FEB 2025

OOPJ ASSIGNMENT NO – 3

Ashwini Vadkar

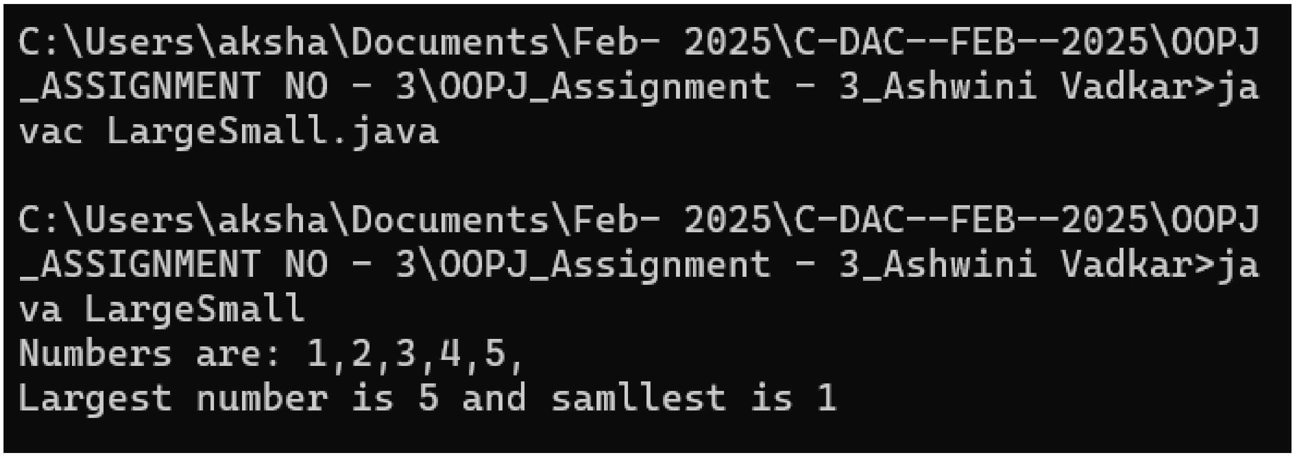
1. Find the Largest and Smallest Element
   ○ Given an array, find the smallest and largest elements in it.

Ans :-

Input :

```
class LargeSmall {

    public static void main(String args[]) {

        int a[] = {1, 2, 3, 4, 5};

        System.out.print("Numbers are: ");

        for (int i = 0; i < 5; i++) {

            System.out.print(a[i] + ",");

        }

        System.out.println(" ");

        int max = 0;

        int min = 1;

        for (int i = 0; i < 5; i++) {

            max = ((a[i]) > max) ? a[i] : max;

            min = ((a[i]) < min) ? a[i] : min;

        }

        System.out.println("Largest number is " + max + " and samllest is " + min);

    }

}
```

Output :



```
C:\Users\aksha\Documents\Feb- 2025\C-DAC--FEB--2025\OOPJ
_ASSIGNMENT NO - 3\OOPJ_Assignment - 3_Ashwini Vadkar>ja
vac LargeSmall.java

C:\Users\aksha\Documents\Feb- 2025\C-DAC--FEB--2025\OOPJ
_ASSIGNMENT NO - 3\OOPJ_Assignment - 3_Ashwini Vadkar>ja
va LargeSmall
Numbers are: 1,2,3,4,5,
Largest number is 5 and samllest is 1
```

----

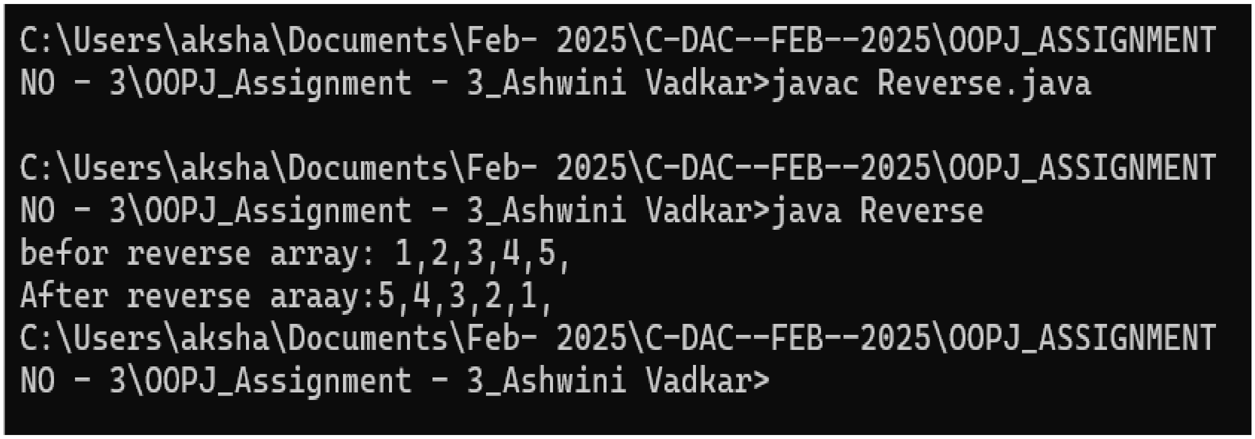
2. Reverse an Array
    ○ Reverse the given array in place.

Ans :-

Input :

```
class Reverse {
   public static void main(String args[]) {
      int a[] = {1, 2, 3, 4, 5};
      int r[] = new int[5];
      System.out.print("befor reverse array: ");
      for (int i = 0; i < 5; i++) {
         System.out.print(a[i] + ",");
      }
      System.out.println();
      for (int i = 0; i < 5; i++) {
         r[i] = a[4 - i];
      }
      System.out.print("After reverse araay:");
      for (int i = 0; i < 5; i++) {
         System.out.print(r[i] + ",");
      }
   }
}
```

Output :



```
C:\Users\aksha\Documents\Feb- 2025\C-DAC--FEB--2025\OOPJ_ASSIGNMENT
NO - 3\OOPJ_Assignment - 3_Ashwini Vadkar>javac Reverse.java

C:\Users\aksha\Documents\Feb- 2025\C-DAC--FEB--2025\OOPJ_ASSIGNMENT
NO - 3\OOPJ_Assignment - 3_Ashwini Vadkar>java Reverse
befor reverse array: 1,2,3,4,5,
After reverse araay:5,4,3,2,1,
C:\Users\aksha\Documents\Feb- 2025\C-DAC--FEB--2025\OOPJ_ASSIGNMENT
NO - 3\OOPJ_Assignment - 3_Ashwini Vadkar>
```

----

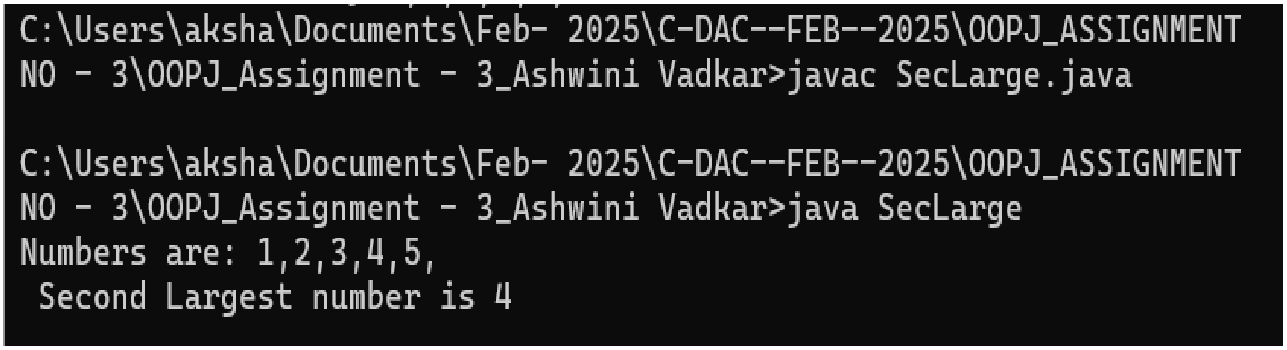
3. Find the Second Largest Element
    ○ Find the second-largest element in the given array.

Ans :-

Input :

```
class SecLarge {
    public static void main(String args[]) {
        int a[] = {1, 2, 3, 4, 5};
        System.out.print("Numbers are: ");
        for (int i = 0; i < 5; i++) {
            System.out.print(a[i] + ",");
        }
        System.out.println(" ");
        int max = 0;
        int max1 = 0;
        for (int i = 0; i < 5; i++) {
            max = ((a[i]) > max) ? a[i] : max;
        }
        for (int i = 0; i < 5; i++) {
            if ((a[i]) > max1) {
                if ((a[i]) < max) {
                    max1 = a[i];
                }
            }
        }
        System.out.println(" Second Largest number is " + max1);
    }
}
```

Output :



```
C:\Users\aksha\Documents\Feb- 2025\C-DAC--FEB--2025\OOPJ_ASSIGNMENT
NO - 3\OOPJ_Assignment - 3_Ashwini Vadkar>javac SecLarge.java

C:\Users\aksha\Documents\Feb- 2025\C-DAC--FEB--2025\OOPJ_ASSIGNMENT
NO - 3\OOPJ_Assignment - 3_Ashwini Vadkar>java SecLarge
Numbers are: 1,2,3,4,5,
 Second Largest number is 4
```

----

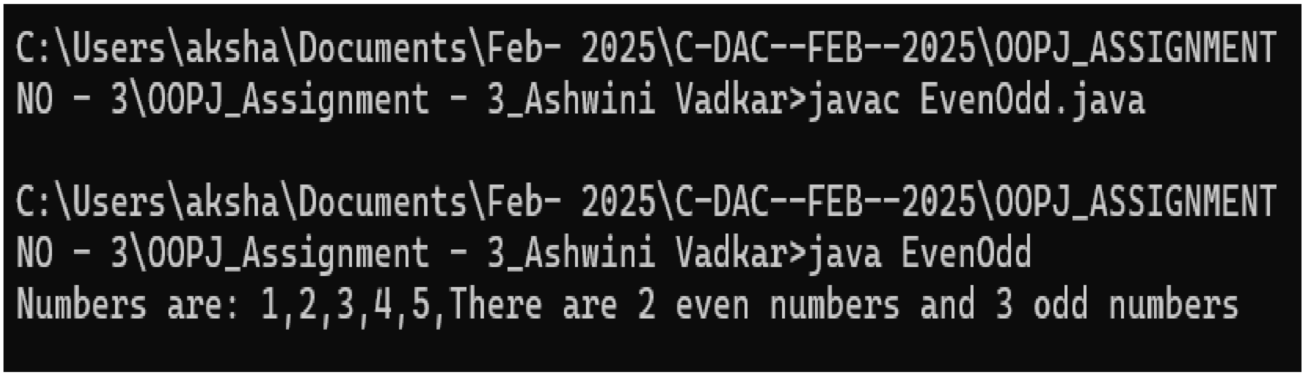
4. Count Even and Odd Numbers
   ○ Count the number of even and odd numbers in an array.

