Array coding question :

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

```
public class Q01{
	public static void main(String[] args){
		int[] arr = {1,2,3,4,25,0,8};

		int min = Integer.MAX_VALUE, max = Integer.MIN_VALUE;
		for(int i=0; i<arr.length; i++){
			if(min>arr[i])
				min = arr[i];
		}

		for(int i=0; i<arr.length; i++){
			if(max<arr[i])
				max = arr[i];
		}

		System.out.println("MIN: " + min + ", MAX: " + max);
	}
}
```

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

```
public class Q02{
	public static void main(String[] args){
		int arr[] = {1,2,3,4,5,6,7,8};
		int low = 0, high = arr.length-1;

		System.out.println("Before: ");
		for(int n : arr){
			System.out.print(n + " ");
		}

		while(low<high){
			arr[high] = arr[high] + arr[low];
			arr[low] = arr[high] - arr[low];
			arr[high] = arr[high] - arr[low];
			low++;
			high--;
		}

		System.out.println("\nAfter: ");
```

```
        for(int n : arr){
            System.out.print(n + " ");
        }
    }
}
```

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

```
public class Q03{
    public static void main(String[] args){
        int arr[] = {1,2,8,3,4,5,6,7,0};
        FindSecLarg(arr);
    }

    public static void FindSecLarg(int[] arr){
        if(arr.length<2){
            System.out.println("Only 1 element");
            return;
        }
        long first = Long.MIN_VALUE, second = Long.MIN_VALUE;
        for(int i=0; i<arr.length; i++){
            if(arr[i]>first){
                second = first;
                first = arr[i];
            } else if(arr[i]>second){
                second = arr[i];
            }
        }
        System.out.println("First largest: " + first +  ", Second largest: " + second);
    }
}
```

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

```
public class Q04{
    public static void main(String[] args){
        int arr[] = {2,3,4,5,6,7,8};
        CountEvenOdd(arr);
    }

    public static void CountEvenOdd(int[] arr){
        int ecount=0, ocount=0;
```

```
                for(int i=0; i<arr.length; i++){
                        if(arr[i]%2==0){
                                ecount++;
                        } else {
                                ocount++;
                        }
                }
                System.out.println("Even Count: " + ecount +  ", Odd Count: " + ocount);


        }
}
```

5. Find Sum and Average

○ **Compute the sum and average of all elements in the array.**

```
public class Q05{
        public static void main(String[] args){
                int arr[] = {2,3,4,5,6,7,8,67};
                for(int a : arr)
                        System.out.print(a + " ");
                System.out.println("\nSum: " + sum((arr)));
                System.out.printf("Average: %.2f",avg((arr)));
        }

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

        public static float avg(int[] arr){
                float sum = (float) sum(arr);
                float avg= sum/arr.length;
                return avg;
        }
}
```

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

```
public class Q06{
	public static void main(String[] args){
		int arr[] = {2,2,3,3,3,4,5,5,6,7,8,67};
		print(arr);
		removeDuplicates(arr);
		System.out.println();
		print(arr);
	}

	public static void removeDuplicates(int[] arr){
		int track=1;
		for(int i=1; i<arr.length; i++){
			if(arr[i]!=arr[i-1])
				arr[track++] = arr[i];
		}
		for(int i=track; i<arr.length; i++){
			if(arr[i]!=arr[i-1])
				arr[track++] = 0;
		}
	}

	public static void print(int[] arr){
		for(int n: arr)
			System.out.print(n + " ");
	}
}
```

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

```
public class Q07{
	public static void main(String[] args){
		int arr1[] = {2,6,7,9,3,8,1,5,4};
		print(arr1);
		rotateright(arr1, 5);
		System.out.println();
		print(arr1);
	}

	public static void rotateright(int[] arr, int r){
		r=r%arr.length;
		reverse(arr, 0, arr.length-1);
		reverse(arr, 0, r-1);
```

```
            reverse(arr, r, arr.length-1);
        }


        public static void reverse(int[] arr, int i, int j){
            while(i<j){
                arr[i] = arr[i] + arr[j];
                arr[j] = arr[i] - arr[j];
                arr[i] = arr[i] - arr[j];
                i++;
                j--;
            }
        }

        public static void print(int[] arr){
            for(int n: arr)
                System.out.print(n + " ");
        }
}
```

