person Weight in kg : ");
        double weight = sc.nextDouble();
        System.out.print("Enter height of person in meter : ");
        double height = sc.nextDouble();
        double BMI = weight/(height * height);
        String result = "";
        if (BMI < 18.5) {
            result = "Under Weight";
        } else if (BMI >= 18.5 && BMI <= 24.9) {
            result = "Normal Weight";
        } else if (BMI >= 25.0 && BMI <= 29.9) {
            result = "Over Weight";
        } else if (BMI > 30.0){
            result = "Obese";
        }
        System.out.println("The person is " + result);
        sc.close();
    }
}
```

Output.

Enter person Weight in kg : 96
Enter height of person in meter : 1.4
The person is Obese

3. Write a java program to check whether a number is a spy number or not.

Code.

```java
package DSA_ASSIGNMENT_1;
import java.util.*;
public class p3 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter a Number : ");
        int n = sc.nextInt();
        int x = n;
        int sum = 0;
        int pro = 1;
        while (n != 0) {
            int r = n % 10;
            sum += r;
            pro *= r;
            n /= 10;
        }
        if (sum == pro) {
            System.out.println(x + " is a spy Number");
        } else {
            System.out.println(x + " is not a spy Number");
        }
        sc.close();
    }
}
```

Output.

Enter a Number : 1124
1124 is a spy Number

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.

Output.

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.

```java
package DSA_ASSIGNMENT_1;
import java.util.Scanner;
public class p5 {
    public static int sum_of_Digits(int n) {
        int sum = 0;
        while (n != 0) {
            int r = n % 10;
            sum += r;
            n /= 10;
        }
        return sum;
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter a Number : ");
        int n = sc.nextInt();
        int sum = sum_of_Digits(n);

        while (sum >= 9) {
            sum = sum_of_Digits(sum);
        }
        System.out.println("sum od digit of " + n + " until the number is a single digit is " + sum);
        sc.close();
    }
}
```

Output.

```
Enter a Number : 9294
sum od digit of 9294 until the number is a single digit is 6
```

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 DSA_ASSIGNMENT_1;
import java.util.*;
public class p6 {
    public static boolean isOdd(int n) {
        return ((n & 1) == 1);
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter a Number : ");
        int n = sc.nextInt();
        System.out.println(n + " is odd : " + isOdd(n));
        sc.close();
    }
}
```

Output.

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

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.

```java
package DSA_ASSIGNMENT_1;
import java.util.*;
public class p7 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter number of Array : ");
        int size = sc.nextInt();

        int arr[] = new int[size];
        System.out.print("Enter the elements of the arrays : ");
        for (int i=0 ; i<size ; i++) {
            arr[i] = sc.nextInt();
        }
        int max = Integer.MIN_VALUE;
        int min = Integer.MAX_VALUE;
        int max_occur = 0;
        int min_occur = 0;
        int first_max_postion = 0;
        int last_min_postion = 0;

        for (int i=0 ; i<arr.length ; i++) {
            if (arr[i] > max) {
                max = arr[i];
                max_occur = 1;
                first_max_postion = i;
            } else  if (arr[i] == max) {
                max_occur++;
            }
            if (arr[i] < min) {
                min = arr[i];
                min_occur = 1;
                last_min_postion = i;
            } else if (arr[i] == min) {
                min_occur++;
                last_min_postion = i;
            }
        }
        System.out.println("Maximum element of Array is " + max + " and occurs " + max_occur + " times");
        System.out.println("Minimum element of Array is " + min + " and occurs " + min_occur + " times");
        System.out.println("First occurrence of maximum element is at 
position " + (first_max_postion + 1));
        System.out.println("Last occurrence of minimum element is at 
position " + (last_min_postion + 1));
        sc.close();
    }
}
```

Output.

```
Enter number of Array : 5
Enter the elements of the arrays : 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
```

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 DSA_ASSIGNMENT_1;
import java.util.*;
public class p8 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter number of Row and Columns of 2D-Array : 
");
        int r = sc.nextInt();
        int c = sc.nextInt();
        System.out.print("Enter element of 2D-Array : ");
        int arr[][] = new int[r][c];
        for (int i=0 ; i<r ; i++) {
            for (int j=0 ; j<c ; j++) {
                arr[i][j] = sc.nextInt();
            }
        }
        for (int i=0 ; i<r ; i++) {
            for (int j=0 ;j<c ; j++) {
                System.out.print(arr[i][j] + " ");
            }
            System.out.println();
        }
        int sum = 0;
        for (int i=0 ; i<r ; i++) {
            for (int j=0 ;j<c ; j++) {
                sum += arr[i][j];
            }
        }
        System.out.println("The sum of element of the 2D-Array is " + 
sum);
```