Ans :-

Input :

```
class EvenOdd {
    public static void main(String args[]) {
        int a[] = {1, 2, 3, 4, 5};
        int evencount = 0;
        int oddcount = 0;
        System.out.print("Numbers are: ");
        for (int i = 0; i < 5; i++) {
            System.out.print(a[i] + ",");
            if (((a[i]) % 2) == 0) {
                evencount++;
            } else {
                oddcount++;
            }
        }
        System.out.println("There are " + evencount + " even numbers and " + oddcount +
" odd numbers");
    }
}
```

Output :



```
C:\Users\aksha\Documents\Feb- 2025\C-DAC--FEB--2025\OOPJ_ASSIGNMENT
NO - 3\OOPJ_Assignment - 3_Ashwini Vadkar>javac EvenOdd.java

C:\Users\aksha\Documents\Feb- 2025\C-DAC--FEB--2025\OOPJ_ASSIGNMENT
NO - 3\OOPJ_Assignment - 3_Ashwini Vadkar>java EvenOdd
Numbers are: 1,2,3,4,5,There are 2 even numbers and 3 odd numbers
```

----

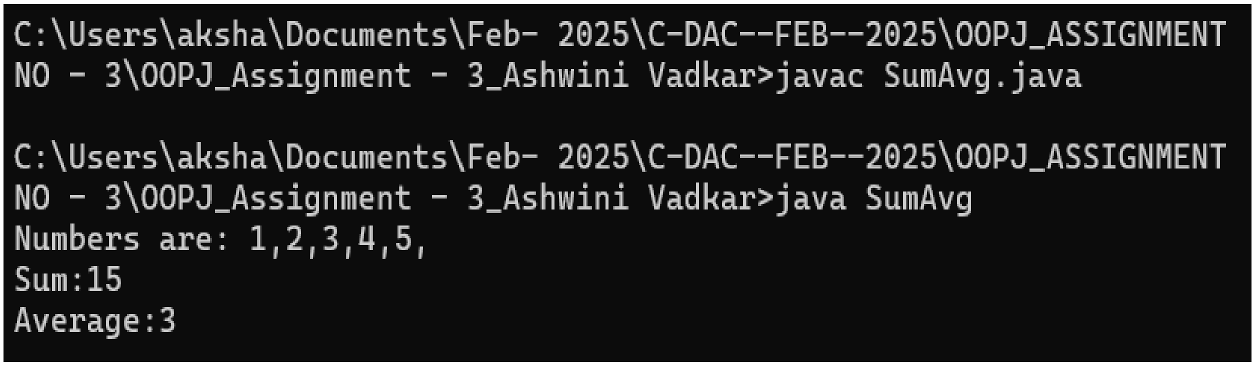
5. Find Sum and Average
   - ○ Compute the sum and average of all elements in the array

Ans :-

Input :

```
class SumAvg {
    public static void main(String args[]) {
        int a[] = {1, 2, 3, 4, 5};
        int sum = 0;
        int avg = 0;
        System.out.print("Numbers are: ");
        for (int i = 0; i < 5; i++) {
            System.out.print(a[i] + ",");
            sum += a[i];
        }
        System.out.println();
        avg = sum / a.length;
        System.out.println("Sum:" + sum);
        System.out.println("Average:" + avg);
    }
}
```

Output :



```
C:\Users\aksha\Documents\Feb- 2025\C-DAC--FEB--2025\OOPJ_ASSIGNMENT
NO - 3\OOPJ_Assignment - 3_Ashwini Vadkar>javac SumAvg.java

C:\Users\aksha\Documents\Feb- 2025\C-DAC--FEB--2025\OOPJ_ASSIGNMENT
NO - 3\OOPJ_Assignment - 3_Ashwini Vadkar>java SumAvg
Numbers are: 1,2,3,4,5,
Sum:15
Average:3
```

----

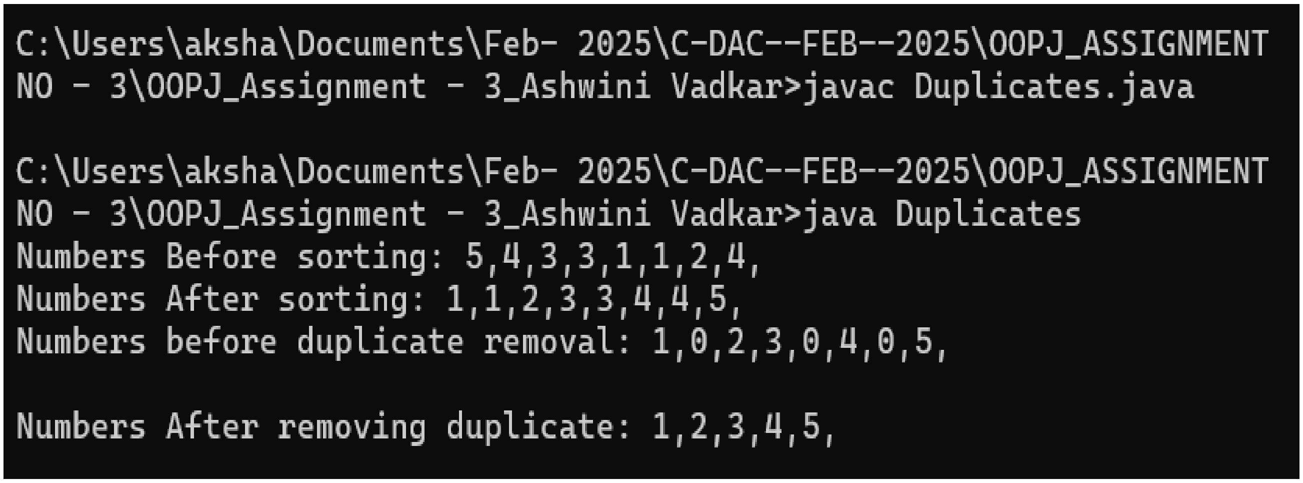
6. Remove Duplicates from a Sorted Array
    - ○ Remove duplicate elements from a sorted array without using extra space.

Ans :-

Input :

```
class Duplicates {
    public static void main(String args[]) {
        int a[] = {5, 4, 3, 3, 1, 1, 2, 4};
        System.out.print("Numbers Before sorting: ");
        for (int i = 0; i < a.length; i++) {
            System.out.print(a[i] + ",");
        }
        System.out.println();

        for (int i = 0; i < a.length - 1; i++) {
            for (int j = 0; j < a.length - 1 - i; j++) {
                if (a[j] > a[j + 1]) {
                    int temp = a[j];
                    a[j] = a[j + 1];
                    a[j + 1] = temp;
                }
            }
        }

        System.out.print("Numbers After sorting: ");
        for (int i = 0; i < a.length; i++) {
            System.out.print(a[i] + ",");
        }
        System.out.println();

        int count = 0;
        for (int i = 0; i < a.length - 1; i++) {
            for (int j = 0; j < a.length - 1 - i; j++) {
                if (a[j] == a[j + 1]) {
                    a[j + 1] = 0;
                    count++;
                }
            }
        }
        System.out.print("Numbers before duplicate removal: ");
        for (int i = 0; i < a.length; i++) {
            System.out.print(a[i] + ",");
        }
        System.out.println();

        int d[] = new int[a.length - count];
        int index = 0;
        for (int i = 0; i < a.length; i++) {
            if (a[i] != 0) {
                d[index++] = a[i];
            }
        }

        System.out.println();
        System.out.print("Numbers After removing duplicate: ");
        for (int i = 0; i < d.length; i++) {
            System.out.print(d[i] + ",");
        }
        System.out.println();
    }
}
```

Output :



```
C:\Users\aksha\Documents\Feb- 2025\C-DAC--FEB--2025\OOPJ_ASSIGNMENT
NO - 3\OOPJ_Assignment - 3_Ashwini Vadkar>javac Duplicates.java

C:\Users\aksha\Documents\Feb- 2025\C-DAC--FEB--2025\OOPJ_ASSIGNMENT
NO - 3\OOPJ_Assignment - 3_Ashwini Vadkar>java Duplicates
Numbers Before sorting: 5,4,3,3,1,1,2,4,
Numbers After sorting: 1,1,2,3,3,4,4,5,
Numbers before duplicate removal: 1,0,2,3,0,4,0,5,

Numbers After removing duplicate: 1,2,3,4,5,
```

----

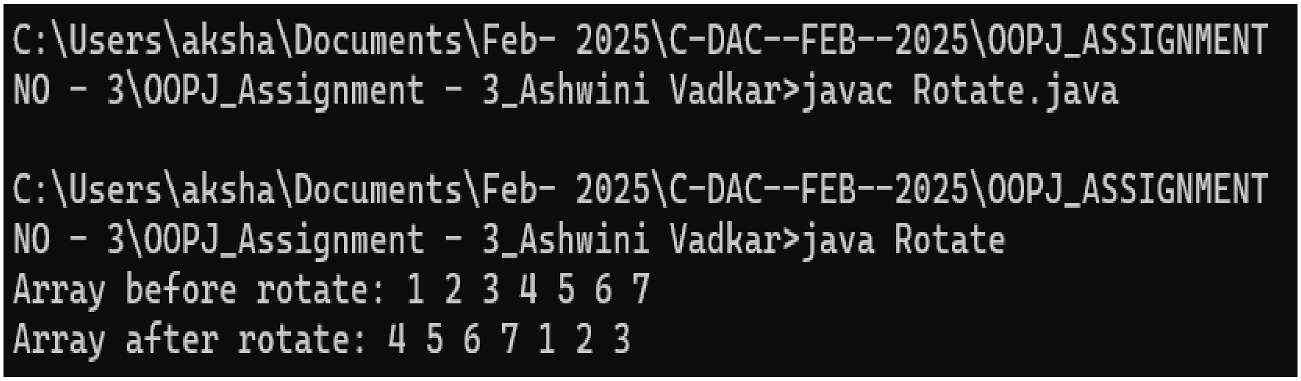
7. Rotate an Array
   ○ Rotate the array to the right by k positions.

Ans :-

Input :

```
class Rotate {
    public static void main(String args[]) {
        int a[] = {1, 2, 3, 4, 5, 6, 7};
        int r[] = new int[7];
        int k = 4;
        int count = 0;

        System.out.print("Array before rotate: ");
        for (int i = 0; i < a.length; i++) {
            System.out.print(a[i] + " ");
        }
        System.out.println();

        for (int i = 0; i < a.length; i++) {
            r[(i + k) % a.length] = a[i];
        }

        System.out.print("Array after rotate: ");
        for (int i = 0; i < a.length; i++) {
            System.out.print(r[i] + " ");
        }
        System.out.println();
    }
}
```

Output :



```
C:\Users\aksha\Documents\Feb- 2025\C-DAC--FEB--2025\OOPJ_ASSIGNMENT
NO - 3\OOPJ_Assignment - 3_Ashwini Vadkar>javac Rotate.java

C:\Users\aksha\Documents\Feb- 2025\C-DAC--FEB--2025\OOPJ_ASSIGNMENT
NO - 3\OOPJ_Assignment - 3_Ashwini Vadkar>java Rotate
Array before rotate: 1 2 3 4 5 6 7
Array after rotate: 4 5 6 7 1 2 3
```

----

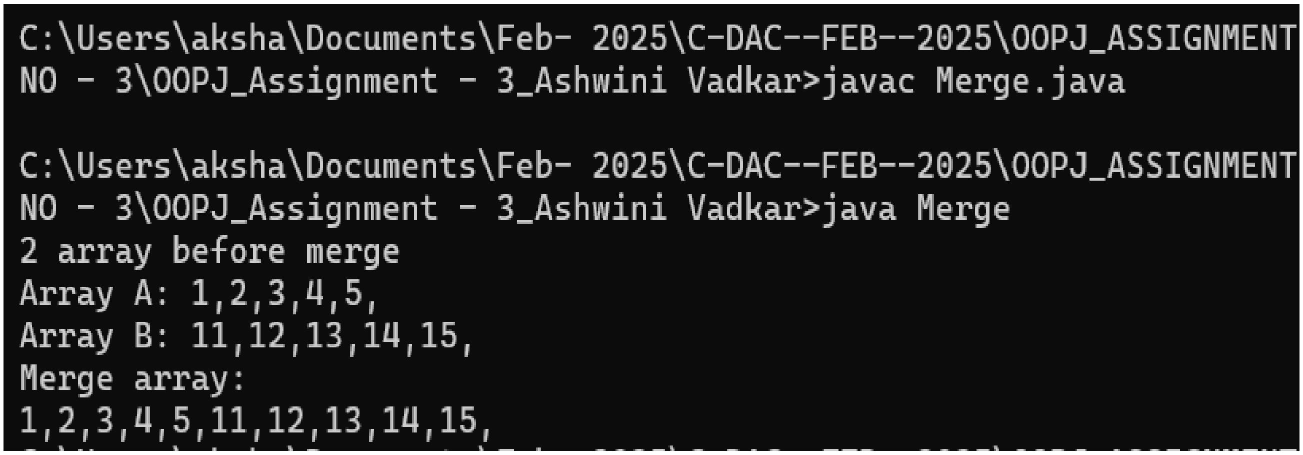
8. Merge Two Sorted Arrays
   ○ Merge two sorted arrays into a single sorted array without using extra space.