8. Merge Two Sorted Arrays

○ **Merge two sorted arrays into a single sorted array without using extra space.**

```
public class Q08{
        public static void main(String[] args){
            int arr1[] = {2,6,8,9,0,0,0,0,0};
            int arr2[] = {1,3,5,7,10};
            mergesort(arr1, arr2);
            print(arr1);
        }

        public static void mergesort(int[] arr1, int[] arr2){
            int m=3, n=arr2.length-1, track=arr1.length-1;
            while(m>=0 && n>=0){
                if(arr1[m]>arr2[n])
                    arr1[track--]=arr1[m--];
                else
                    arr1[track--]=arr2[n--];
            }
            while(m>=0){
                arr1[track--]=arr1[m--];
            }
            while(n>=0){
                arr1[track--]=arr2[n--];
            }
        }
```

```
	public static void print(int[] arr){
		for(int n: arr)
			System.out.print(n + " ");
	}
}
```

9. Find Missing Number in an Array

○ **Given an array of size n-1 containing numbers from 1 to n, find the missing number.**

```
public class Q09{
	public static void main(String[] args){
		int arr1[] = {2,1,4,5,0};
		System.out.println("Missing Number: " + missingNumber(arr1));
	}
	public static int missingNumber(int[] arr){
		int xor1 = 0, xor2 = 0;
		for(int i=0; i<arr.length; i++){
			xor1 = xor1 ^ arr[i];
		}
		for(int i=0; i<=arr.length; i++){
			xor2 = xor2 ^ i;
		}
		return xor1 ^ xor2;
	}
}
```

10. Find Intersection and Union of Two Arrays

○ **Find the intersection and union of two unsorted arrays.**

```
import java.util.HashSet;
public class Q10{
	public static void main(String[] args){
		int arr1[] = {2,1,4,5};
		int arr2[] = {2,7,3,5};
		int res1[] = intersection(arr1,arr2);
		int res2[] = union(arr1,arr2);
		System.out.println("Intersection: ");
		for(int i : res1){
			System.out.print(i + " ");
		}
		System.out.println("\nUnion: ");
		for(int i : res2){
```

```
                System.out.print(i + " ");
            }
        }

        public static int[] intersection(int[] arr1,int[] arr2){
            HashSet<Integer> set = new HashSet<>();
            HashSet<Integer> res = new HashSet<>();
            for(int i=0; i<arr1.length; i++){
                    set.add(arr1[i]);
            }

            for(int i=0; i<arr2.length; i++){
                if(set.contains(arr2[i]))
                    res.add(arr1[i]);
            }

            int i=0;
            int Res[] = new int[res.size()];
            for(int n : res){
                Res[i++] = n;
            }
            return Res;
        }


        public static int[] union(int[] arr1,int[] arr2){
            HashSet<Integer> set = new HashSet<>();
            for(int i=0; i<arr1.length; i++){
                    set.add(arr1[i]);
            }

            for(int i=0; i<arr2.length; i++){
                set.add(arr2[i]);
            }

            int i=0;
            int Res[] = new int[set.size()];
            for(int n : set){
                Res[i++] = n;
            }
            return Res;
        }
}
```

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

```
import java.util.*;
public class Q11{
	public static void main(String[] args){
		Scanner sc = new Scanner(System.in);
		int A[] = {1, 3, -7, 3, 2, 3, 1, -3, -2, -2};
		System.out.println(Arrays.toString(A));
		int n = sc.nextInt();
		ArrayList<Integer> list = new ArrayList<>();
		for(int i=0; i<A.length; i++){
			list.removeAll(list);
			int sum =0;
			for(int j=i; j<A.length; j++){
				sum += A[j];
				list.add(A[j]);
				if(sum==n){
					System.out.println(list.toString());
				}
			}
		}
	}
}
```

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**

```
import java.util.Scanner;
public class Q12{
	public static void main(String ...args){
		int numbers[] = new int[20];
		getInput(numbers);
		System.out.println("Even count: " + countEven(numbers));
		System.out.println("Odd count: " + countOdd(numbers));
		multipleOfThree(numbers);
	}


	public static void getInput(int[] arr){
		Scanner sc = new Scanner(System.in);
		System.out.println("Please enter an input: ");
```

```
		for (int i=0; i<arr.length; i++){
			arr[i]=sc.nextInt();
		}
	}

	public static int countEven(int[] nums){
		int count=0;
		for(int i=0; i<nums.length; i++){
			if(nums[i]%2==0){
				count++;
			}
		}
		return count;
	}

	public static int countOdd(int[] nums){
		int count=0;
		for(int i=0; i<nums.length; i++){
			if(nums[i]%2==1){
				count++;
			}
		}
		return count;
	}

	public static void multipleOfThree(int[] nums){
		for(int i=0; i<nums.length; i++){
			if(nums[i]%3==0 && nums[i] !=0)
				System.out.print(nums[i] + " ");
		}
	}
}
```