```
        sc.close();
    }
}
```

Output.

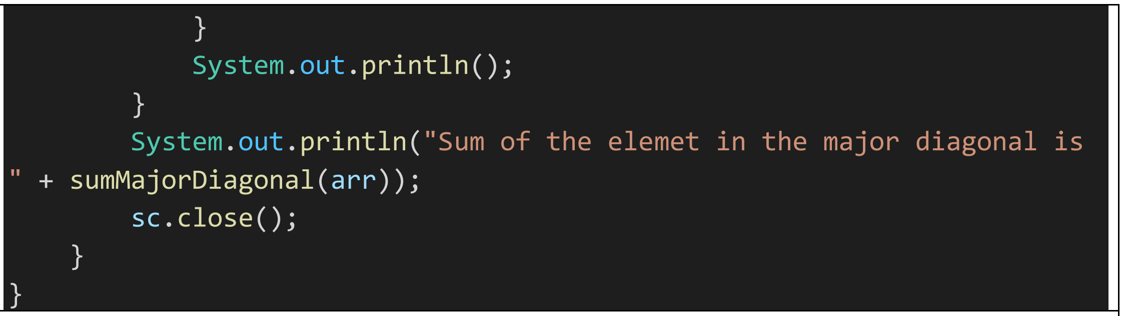
```
Enter number of Row and Columns of 2D-Array : 3 3
Enter element of 2D-Array : 1 2 3 2 3 4 3 4 5
1 2 3
2 3 4
3 4 5
The sum of element of the 2D-Array is 27
```

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 DSA_ASSIGNMENT_1;
import java.util.*;
public class p9 {
    public static double sumMajorDiagonal(double [][]m) {
        int r = m.length;
        int c = m[0].length;
        double sum = 0;
        for (int i=0 ; i<r ; i++) {
            for (int j=0 ; j<c ; j++) {
                if (i == j) {
                    sum += m[i][j];
                }
            }
        }
        return sum;
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the Row and Column of 2D-Array : ");
        int r = sc.nextInt();
        int c = sc.nextInt();
        double arr[][] = new double[r][c];
        System.out.print("Enter the matric row by row : ");
        for (int i=0 ; i<r ; i++) {
            for (int j=0 ; j<c ; j++) {
                arr[i][j] = sc.nextDouble();
            }
        }
        for (int i=0 ; i<r ; i++) {
            for (int j=0 ; j<c ; j++) {
                System.out.print(arr[i][j] + " ");
```



```
            }
            System.out.println();
        }
        System.out.println("Sum of the elemet in the major diagonal is 
" + sumMajorDiagonal(arr));
        sc.close();
    }
}
```