Ans :-

Input :

```
class Merge {
    public static void main(String args[]) {
        int a[] = {1, 2, 3, 4, 5};
        int b[] = {11, 12, 13, 14, 15, 16};
        int merge[] = new int[11];

        System.out.println("2 array before merge");
        System.out.print("Array A: ");
        for (int i = 0; i < 5; i++) {
            System.out.print(a[i] + ",");
            merge[i] = a[i];
        }
        System.out.println();

        System.out.print("Array B: ");
        for (int i = 0; i < 5; i++) {
            System.out.print(b[i] + ",");
            merge[i + 5] = b[i];
        }
        System.out.println();

        System.out.println("Merge array:");
        for (int i = 0; i < merge.length - 1; i++) {
            System.out.print(merge[i] + ",");
        }
    }
}
```

Output :



```
C:\Users\aksha\Documents\Feb- 2025\C-DAC--FEB--2025\OOPJ_ASSIGNMENT
NO - 3\OOPJ_Assignment - 3_Ashwini Vadkar>javac Merge.java

C:\Users\aksha\Documents\Feb- 2025\C-DAC--FEB--2025\OOPJ_ASSIGNMENT
NO - 3\OOPJ_Assignment - 3_Ashwini Vadkar>java Merge
2 array before merge
Array A: 1,2,3,4,5,
Array B: 11,12,13,14,15,
Merge array:
1,2,3,4,5,11,12,13,14,15,
```

----

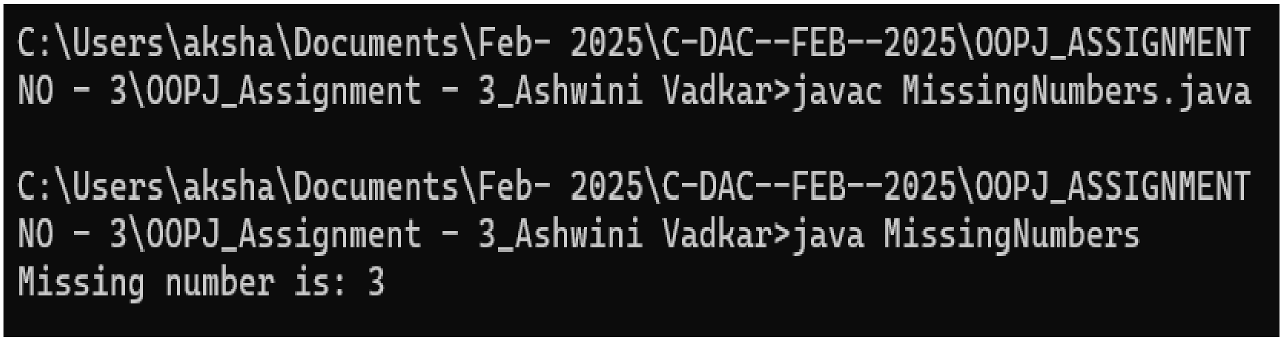
9. Find Missing Number in an Array
   - ○ Given an array of size n-1 containing numbers from 1 to n, find the missing number.

Ans :-

Input :

```
class MissingNumbers {
    public static void main(String args[]) {
        int a[] = {1, 2, 4, 5};
        int n = a.length + 1;
        int totalSum = (n * (n + 1)) / 2;
        int arraySum = 0;

        for (int i = 0; i < a.length; i++) {
            arraySum += a[i];
        }

        int missingNumber = totalSum - arraySum;
        System.out.println("Missing number is: " + missingNumber);
    }
}
```

Output :



```
C:\Users\aksha\Documents\Feb- 2025\C-DAC--FEB--2025\OOPJ_ASSIGNMENT
NO - 3\OOPJ_Assignment - 3_Ashwini Vadkar>javac MissingNumbers.java

C:\Users\aksha\Documents\Feb- 2025\C-DAC--FEB--2025\OOPJ_ASSIGNMENT
NO - 3\OOPJ_Assignment - 3_Ashwini Vadkar>java MissingNumbers
Missing number is: 3
```

----

10. Find Intersection and Union of Two Arrays

    Find the intersection and union of two unsorted arrays.

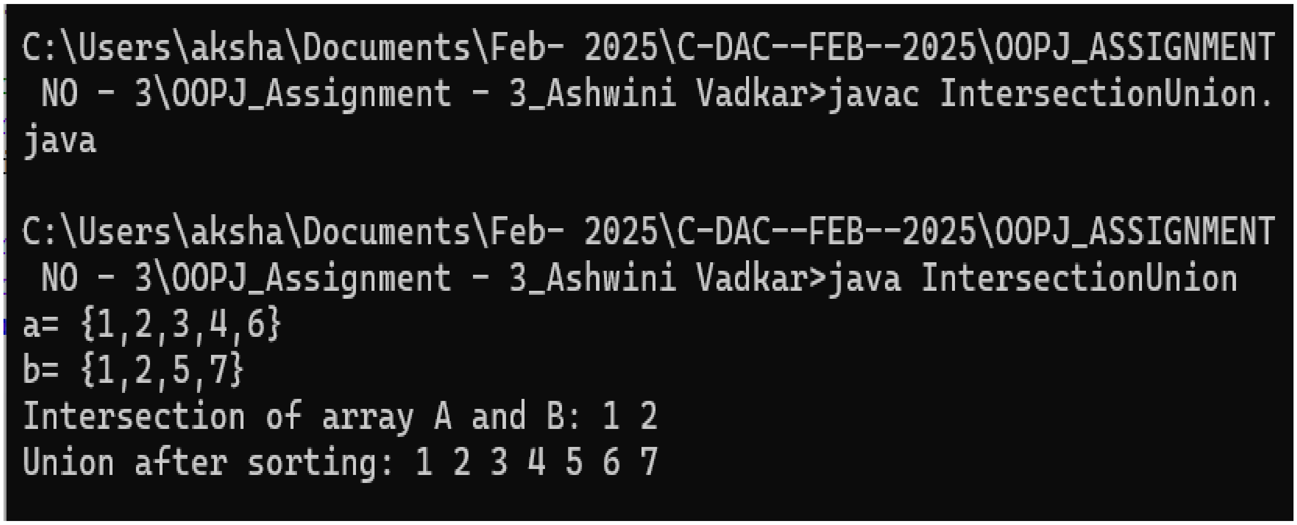
Ans :-

Input :

```
class IntersectionUnion {
    public static void main(String args[]) {
        int a[] = {1, 2, 3, 4, 6};
        int b[] = {1, 2, 5, 7};
        int s = (a.length < b.length) ? a.length : b.length;
        int inter[] = new int[s];
        int union[] = new int[a.length + b.length];
        int interIndex = 0;
        int unionIndex = 0;

        System.out.println("a= {1,2,3,4,6}");
        System.out.println("b= {1,2,5,7}");

        // Finding intersection
        for (int i = 0; i < a.length; i++) {
            for (int j = 0; j < b.length; j++) {
                if (a[i] == b[j]) {
                    inter[interIndex++] = a[i];
                    break;
                }
            }
        }

        System.out.print("Intersection of array A and B: ");
        for (int i = 0; i < interIndex; i++) {
            System.out.print(inter[i] + " ");
        }
        System.out.println();

        // Finding union
        for (int i = 0; i < a.length; i++) {
            union[unionIndex++] = a[i];
        }
        for (int i = 0; i < b.length; i++) {
            boolean found = false;
            for (int j = 0; j < a.length; j++) {
                if (b[i] == a[j]) {
                    found = true;
                    break;
                }
            }
            if (!found) {
                union[unionIndex++] = b[i];
            }
```

```
        }

        // Sorting union array
        for (int i = 0; i < unionIndex - 1; i++) {
            for (int j = 0; j < unionIndex - 1 - i; j++) {
                if (union[j] > union[j + 1]) {
                    int temp = union[j];
                    union[j] = union[j + 1];
                    union[j + 1] = temp;
                }
            }
        }

        System.out.print("Union after sorting: ");
        for (int i = 0; i < unionIndex; i++) {
            System.out.print(union[i] + " ");
        }
        System.out.println();
    }
}
```

Output :



```
C:\Users\aksha\Documents\Feb- 2025\C-DAC--FEB--2025\OOPJ_ASSIGNMENT
 NO - 3\OOPJ_Assignment - 3_Ashwini Vadkar>javac IntersectionUnion.
java

C:\Users\aksha\Documents\Feb- 2025\C-DAC--FEB--2025\OOPJ_ASSIGNMENT
 NO - 3\OOPJ_Assignment - 3_Ashwini Vadkar>java IntersectionUnion
a= {1,2,3,4,6}
b= {1,2,5,7}
Intersection of array A and B: 1 2
Union after sorting: 1 2 3 4 5 6 7
```

----

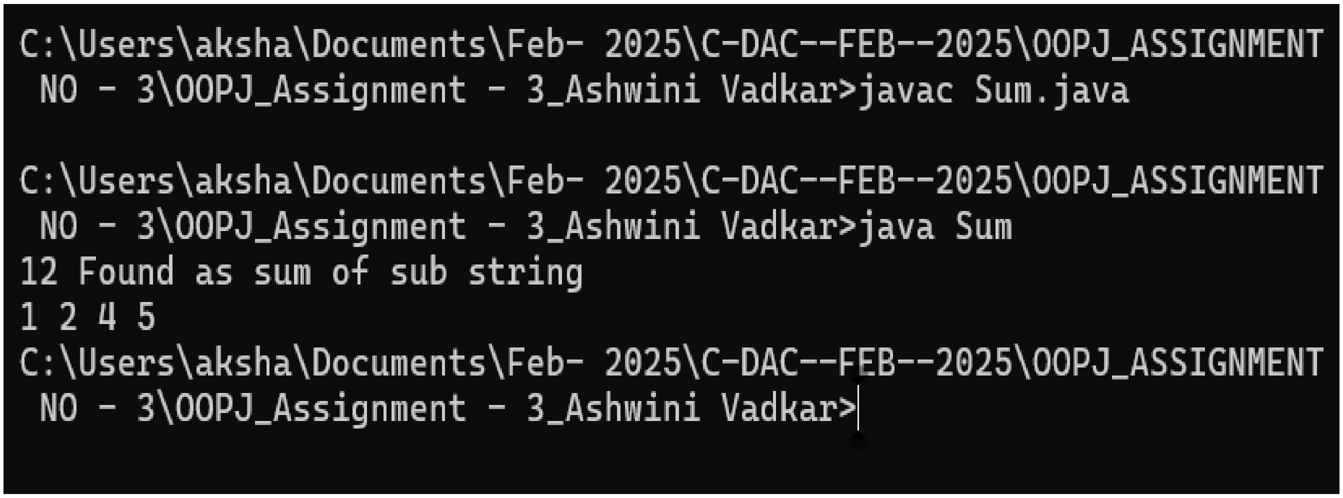
11. Find a Subarray with Given Sum
    Given an array of integers, find the subarray that sums to a given value S.

Ans :-

Input :

```
class Sum {
    public static void main(String args[]) {
        int a[] = {1, 2, 4, 5, 6, 7};
        int s[] = new int[a.length];
        int sum = 0;
        int S = 12;

        for (int i = 0; i <= a.length - 1; i++) {
            if (sum <= S) {
                sum += a[i];
                s[i] = a[i];
                if (sum == S) {
                    System.out.println(S + " Found as sum of sub string");
                    for (int f = 0; f <= i; f++) {
                        System.out.print(s[f] + " ");
                    }
                    break;
                }
            } else {
                System.out.println(S + " as sum of sub string not found");
            }
        }
    }
}
```

Output :



```
C:\Users\aksha\Documents\Feb- 2025\C-DAC--FEB--2025\OOPJ_ASSIGNMENT
 NO - 3\OOPJ_Assignment - 3_Ashwini Vadkar>javac Sum.java

C:\Users\aksha\Documents\Feb- 2025\C-DAC--FEB--2025\OOPJ_ASSIGNMENT
 NO - 3\OOPJ_Assignment - 3_Ashwini Vadkar>java Sum
12 Found as sum of sub string
1 2 4 5
C:\Users\aksha\Documents\Feb- 2025\C-DAC--FEB--2025\OOPJ_ASSIGNMENT
 NO - 3\OOPJ_Assignment - 3_Ashwini Vadkar>
```