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.**

```
import java.util.Scanner;
public class Q13{
	public static void main(String[] args){
		int n = 20;
		Scanner sc = new Scanner(System.in);
		int Chemistry[] = new int[n];
		int Physics[] = new int[n];
		int Maths[] = new int[n];
```

```
		for(int i=0; i<n; i++){
			System.out.print("Chemistry marks of student " + (i+1) + ": ");
			Chemistry[i] = sc.nextInt();
			System.out.print("Physics marks of student " + (i+1) + ": ");
			Physics[i] = sc.nextInt();
			System.out.print("Mathematics marks of student " + (i+1) + ": ");
			Maths[i] = sc.nextInt();
		}
		int count1=0, count2=0;
		for(int i=0; i<n; i++){
			double avgp=((Chemistry[i]+Physics[i]+Maths[i])/300.0)*100;
			if(avgp>=75){
				count1++;
			} else if(avgp<=40){
				count2++;
			}
		}
		System.out.println(" Number of students securing 75% and above in
aggregate: " + count1);
		System.out.println(" Number of students securing 40% and below in
aggregate: " + count2);
	}
}
```

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.

```
import java.util.*;

public class Q14{
	public static void main(String[] args){
		int n = 20;
		Scanner sc = new Scanner(System.in);
		int[] even = new int[n];
		int[] odd = new int[n];
		int arr[] = new int[n];
		int evencount=0, oddcount=0;
		for(int i=0; i<arr.length; i++){
			arr[i]=sc.nextInt();
		}

		for(int i=0; i<arr.length; i++){
```

```
                if(arr[i]%2==0){
                        even[evencount++] = arr[i];
                } else {
                        odd[oddcount++] = arr[i];
                }
        }
        System.out.println("Odd Numbers: ");
        for(int i=0;i<oddcount; i++){
                System.out.print(odd[i] + " ");
        }

        System.out.println("\nEven Numbers: ");
        for(int i=0;i<evencount; i++){
                System.out.print(even[i] + " ");
        }
    }
}
```

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]

```
import java.util.*;
public class Q15{
    public static void main(String[] args){

        int A[] = {1, 3, -7, -3, 2, 4, 1, -3, -2, 5};
        System.out.println(Arrays.toString(A));

        ArrayList<Integer> list = new ArrayList<>();
        for(int i=0; i<A.length; i++){
                list.removeAll(list);
                int sum =0;
                for(int j=i; j<A.length; j++){
                        sum += A[j];
                        list.add(A[j]);
                        if(sum==0){
```

```
                                        System.out.println(list.toString());
                                }
                        }
                }
        }
}
```

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]

```
import java.util.Arrays;
public class Q16{
        public static void main(String[] args){
                int[] A = {1, 5, 6, 7, 8, 10};
                int[] B = { 2, 4, 9 };
                int[] res = new int[A.length];
                int i = 0, j = 0, track = 0;
                while(track<res.length){
                        if(A[i]<B[j]){
                                res[track++] = A[i++];
                        } else {
                                res[track++] = B[j++];
                        }
                }
                track = 0;
                while(j<B.length){
                        B[track++] = B[j++];
                }
                while(i<A.length){
                        B[track++] = A[i++];
                }
                System.out.println(Arrays.toString(res));
                System.out.println(Arrays.toString(B));
        }
}
```

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

```
public class Q17{
	public static void main(String args[]){
		int nums[] = { 2, 3, 5, 7, -7, 5, 8, -5 };
		int first=Integer.MIN_VALUE, second=Integer.MIN_VALUE;
		int firstmin=Integer.MAX_VALUE, secondmin=Integer.MAX_VALUE;
		for(int i=0; i<nums.length; i++){
			if(nums[i]>first){
				second = first;
				first = nums[i];
			} else if(nums[i]>second){
				second = nums[i];
			}

			if(nums[i]<firstmin){
				secondmin = firstmin;
				firstmin = nums[i];
			} else if(nums[i]<secondmin){
				secondmin = nums[i];
			}
		}
		int prod2=firstmin * secondmin;
		int prod1=first * second;
		int maxprod = Math.max(prod1,prod2);

		if(prod1==maxprod)
			System.out.println("Pair is (" + first + "," + second  + "), Maximum
Product: " + maxprod);
		else
			System.out.println("Pair is (" + firstmin + "," + secondmin  + "),
Maximum Product: " + maxprod);
	}
}
```

18. Print a Matrix

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

```
public class Q18{
	public static void main(String args[]){
		int[][] matrix = {{1,2,3},{4,5,6},{7,8,9}};
		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();
		}
	}
}
```