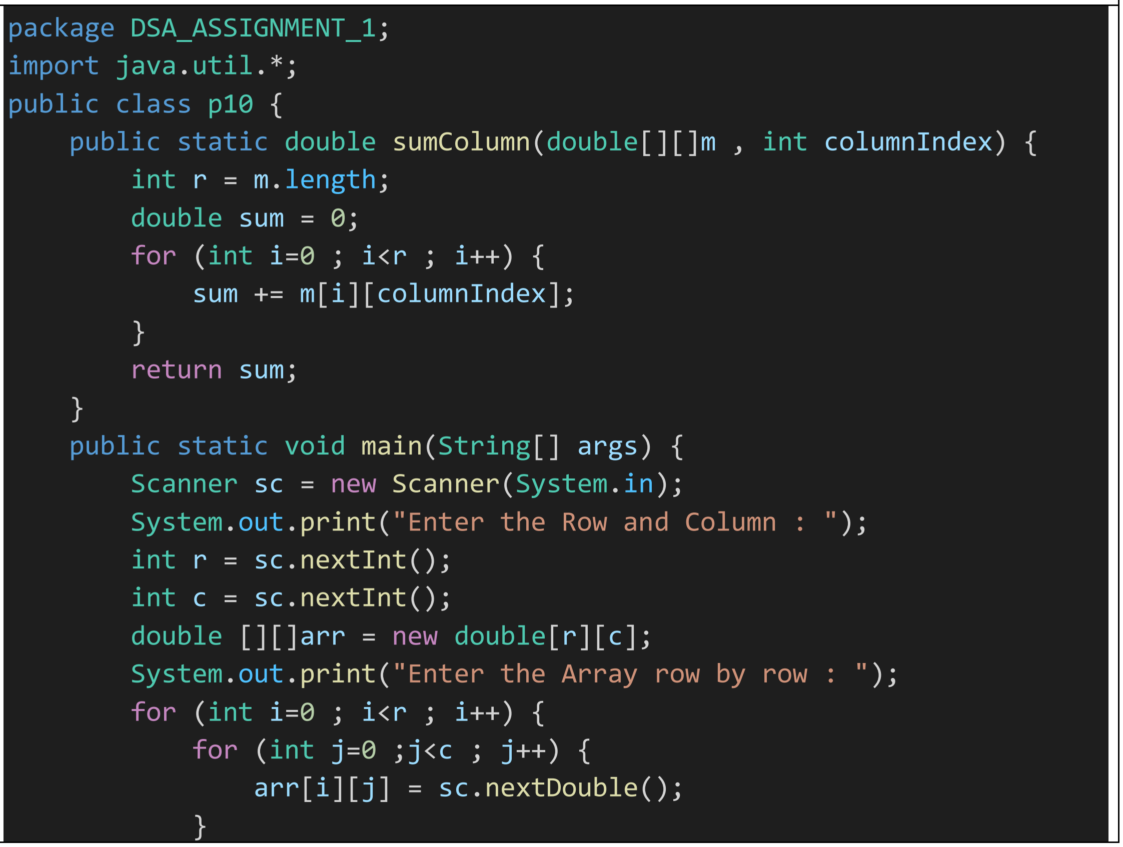
Output.

```
Enter the Row and Column of 2D-Array : 4 4
Enter the matric row by row : 1 2 3 4.0 5 6.5 7 8 9 10
11 12 13 14 15 16

1.0  2.0  3.0  4.0
5.0  6.5  7.0  8.0
9.0 10.0 11.0 12.0
13.0 14.0 15.0 16.0

Sum of the elemet in the major diagonal is 34.5
```

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 DSA_ASSIGNMENT_1;
import java.util.*;
public class p10 {
    public static double sumColumn(double[][]m , int columnIndex) {
        int r = m.length;
        double sum = 0;
        for (int i=0 ; i<r ; i++) {
            sum += m[i][columnIndex];
        }
        return sum;
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the Row and Column : ");
        int r = sc.nextInt();
        int c = sc.nextInt();
        double [][]arr = new double[r][c];
        System.out.print("Enter the Array row by row : ");
        for (int i=0 ; i<r ; i++) {
            for (int j=0 ;j<c ; j++) {
                arr[i][j] = sc.nextDouble();
            }
```

```java
        }
        for (int i=0 ; i<r ; i++) {
            for (int j=0 ; j<c ; j++) {
                System.out.print(arr[i][j] + " ");
            }
            System.out.println();
        }
        for (int i=0 ; i<c ; i++) {
            System.out.println("Sum of the element at Column " + i + " is " + sumColumn(arr, i));
        }
        sc.close();
    }
}
```

Output.

```
Enter the Row and Column : 3 4
Enter the Array row by row : 1.5 2 3 4 5.5 6 7 8 9.5 1
3 1

1.5  2.0  3.0  4.0
5.5  6.0  7.0  8.0
9.5  1.0  3.0  1.0

Sum of the element at Column 0 is 16.5
Sum of the element at Column 1 is 9.0
Sum of the element at Column 2 is 13.0
Sum of the element at Column 3 is 13.0
```

HW_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.

```java
package DSA_ASSIGNMENT_1;
import java.util.*;
public class HW_1 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter three integers : ");
        int a = sc.nextInt();
        int b = sc.nextInt();
        int c = sc.nextInt();
        if (a + b == c) {
            System.out.println(a + " + " + b + " = " + c);
        } else if (a == b - c) {
            System.out.println(a + " = " + b + " - " + c);
        } else if (a * b == c) {
            System.out.println(a + " * " + b + " = " + c);
        } else {
            System.out.println("The Number can not be used in 
arithmetic formula.");
        }
        sc.close();
    }
}
```

Output.

```
Enter three integers : 1 2 3
1 + 2 = 3

Enter three integers : 2 3 4
The Number can not be used in arithmetic formula.
```

HW_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.