----

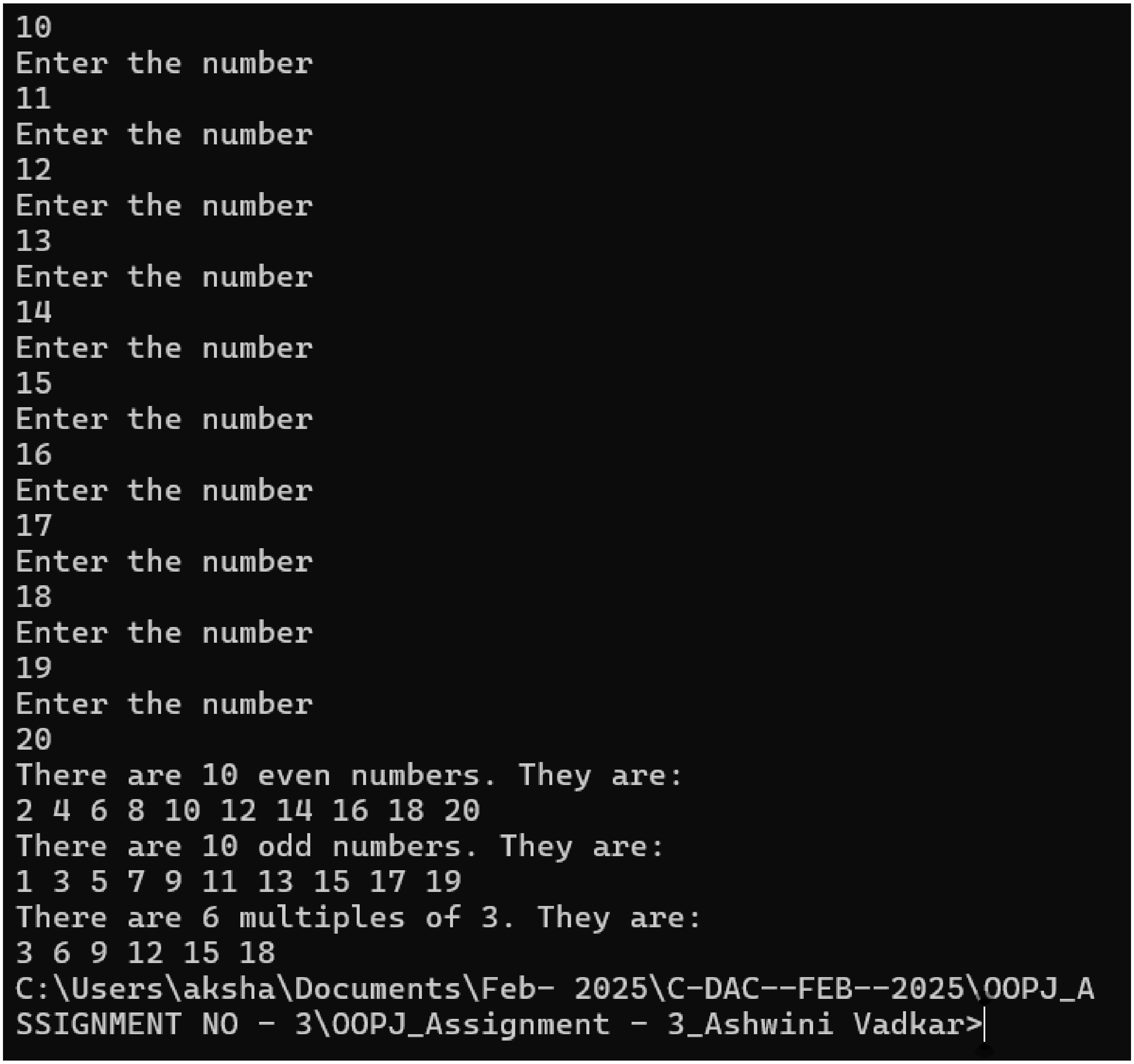
12. Write a program to accept 20 integer numbers in a single Dimensional Array.
    Find and Display the following:
    Number of even numbers.
    Number of odd numbers.
    Number of multiples of 3.

Ans :-

Input :

```
import java.util.Scanner;

class FindDisplay {
    public static void main(String args[]) {
        Scanner input = new Scanner(System.in);
        int a[] = new int[20];
        int even[] = new int[20];
        int odd[] = new int[20];
        int mult[] = new int[20];
        int evencount = 0;
        int oddcount = 0;
        int multcount = 0;

        for (int i = 0; i < 20; i++) {
            System.out.println("Enter the number");
            a[i] = input.nextInt();
        }

        for (int i = 0; i < 20; i++) {
            if ((a[i] % 2) == 0) {
                even[evencount] = a[i];
                evencount++;
            } else {
                odd[oddcount] = a[i];
                oddcount++;
            }

            if ((a[i] % 3) == 0) {
                mult[multcount] = a[i];
                multcount++;
            }
        }
        System.out.println("There are " + evencount + " even numbers. They are:");
        for (int i = 0; i < evencount; i++) {
            System.out.print(even[i] + " ");
        }
        System.out.println();

        System.out.println("There are " + oddcount + " odd numbers. They are:");
        for (int i = 0; i < oddcount; i++) {
            System.out.print(odd[i] + " ");
        }
        System.out.println();

        System.out.println("There are " + multcount + " multiples of 3. They are:");
        for (int i = 0; i < multcount; i++) {
            System.out.print(mult[i] + " ");
        }

        input.close();
    }
}
```

Output :



```
10
Enter the number
11
Enter the number
12
Enter the number
13
Enter the number
14
Enter the number
15
Enter the number
16
Enter the number
17
Enter the number
18
Enter the number
19
Enter the number
20
There are 10 even numbers. They are:
2 4 6 8 10 12 14 16 18 20
There are 10 odd numbers. They are:
1 3 5 7 9 11 13 15 17 19
There are 6 multiples of 3. They are:
3 6 9 12 15 18
C:\Users\aksha\Documents\Feb- 2025\C-DAC--FEB--2025\OOPJ_A
SSIGNMENT NO - 3\OOPJ_Assignment - 3_Ashwini Vadkar>
```

----

13. Write a program to accept the marks in Physics, Chemistry and Maths secured by 20 class students in a single Dimensional Array. Find and display the following:

    Number of students securing 75% and above in aggregate.

    Number of students securing 40% and below in aggregate.

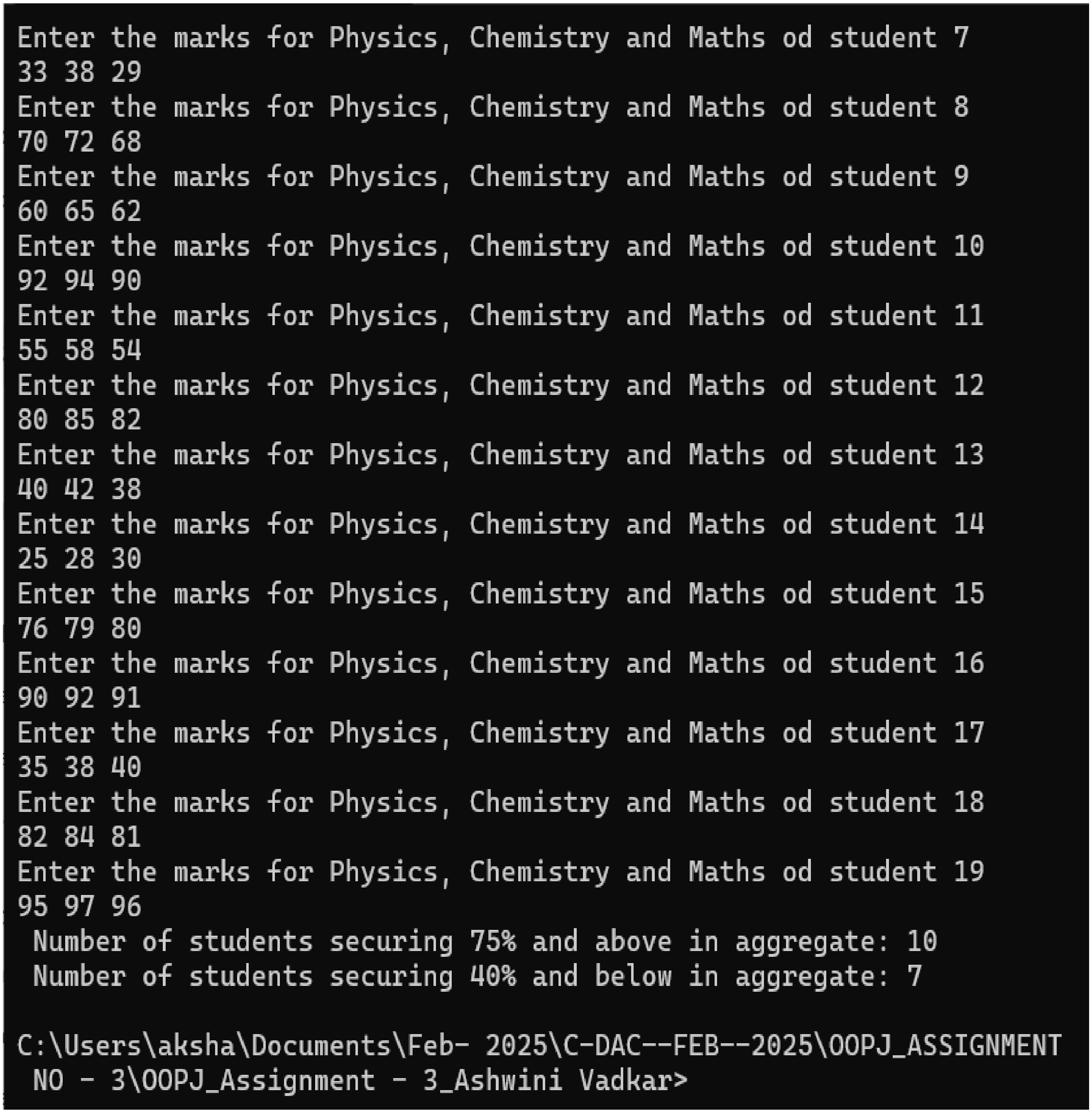
    Ans :-

    Input :

    ```
    import java.util.Scanner;

    class Aggregate {
    ```

```
    public static void main(String args[]) {
        Scanner input = new Scanner(System.in);
        int p[] = new int[20];
        int c[] = new int[20];
        int m[] = new int[20];

        for (int i = 0; i < 20; i++) {
            System.out.println("Enter the marks for Physics, Chemistry and Maths od
student " + i);
            p[i] = input.nextInt();
            c[i] = input.nextInt();
            m[i] = input.nextInt();
        }

        int above75 = 0;
        int below45 = 0;

        for (int i = 0; i < 20; i++) {
            int total = p[i] + c[i] + m[i];
            double agg = total / 3;
            if (agg >= 75) {
                above75++;
            } else if (agg <= 45) {
                below45++;
            }
        }

        System.out.println(" Number of students securing 75% and above in
aggregate: " + above75);
        System.out.println(" Number of students securing 40% and below in
aggregate: " + below45);
    }
}
```

Output :



```
Enter the marks for Physics, Chemistry and Maths od student 7
33 38 29
Enter the marks for Physics, Chemistry and Maths od student 8
70 72 68
Enter the marks for Physics, Chemistry and Maths od student 9
60 65 62
Enter the marks for Physics, Chemistry and Maths od student 10
92 94 90
Enter the marks for Physics, Chemistry and Maths od student 11
55 58 54
Enter the marks for Physics, Chemistry and Maths od student 12
80 85 82
Enter the marks for Physics, Chemistry and Maths od student 13
40 42 38
Enter the marks for Physics, Chemistry and Maths od student 14
25 28 30
Enter the marks for Physics, Chemistry and Maths od student 15
76 79 80
Enter the marks for Physics, Chemistry and Maths od student 16
90 92 91
Enter the marks for Physics, Chemistry and Maths od student 17
35 38 40
Enter the marks for Physics, Chemistry and Maths od student 18
82 84 81
Enter the marks for Physics, Chemistry and Maths od student 19
95 97 96
 Number of students securing 75% and above in aggregate: 10
 Number of students securing 40% and below in aggregate: 7

C:\Users\aksha\Documents\Feb- 2025\C-DAC--FEB--2025\OOPJ_ASSIGNMENT
 NO - 3\OOPJ_Assignment - 3_Ashwini Vadkar>
```

----

14. Write a program in Java to accept 20 numbers in a single dimensional array arr[20]. Transfer and store all the even numbers in an array even[ ] and all the odd numbers in another array odd[ ]. Finally, print the elements of the even & the odd array.