19. Transpose of a Matrix

○ **Given a matrix, return its transpose (swap rows and columns).**

```
public class Q19{
	public static void main(String args[]){
		int[][] matrix1 = {{1,2,3},{4,5,6},{7,8,9}};
		int[][] matrix2 = {{11,21,31},{14,15,16},{27,28,92}};
		int result[][] = new int[matrix1.length][matrix2[0].length];
		for(int i=0; i<matrix1.length; i++){
			for(int j=0; j<matrix2[i].length; j++){
				result[i][j] =	matrix1[i][j] + matrix2[i][j];
			}
		}

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

20. Sum of Two Matrices

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

```
public class Q20{
	public static void main(String args[]){
		int[][] matrix1 = {{1,2,3},{4,5,6},{7,8,9}};
		int[][] matrix2 = {{11,21,31},{14,15,16},{27,28,92}};
		int result[][] = new int[matrix1.length][matrix2[0].length];
		for(int i=0; i<matrix1.length; i++){
			for(int j=0; j<matrix2[i].length; j++){
				result[i][j] =	matrix1[i][j] + matrix2[i][j];
			}
		}

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

21. Row-wise and Column-wise Sum

○ **Find the sum of each row and each column of a given matrix.**

```
public class Q21{
	public static void main(String args[]){
		int rowsum = 0, colsum = 0;
		int[][] matrix = {{1,2,3},{4,5,6},{7,8,9}};
		for(int i=0; i<matrix.length; i++){
			rowsum = 0;
			colsum = 0;
			for(int j=0; j<matrix[i].length; j++){
				rowsum += matrix[i][j];
				colsum += matrix[j][i];
			}
			System.out.println("Row Sum: " + rowsum + "\tColumn Sum: " +
colsum);
		}
	}
}
```

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

```
public class Q22{
	public static void main(String args[]){
		int max= Integer.MIN_VALUE;
		int[][] matrix = {{11,21,32},{42,53,61},{17,128,32}};
		for(int i=0; i<matrix.length; i++){
			for(int j=0; j<matrix[i].length; j++){
				max = (matrix[i][j]>max)? matrix[i][j] : max;
			}
		}
		System.out.println("Largest element: " + max);
	}
}
```

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

```
public class Q23{
	public static void main(String args[]){
		int[][] matrix1 = {{1,2,3},{1,2,3},{1,2,3}};
		int[][] matrix2 = {{1,2,3},{1,2,3},{1,2,3}};
		int[][] res = new int[matrix1.length][matrix1[0].length];
		for(int i=0; i<matrix1.length; i++){
			for(int j=0; j<matrix1.length; j++){
				for(int k=0; k<matrix1.length; k++){
					res[i][j] = res[i][j] + (matrix1[i][k] + matrix2[k][j]);
				}
			}
		}
		for(int i=0; i<res.length; i++){
			for(int j=0; j<res.length; j++){
				System.out.print(res[i][j] + " ");
			}
			System.out.println();
		}
	}
}
```

24. Rotate a Matrix by 90 Degrees

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

```
public class Q24{
	public static void main(String[] args){
		int matrix[][] = {{1,2,3},{4,5,6},{7,8,9}};
		int res[][] = new int[matrix.length][matrix[0].length];
		print(matrix);
		transpose(matrix);
		for(int i=0; i<matrix.length; i++)
			reverse(matrix[i]);
		System.out.println("-----");
		print(matrix);
	}

	public static void transpose(int[][] matrix){
		for(int i=0; i<matrix.length; i++){
			for(int j=i+1; j<matrix[i].length; j++){
				int temp = matrix[i][j];
				matrix[i][j] = matrix[j][i];
				matrix[j][i] = temp;
			}
		}
	}

	public static void reverse(int[] row){
		int low=0, high=row.length-1;
		while(low<high){
			row[high] = row[high] + row[low];
			row[low] = row[high] - row[low];
			row[high] = row[high] - row[low];
			low++;
			high--;
		}
	}

	public static void print(int[][] input){
		for(int i=0; i<input.length; i++){
			for(int j=0; j<input[0].length; j++){
				System.out.print(input[i][j]+ " ");
			}
			System.out.println();
		}
	}
}
```

25. Find the Diagonal Sum

○ **Compute the sum of both diagonals in a square matrix.**

```
public class Q25{
	public static void main(String args[]){
		int diagonalsum=0;
		int[][] matrix = {{11,21,32},{42,2,61},{17,128,3}};
		for(int i=0; i<matrix.length; i++){
			diagonalsum += matrix[i][i];
		}
		System.out.println("Diagonal Sum: " + diagonalsum);
	}
}
```