```java
package DSA_ASSIGNMENT_1;
import java.util.*;
public class HW_2 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter your word : ");
        String str = sc.next();
        for (int i=str.length()-1 ; i>=0 ; i--) {
            System.out.print(str.charAt(i));
        }
        sc.close();
    }
}
```

Output.

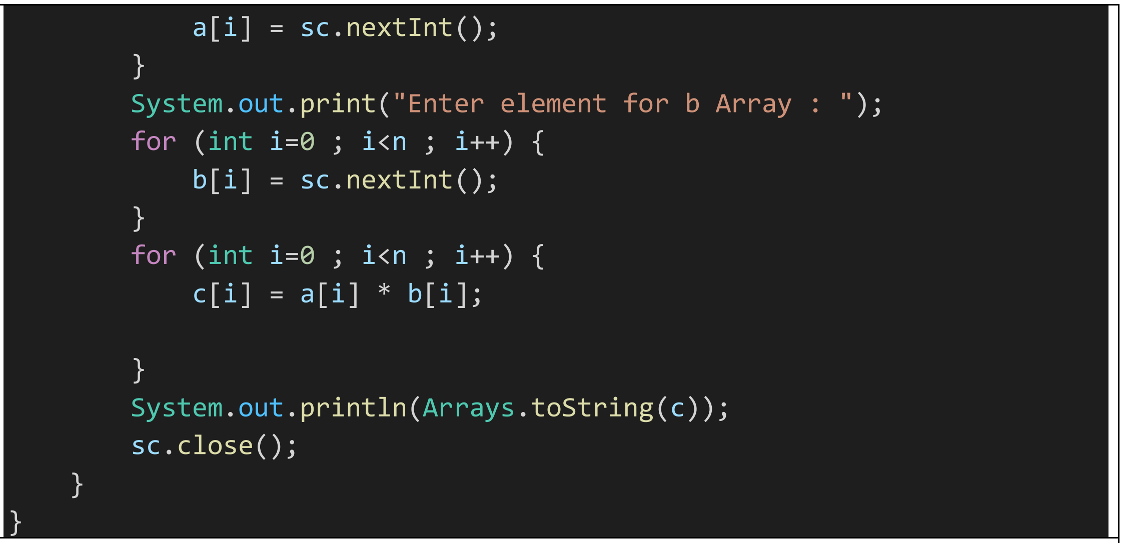
```
Enter your line : java program
The Reversed line is : margorp avaj
```

HW_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.

```java
package DSA_ASSIGNMENT_1;
import java.util.*;
public class HW_3 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the size of the Array : ");
        int n = sc.nextInt();

        int a[] = new int[n];
        int b[] = new int[n];
        int c[] = new int[n];
        System.out.print("Enter element for a Array : ");
        for (int i=0 ; i<n ; i++) {
```



```
            a[i] = sc.nextInt();
        }
        System.out.print("Enter element for b Array : ");
        for (int i=0 ; i<n ; i++) {
            b[i] = sc.nextInt();
        }
        for (int i=0 ; i<n ; i++) {
            c[i] = a[i] * b[i];