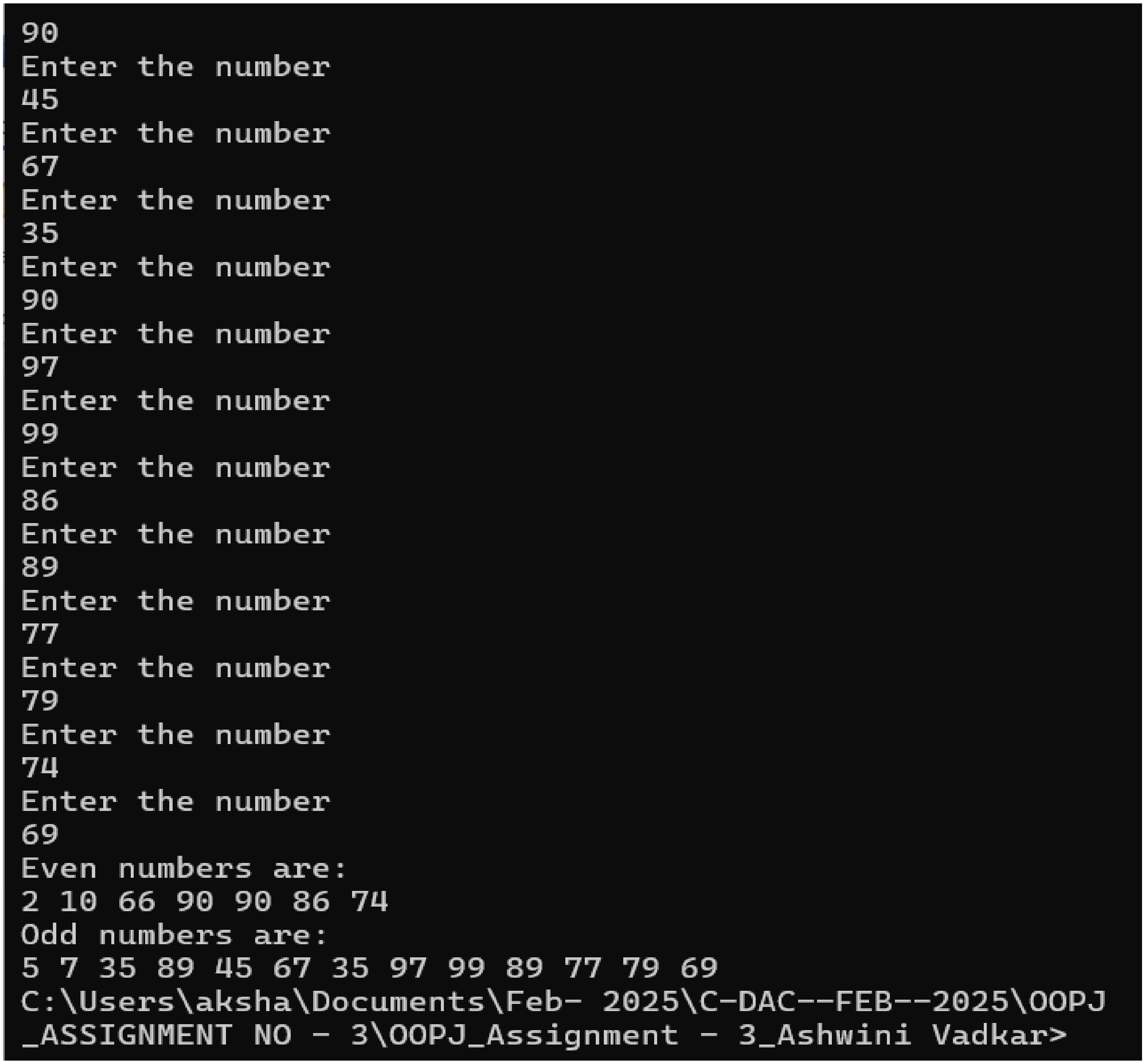
Ans :-

Input :

import java.util.Scanner;

```
class Even {
    public static void main(String args[]) {
        Scanner input = new Scanner(System.in);
        int[] a = new int[20];
        int[] even = new int[20];
        int[] odd = new int[20];
        int evencount = 0;
        int oddcount = 0;

        for (int i = 0; i < 20; i++) {
            System.out.println("Enter the number");
            a[i] = input.nextInt();
            if ((a[i] % 2) == 0) {
                even[evencount] = a[i];
                evencount++;
            } else {
                odd[oddcount] = a[i];
                oddcount++;
            }
        }

        System.out.println("Even numbers are: ");
        for (int i = 0; i < evencount; i++) {
            System.out.print(even[i] + " ");
        }
        System.out.println();

        System.out.println("Odd numbers are: ");
        for (int i = 0; i < oddcount; i++) {
            System.out.print(odd[i] + " ");
        }
    }
}
```

Output :



```
90
Enter the number
45
Enter the number
67
Enter the number
35
Enter the number
90
Enter the number
97
Enter the number
99
Enter the number
86
Enter the number
89
Enter the number
77
Enter the number
79
Enter the number
74
Enter the number
69
Even numbers are:
2 10 66 90 90 86 74
Odd numbers are:
5 7 35 89 45 67 35 97 99 89 77 79 69
C:\Users\aksha\Documents\Feb- 2025\C-DAC--FEB--2025\OOPJ
_ASSIGNMENT NO - 3\OOPJ_Assignment - 3_Ashwini Vadkar>
```

----

15. Write a Java program to print all sub-arrays with 0 sum present in a given array of integers.
    Example:
    Input :
    nums1 = { 1, 3, -7, 3, 2, 3, 1, -3, -2, -2 }
    nums2 = { 1, 2, -3, 4, 5, 6 }
    nums3= { 1, 2, -2, 3, 4, 5, 6 }
    Output: Sub-arrays with 0 sum : [1, 3, -7, 3]
    Sub-arrays with 0 sum : [3, -7, 3, 2, 3, 1, -3, -2]
    Sub-arrays with 0 sum : [1, 2, -3] Sub-arrays with 0 sum : [2, -2]

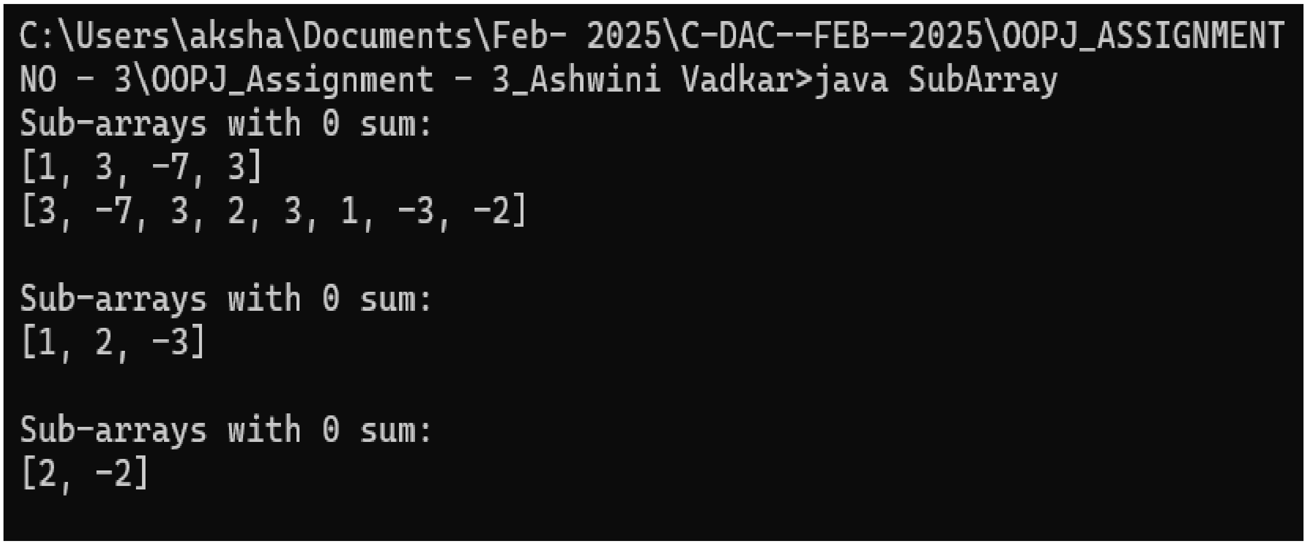
Ans:-

Input:

```
import java.util.ArrayList;
import java.util.List;

class SubArray {
    public static void main(String args[]) {
        int[][] testCases = {
            {1, 3, -7, 3, 2, 3, 1, -3, -2, -2},
            {1, 2, -3, 4, 5, 6},
            {1, 2, -2, 3, 4, 5, 6}
        };

        for (int[] nums : testCases) {
            System.out.println("Sub-arrays with 0 sum:");
            findZeroSumSubarrays(nums);
            System.out.println();
        }
    }

    public static void findZeroSumSubarrays(int[] arr) {
        for (int start = 0; start < arr.length; start++) {
            int sum = 0;
            for (int end = start; end < arr.length; end++) {
                sum += arr[end];
                if (sum == 0) {
                    printSubArray(arr, start, end);
                }
            }
        }
    }

    public static void printSubArray(int[] arr, int start, int end) {
        System.out.print("[");
        for (int i = start; i <= end; i++) {
            System.out.print(arr[i]);
            if (i < end) {
                System.out.print(", ");
            }
        }
```

```
            System.out.println("]");
        }
    }
```

Output :



```
C:\Users\aksha\Documents\Feb- 2025\C-DAC--FEB--2025\OOPJ_ASSIGNMENT
NO - 3\OOPJ_Assignment - 3_Ashwini Vadkar>java SubArray
Sub-arrays with 0 sum:
[1, 3, -7, 3]
[3, -7, 3, 2, 3, 1, -3, -2]

Sub-arrays with 0 sum:
[1, 2, -3]

Sub-arrays with 0 sum:
[2, -2]
```

----

16. Given two sorted arrays A and B of size p and q, write a Java program to merge elements of A with B by maintaining the sorted order i.e. fill A with first p smallest elements and fill B with remaining elements.
    Example:
    Input :
    int[] A = { 1, 5, 6, 7, 8, 10 }
    int[] B = { 2, 4, 9 }
    Output: Sorted Arrays:
    A: [1, 2, 4, 5, 6, 7]
    B: [8, 9, 10]

Ans :-

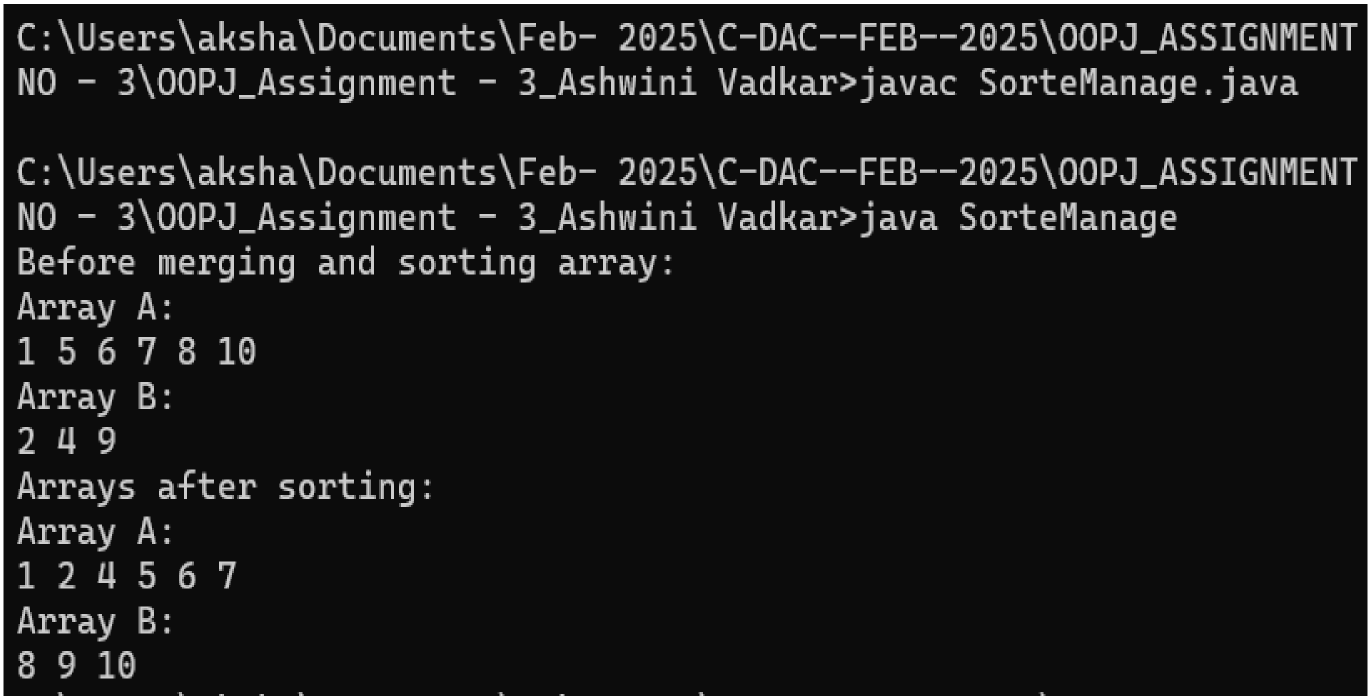
Input :

```
class SorteManage {
    public static void main(String args[]) {
        int a[] = {1, 5, 6, 7, 8, 10};
        int b[] = {2, 4, 9};
        int c[] = new int[a.length + b.length];

        System.out.println("Before merging and sorting array:");
        System.out.println("Array A: ");
        for (int i = 0; i < a.length; i++) {
```

```
    System.out.print(a[i] + " ");
}
System.out.println();
System.out.println("Array B: ");
for (int i = 0; i < b.length; i++) {
    System.out.print(b[i] + " ");
}
System.out.println();

for (int i = 0; i < a.length; i++) {
    c[i] = a[i];
}
for (int i = 0; i < b.length; i++) {
    c[a.length + i] = b[i];
}

for (int i = 0; i < c.length - 1; i++) {
    for (int j = 0; j < c.length - 1 - i; j++) {
        if (c[j] > c[j + 1]) {
            int temp = c[j];
            c[j] = c[j + 1];
            c[j + 1] = temp;
        }
    }
}

for (int i = 0; i < a.length; i++) {
    a[i] = c[i];
}
for (int i = 0; i < b.length; i++) {
    b[i] = c[a.length + i];
}

System.out.println("Arrays after sorting:");
System.out.println("Array A: ");
for (int i = 0; i < a.length; i++) {
    System.out.print(a[i] + " ");
}
System.out.println();
System.out.println("Array B: ");
```

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

Output :



```
C:\Users\aksha\Documents\Feb- 2025\C-DAC--FEB--2025\OOPJ_ASSIGNMENT
NO - 3\OOPJ_Assignment - 3_Ashwini Vadkar>javac SorteManage.java

C:\Users\aksha\Documents\Feb- 2025\C-DAC--FEB--2025\OOPJ_ASSIGNMENT
NO - 3\OOPJ_Assignment - 3_Ashwini Vadkar>java SorteManage
Before merging and sorting array:
Array A:
1 5 6 7 8 10
Array B:
2 4 9
Arrays after sorting:
Array A:
1 2 4 5 6 7
Array B:
8 9 10
```

----

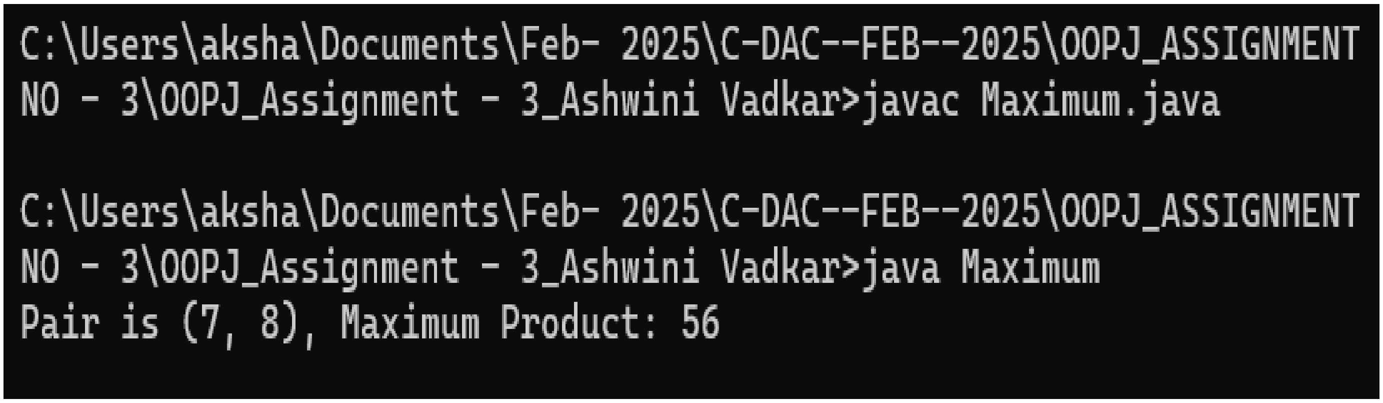
17. Write a Java program to find the maximum product of two integers in a given array of integers.
    Example:
    Input :
    nums = { 2, 3, 5, 7, -7, 5, 8, -5 }
    Output: Pair is (7, 8),
    Maximum Product: 56

Ans :-
Input :

class Maximum { public static void main(String args[]) { int nums[] = { 2, 3, 5, 7, -7, 5, 8, -5 }; int max1 = Integer.MIN_VALUE, max2 = Integer.MIN_VALUE;

```
    for (int i = 0; i < nums.length; i++) {
        if (nums[i] > max1) {
                max2 = max1;
                max1 = nums[i];
            } else if (nums[i] > max2 && nums[i] < max1) {
                max2 = nums[i];
            }
        }

        int maxProduct = max1 * max2;
        System.out.println("Pair is (" + max2 + ", " + max1 + "), Maximum
Product: " + maxProduct);
}


}
```

Output :



```
C:\Users\aksha\Documents\Feb- 2025\C-DAC--FEB--2025\OOPJ_ASSIGNMENT
NO - 3\OOPJ_Assignment - 3_Ashwini Vadkar>javac Maximum.java

C:\Users\aksha\Documents\Feb- 2025\C-DAC--FEB--2025\OOPJ_ASSIGNMENT
NO - 3\OOPJ_Assignment - 3_Ashwini Vadkar>java Maximum
Pair is (7, 8), Maximum Product: 56
```

----

18. Print a Matrix

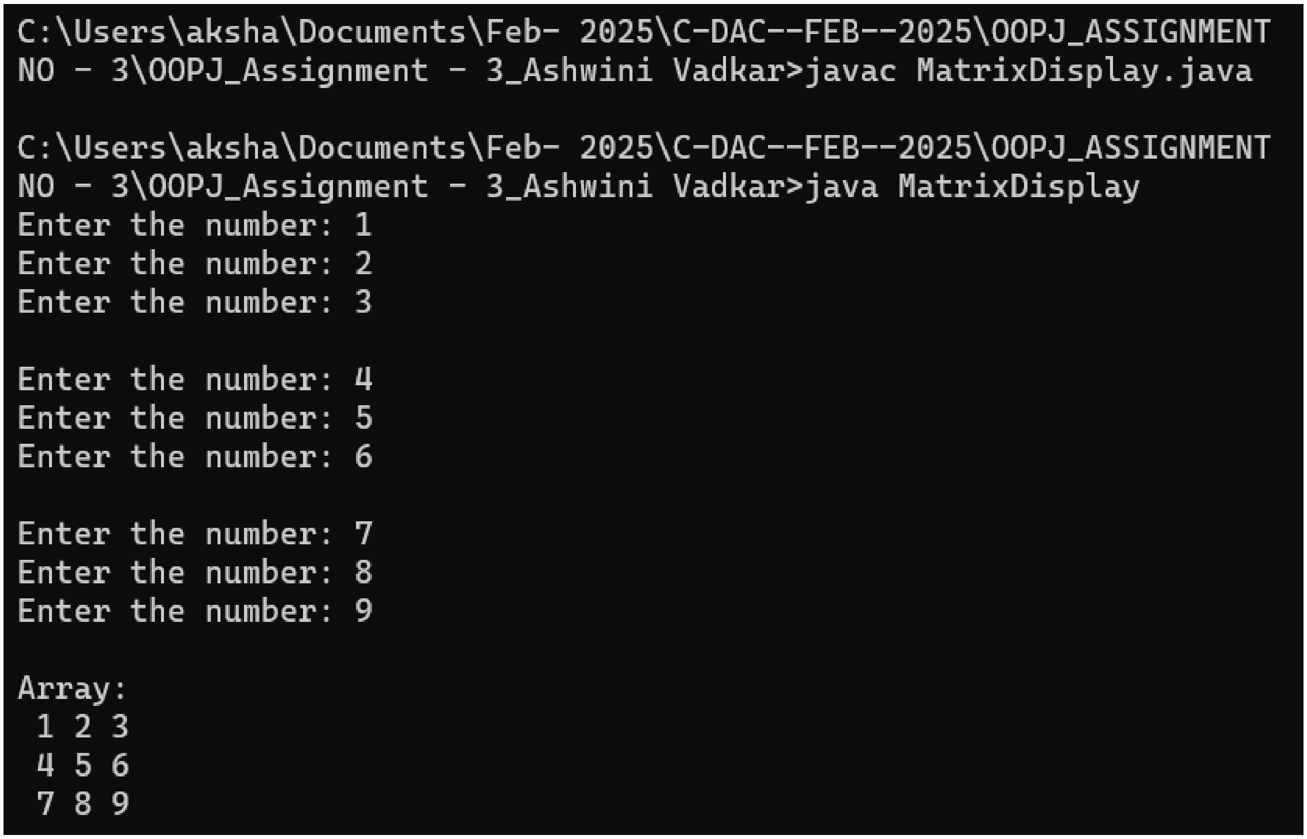
    Given an m x n matrix, print all its elements row-wise.

    Ans :-

    Input :

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

        for (int i = 0; i < 3; i++) {
            for (int j = 0; j < 3; j++) {
                System.out.print("Enter the number: ");
                a[i][j] = input.nextInt();
            }
            System.out.println();
        }

        System.out.println("Array:");
        for (int ar[] : a) {
            for (int x : ar) {
                System.out.print(" " + x);
            }
            System.out.println();
        }
    }
}
```

Output :



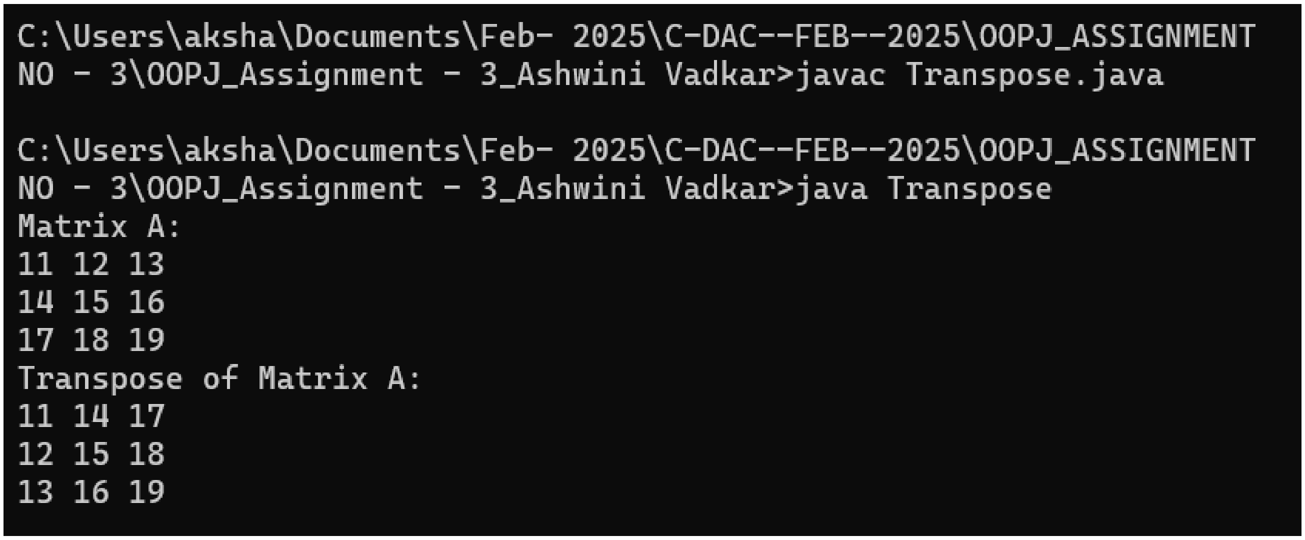
```
C:\Users\aksha\Documents\Feb- 2025\C-DAC--FEB--2025\OOPJ_ASSIGNMENT
NO - 3\OOPJ_Assignment - 3_Ashwini Vadkar>javac MatrixDisplay.java

C:\Users\aksha\Documents\Feb- 2025\C-DAC--FEB--2025\OOPJ_ASSIGNMENT
NO - 3\OOPJ_Assignment - 3_Ashwini Vadkar>java MatrixDisplay
Enter the number: 1
Enter the number: 2
Enter the number: 3