        }
        System.out.println(Arrays.toString(c));
        sc.close();
    }
}
```

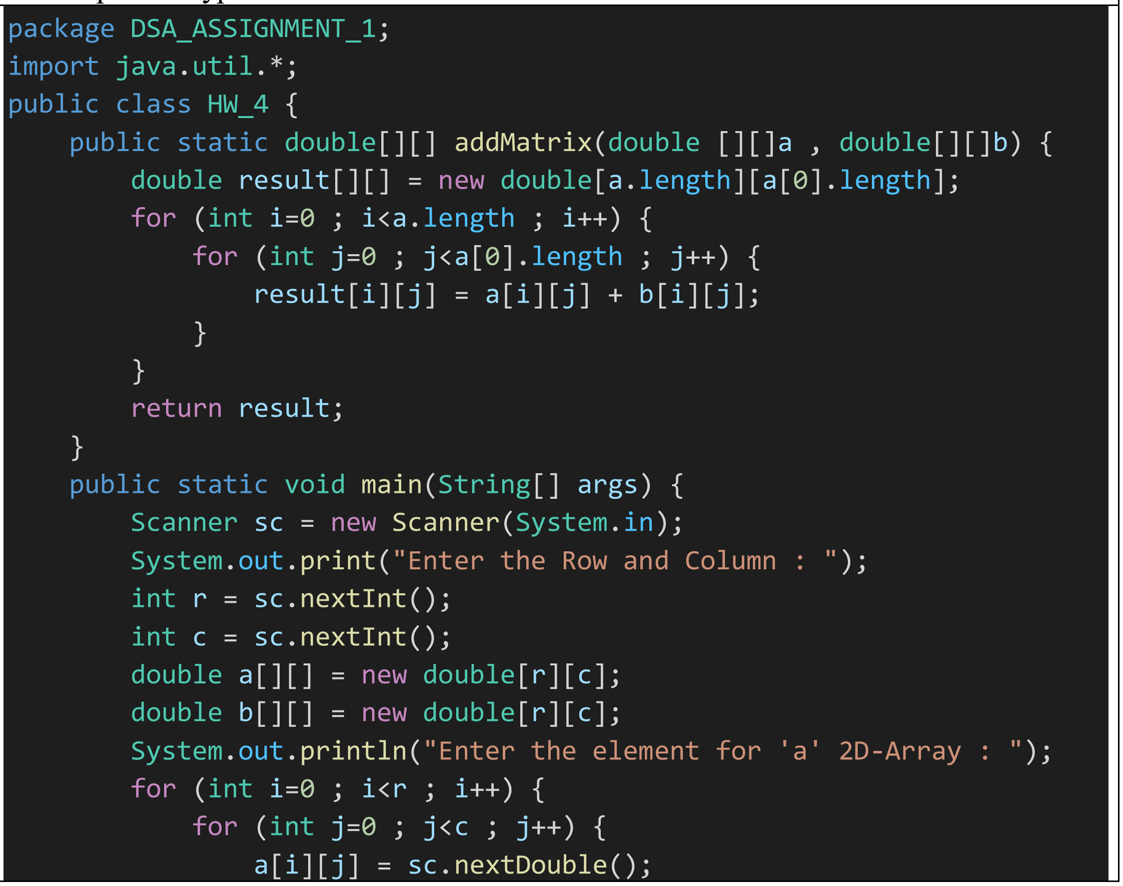
Output.

```
Enter the size of the Array : 3
Enter element for a Array : 1 2 3
Enter element for b Array : 1 2 3
[1, 4, 9]
```

HW_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 DSA_ASSIGNMENT_1;
import java.util.*;
public class HW_4 {
    public static double[][] addMatrix(double [][]a , double[][]b) {
        double result[][] = new double[a.length][a[0].length];
        for (int i=0 ; i<a.length ; i++) {
            for (int j=0 ; j<a[0].length ; j++) {
                result[i][j] = a[i][j] + b[i][j];
            }
        }
        return result;
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the Row and Column : ");
        int r = sc.nextInt();
        int c = sc.nextInt();
        double a[][] = new double[r][c];
        double b[][] = new double[r][c];
        System.out.println("Enter the element for 'a' 2D-Array : ");
        for (int i=0 ; i<r ; i++) {
            for (int j=0 ; j<c ; j++) {
                a[i][j] = sc.nextDouble();
            }
        }
        System.out.println("Enter the element for 'b' 2D-Array : ");
        for (int i=0 ; i<r ; i++) {
            for (int j=0 ; j<c ; j++) {
                b[i][j] = sc.nextDouble();
            }
        }
        double[][] arr = addMatrix(a, b);
        System.out.println("The Sum of the Two Matrix is ");
        for (int i=0 ; i<arr.length ; i++) {
            for (int j=0 ; j<arr[0].length ; j++) {
                System.out.print(arr[i][j] + " ");
            }
            System.out.println();
        }
        sc.close();
    }
}
```