Enter the number: 4
Enter the number: 5
Enter the number: 6

Enter the number: 7
Enter the number: 8
Enter the number: 9

Array:
 1 2 3
 4 5 6
 7 8 9
```

----

19. Transpose of a Matrix
Given a matrix, return its transpose (swap rows and columns).
Ans :-
Input :

```
class Transpose {
    public static void main(String args[]) {
        int a[][] = { {11, 12, 13}, {14, 15, 16}, {17, 18, 19} };
        int at[][] = new int[3][3];

        System.out.println("Matrix A: ");
        for (int i = 0; i < 3; i++) {
            for (int j = 0; j < 3; j++) {
                System.out.print(a[i][j] + " ");
            }
            System.out.println();
        }

        System.out.println("Transpose of Matrix A: ");
        for (int i = 0; i < 3; i++) {
            for (int j = 0; j < 3; j++) {
                at[j][i] = a[i][j];
            }
        }

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

Output :



```
C:\Users\aksha\Documents\Feb- 2025\C-DAC--FEB--2025\OOPJ_ASSIGNMENT
NO - 3\OOPJ_Assignment - 3_Ashwini Vadkar>javac Transpose.java

C:\Users\aksha\Documents\Feb- 2025\C-DAC--FEB--2025\OOPJ_ASSIGNMENT
NO - 3\OOPJ_Assignment - 3_Ashwini Vadkar>java Transpose
Matrix A:
11 12 13
14 15 16
17 18 19
Transpose of Matrix A:
11 14 17
12 15 18
13 16 19
```

----

20. Sum of Two Matrices

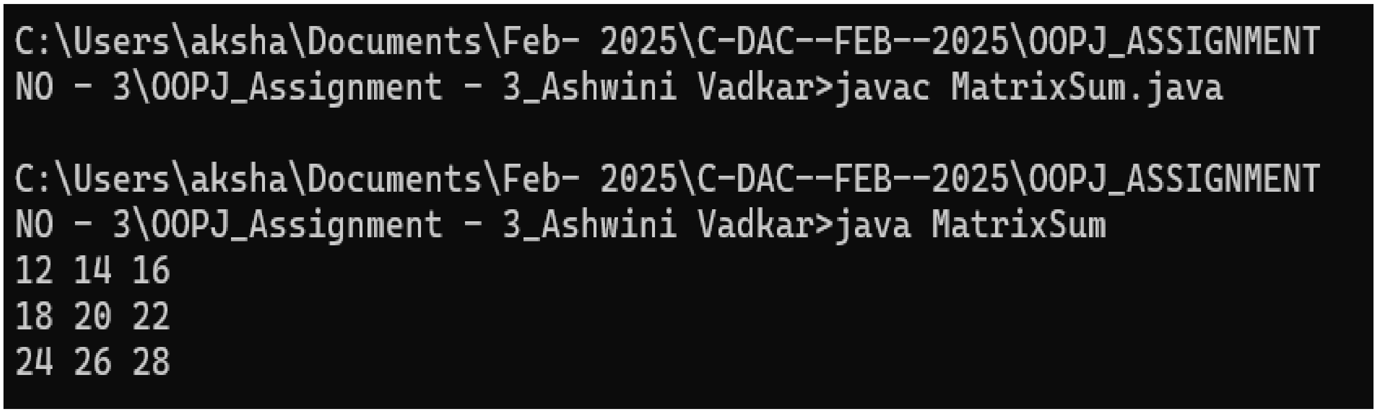
    Given two matrices of the same size, compute their sum.

Ans :-

Input :

```
class MatrixSum {
    public static void main(String args[]) {
        int a[][] = { {1, 2, 3}, {4, 5, 6}, {7, 8, 9} };
        int b[][] = { {11, 12, 13}, {14, 15, 16}, {17, 18, 19} };
        int sum[][] = new int[3][3];

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

Output :



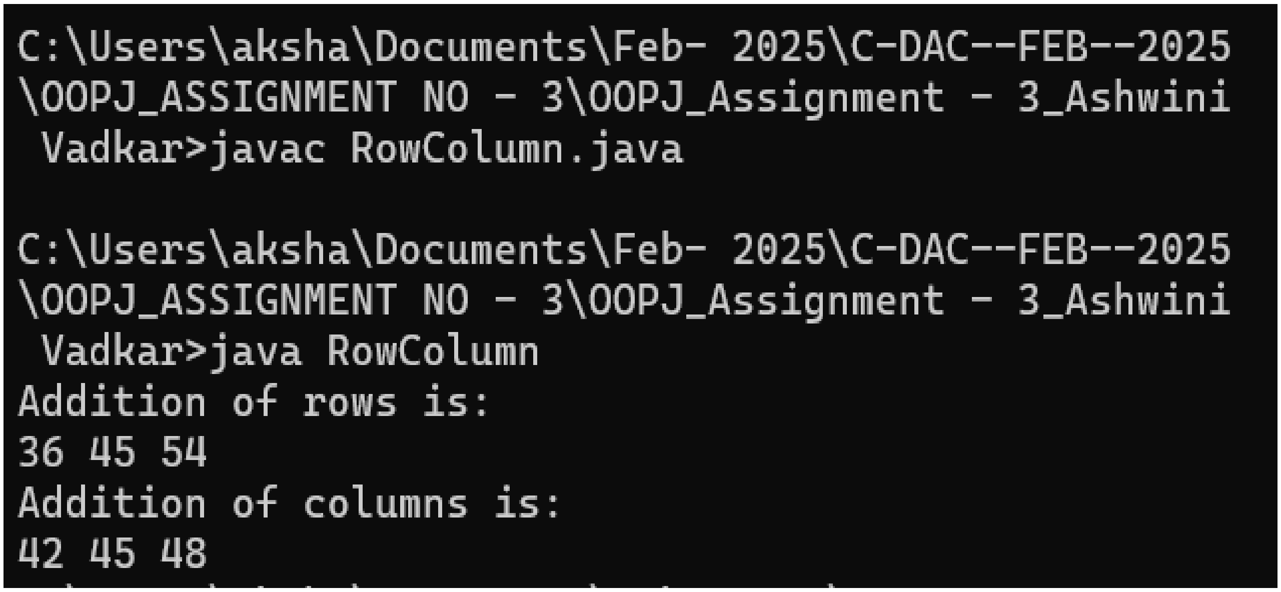
```
C:\Users\aksha\Documents\Feb- 2025\C-DAC--FEB--2025\OOPJ_ASSIGNMENT
NO - 3\OOPJ_Assignment - 3_Ashwini Vadkar>javac MatrixSum.java

C:\Users\aksha\Documents\Feb- 2025\C-DAC--FEB--2025\OOPJ_ASSIGNMENT
NO - 3\OOPJ_Assignment - 3_Ashwini Vadkar>java MatrixSum
12 14 16
18 20 22
24 26 28
```

----

21. Row-wise and Column-wise Sum
    Find the sum of each row and each column of a given matrix.
    Ans :-
    Input :

```
class RowColumn {
    public static void main(String args[]) {
        int a[][] = {{11, 12, 13}, {14, 15, 16}, {17, 18, 19}};
        int row[] = new int[a.length];
        int col[] = new int[a[0].length];

        for (int i = 0; i < 3; i++) {
            for (int j = 0; j < 3; j++) {
                row[i] += a[i][j];
                col[j] += a[i][j];
            }
        }

        System.out.println("Addition of rows is: ");
        for (int i = 0; i < 3; i++) {
            System.out.print(row[i] + " ");
        }
        System.out.println();

        System.out.println("Addition of columns is: ");
        for (int i = 0; i < 3; i++) {
            System.out.print(col[i] + " ");
        }
    }
}
```

Output :



```
C:\Users\aksha\Documents\Feb- 2025\C-DAC--FEB--2025
\OOPJ_ASSIGNMENT NO - 3\OOPJ_Assignment - 3_Ashwini
 Vadkar>javac RowColumn.java

C:\Users\aksha\Documents\Feb- 2025\C-DAC--FEB--2025
\OOPJ_ASSIGNMENT NO - 3\OOPJ_Assignment - 3_Ashwini
 Vadkar>java RowColumn
Addition of rows is:
36 45 54
Addition of columns is:
42 45 48
```

----

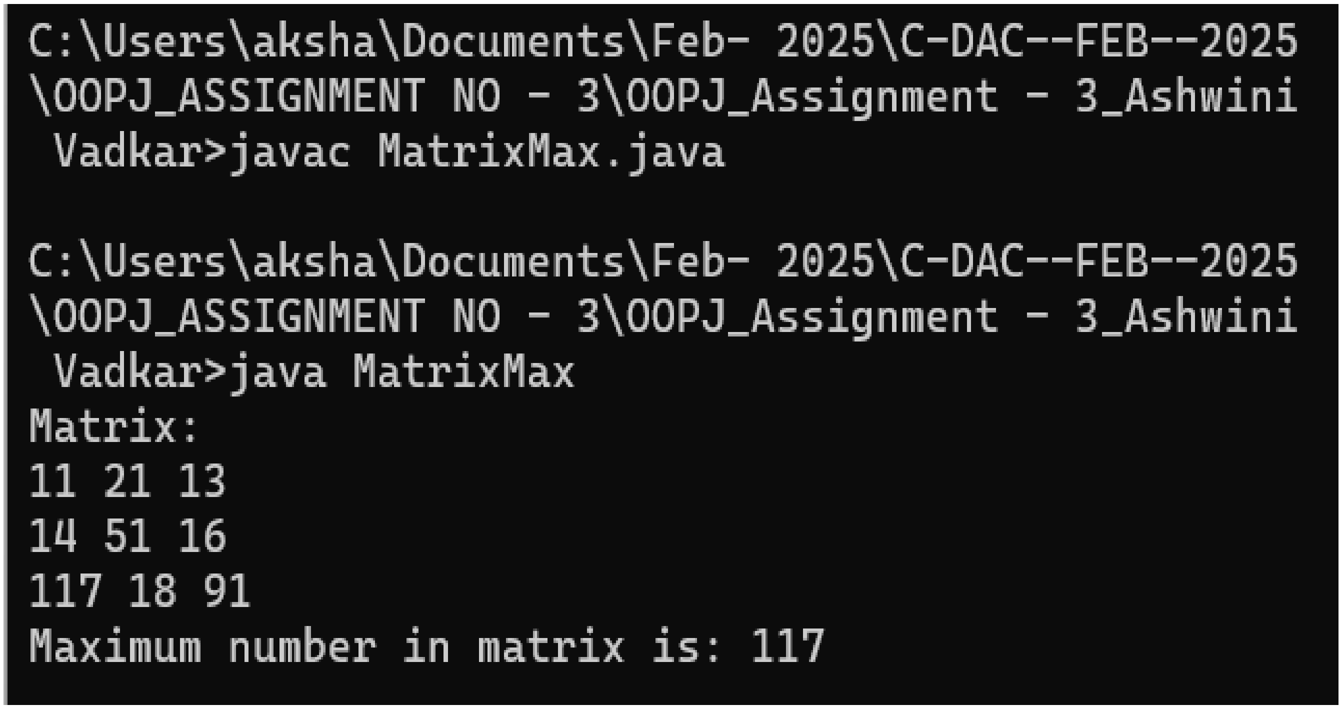
22. Find the Maximum Element in a Matrix
    Find the largest element in a given matrix.

Ans :-

Input :

```
class MatrixMax {
    public static void main(String args[]) {
        int a[][] = {{11, 21, 13}, {14, 51, 16}, {117, 18, 91}};
        int max = 0;

        System.out.println("Matrix:");
        for (int i = 0; i < 3; i++) {
            for (int j = 0; j < 3; j++) {
                System.out.print(a[i][j] + " ");
                if (a[i][j] > max) {
                    max = a[i][j];
                }
            }
            System.out.println();
        }

        System.out.println("Maximum number in matrix is: " + max);
    }
}
```

Output :



```
C:\Users\aksha\Documents\Feb- 2025\C-DAC--FEB--2025
\OOPJ_ASSIGNMENT NO - 3\OOPJ_Assignment - 3_Ashwini
 Vadkar>javac MatrixMax.java