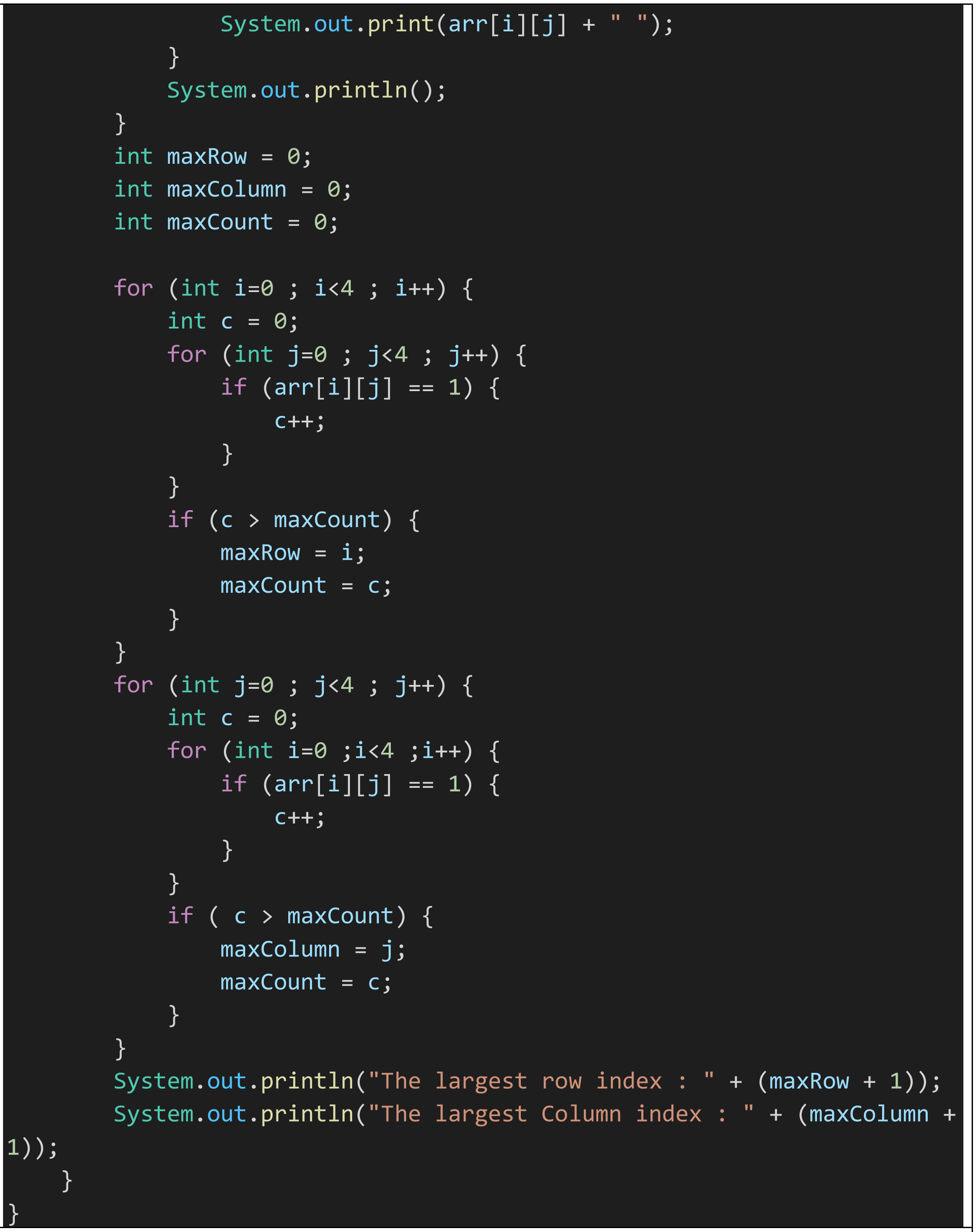
Output.

```
Enter the Row and Column : 3 3
Enter the element for 'a' 2D-Array :
1 2 3
4 5 6
7 8 9
Enter the element for 'b' 2D-Array :
1 2 3
4 5 6
7 8 9
The Sum of the Two Matrix is
2.0 4.0 6.0
8.0 10.0 12.0
14.0 16.0 18.0
```

HW_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 DSA_ASSIGNMENT_1;
public class HW_5 {
    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<4 ; i++) {
            for (int j=0 ; j<4 ; j++) {
```



```
                System.out.print(arr[i][j] + " ");
            }
            System.out.println();
        }
        int maxRow = 0;
        int maxColumn = 0;
        int maxCount = 0;

        for (int i=0 ; i<4 ; i++) {
            int c = 0;
            for (int j=0 ; j<4 ; j++) {
                if (arr[i][j] == 1) {
                    c++;
                }
            }
            if (c > maxCount) {
                maxRow = i;
                maxCount = c;
            }
        }
        for (int j=0 ; j<4 ; j++) {
            int c = 0;
            for (int i=0 ;i<4 ;i++) {
                if (arr[i][j] == 1) {
                    c++;
                }
            }
            if ( c > maxCount) {
                maxColumn = j;
                maxCount = c;
            }
        }
        System.out.println("The largest row index : " + (maxRow + 1));
        System.out.println("The largest Column index : " + (maxColumn +
1));
    }
}
```

Output.

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