C:\Users\aksha\Documents\Feb- 2025\C-DAC--FEB--2025
\OOPJ_ASSIGNMENT NO - 3\OOPJ_Assignment - 3_Ashwini
 Vadkar>java MatrixMax
Matrix:
11 21 13
14 51 16
117 18 91
Maximum number in matrix is: 117
```

----

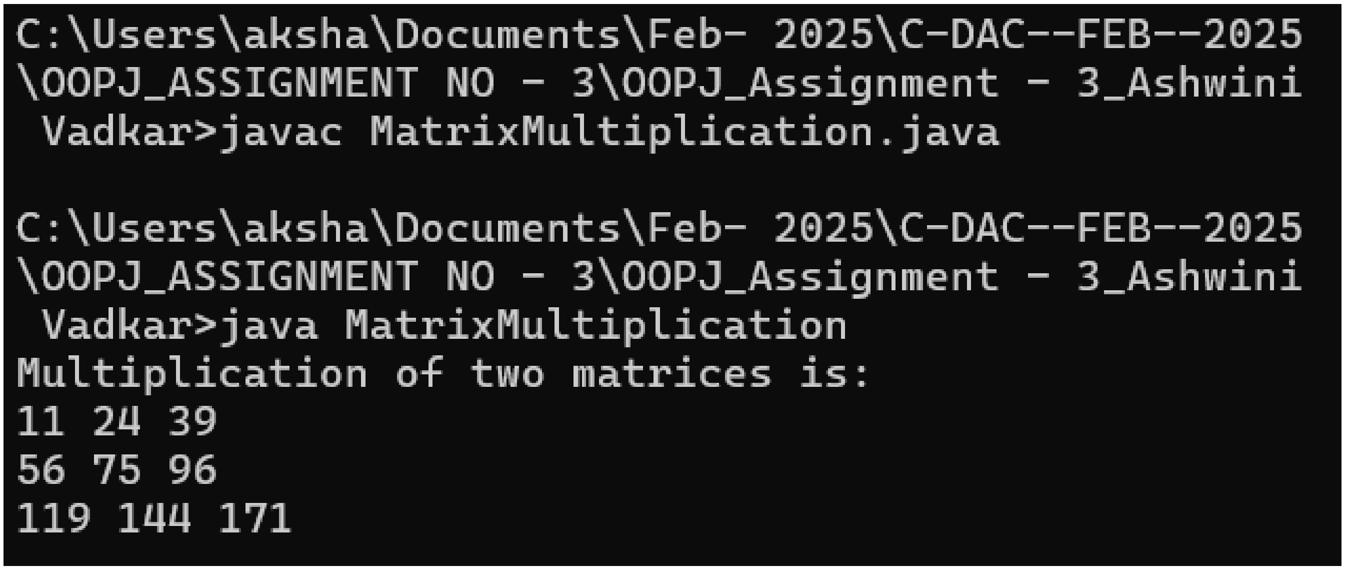
23. Matrix Multiplication
    Multiply two matrices and return the resultant matrix.

Ans :-

Input :

```
class MatrixMultiplication {
    public static void main(String args[]) {
        int a[][] = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}};
        int b[][] = {{11, 12, 13}, {14, 15, 16}, {17, 18, 19}};
        int mult[][] = new int[3][3];

        for (int i = 0; i < 3; i++) {
            for (int j = 0; j < 3; j++) {
                mult[i][j] = a[i][j] * b[i][j];
            }
        }

        System.out.println("Multiplication of two matrices is: ");
        for (int i = 0; i < 3; i++) {
            for (int j = 0; j < 3; j++) {
                System.out.print(mult[i][j] + " ");
            }
            System.out.println();
        }
    }
}
```

Output :



```
C:\Users\aksha\Documents\Feb- 2025\C-DAC--FEB--2025
\OOPJ_ASSIGNMENT NO - 3\OOPJ_Assignment - 3_Ashwini
 Vadkar>javac MatrixMultiplication.java

C:\Users\aksha\Documents\Feb- 2025\C-DAC--FEB--2025
\OOPJ_ASSIGNMENT NO - 3\OOPJ_Assignment - 3_Ashwini
 Vadkar>java MatrixMultiplication
Multiplication of two matrices is:
11 24 39
56 75 96
119 144 171
```

----

24. Rotate a Matrix by 90 Degrees

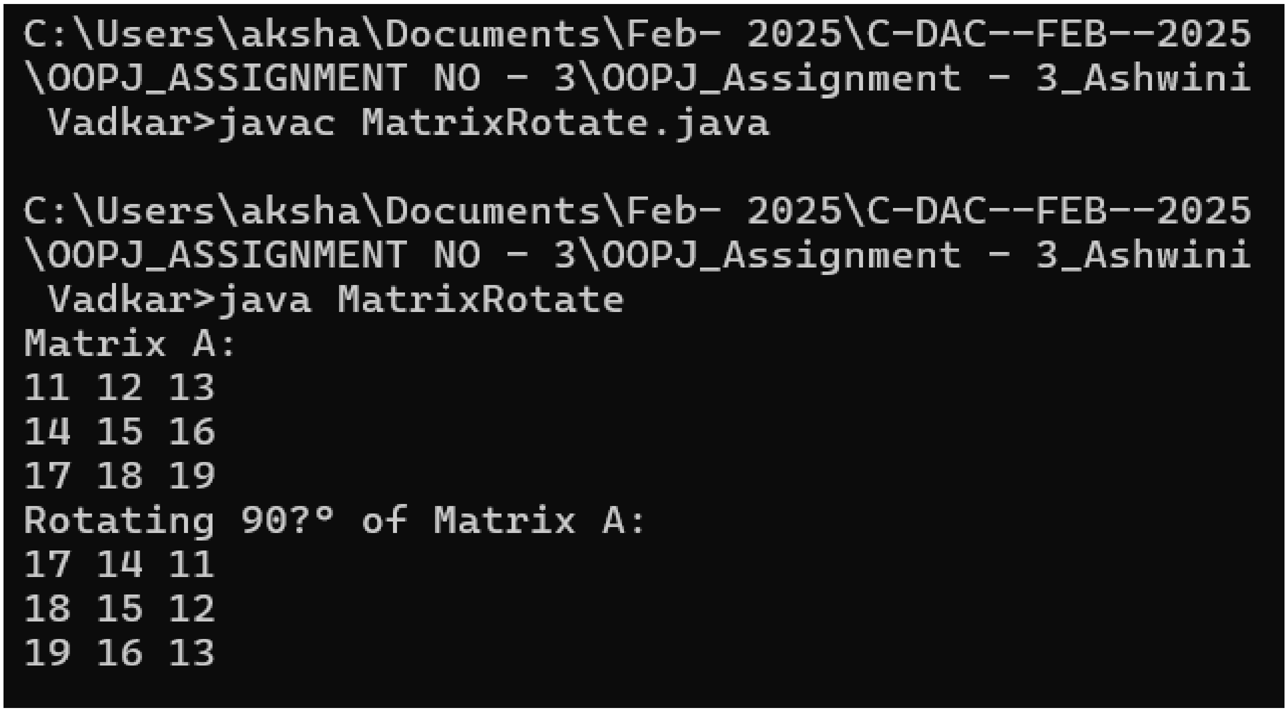
    Rotate a given N x N matrix by 90 degrees clockwise.

Ans :-

Input :

```
class MatrixRotate {
    public static void main(String args[]) {
        int a[][] = {{11, 12, 13}, {14, 15, 16}, {17, 18, 19}};
        int r[][] = new int[3][3];

        System.out.println("Matrix A: ");
        for (int i = 0; i < 3; i++) {
            for (int j = 0; j < 3; j++) {
                System.out.print(a[i][j] + " ");
            }
            System.out.println();
        }
        System.out.println("Rotating 90° of Matrix A: ");
        for (int i = 0; i < 3; i++) {
            for (int j = 0; j < 3; j++) {
                r[j][2 - i] = a[i][j];
            }
        }

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

Output :



```
C:\Users\aksha\Documents\Feb- 2025\C-DAC--FEB--2025
\OOPJ_ASSIGNMENT NO - 3\OOPJ_Assignment - 3_Ashwini
 Vadkar>javac MatrixRotate.java

C:\Users\aksha\Documents\Feb- 2025\C-DAC--FEB--2025
\OOPJ_ASSIGNMENT NO - 3\OOPJ_Assignment - 3_Ashwini
 Vadkar>java MatrixRotate
Matrix A:
11 12 13
14 15 16
17 18 19
Rotating 90?° of Matrix A:
17 14 11
18 15 12
19 16 13
```

----

25. Find the Diagonal Sum

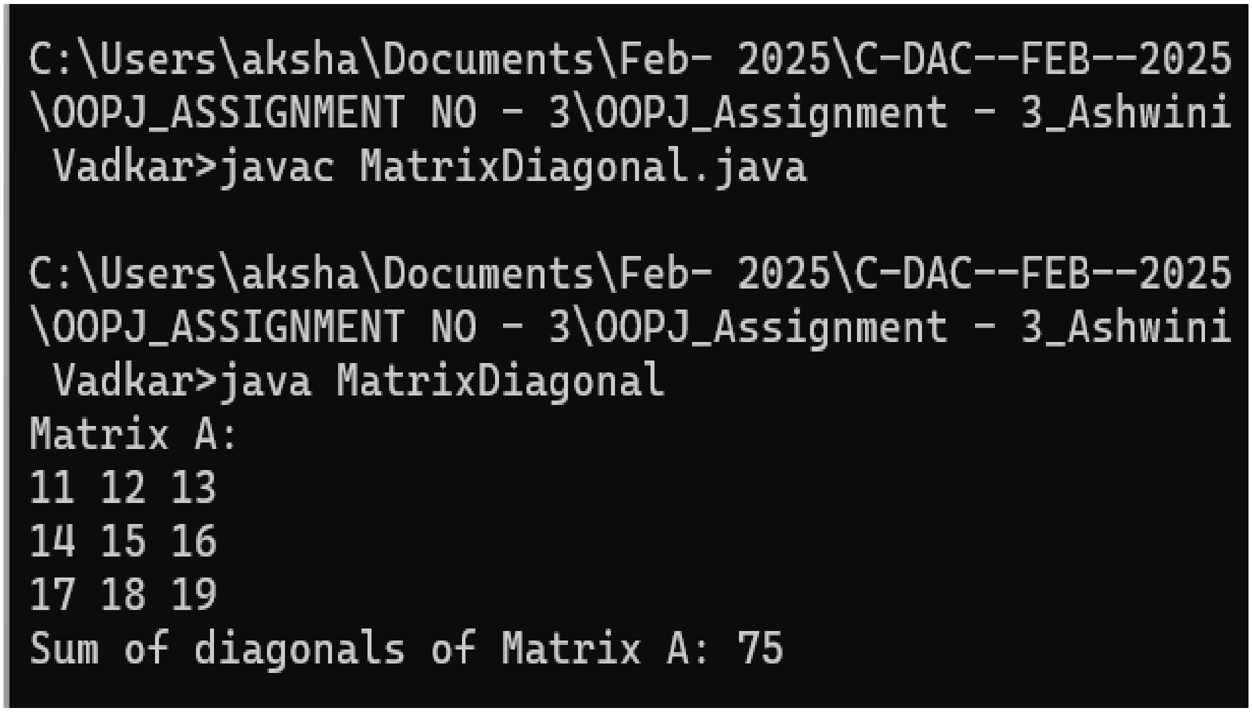
    Compute the sum of both diagonals in a square matrix.

Ans :-

Input :

```
class MatrixDiagonal {
    public static void main(String args[]) {
        int a[][] = {{11, 12, 13}, {14, 15, 16}, {17, 18, 19}};
        int d = 0;

        System.out.println("Matrix A: ");
        for (int i = 0; i < 3; i++) {
            for (int j = 0; j < 3; j++) {
                System.out.print(a[i][j] + " ");
            }
            System.out.println();
        }

        for (int i = 0; i < 3; i++) {
            for (int j = 0; j < 3; j++) {
                if (i == j || i == 0 && j == 2 || i == 2 && j == 0) {
                    d += a[i][j];
                }
            }
        }

        System.out.println("Sum of diagonals of Matrix A: " + d);
    }
}
```

Output :



```
C:\Users\aksha\Documents\Feb- 2025\C-DAC--FEB--2025
\OOPJ_ASSIGNMENT NO - 3\OOPJ_Assignment - 3_Ashwini
 Vadkar>javac MatrixDiagonal.java

C:\Users\aksha\Documents\Feb- 2025\C-DAC--FEB--2025
\OOPJ_ASSIGNMENT NO - 3\OOPJ_Assignment - 3_Ashwini
 Vadkar>java MatrixDiagonal
Matrix A:
11 12 13
14 15 16
17 18 19
Sum of diagonals of Matrix A: 75
```