CDAC Mumbai

Array coding question :

1. Find the Largest and Smallest Element

○ Given an array, find the smallest and largest elements in it.

```
class Q1
{
    public static void main(String[] args)
    {
        int[] arr = {5,10,12,7,9};
        int smallest = arr[0];
        int largest = arr[0];
        for (int i = 1; i < arr.length; i++)
        {
            if (arr[i] < smallest)
            {
                smallest = arr[i];
             }
             if (arr[i] > largest)
            {
                 largest = arr[i];
             }
         }

         System.out.println("Smallest: " + smallest);
         System.out.println("largest: " + largest);
    }
}
```

---

2. Reverse an Array

○ Reverse the given array in place.

```
class Q2
{
    public static void main(String[] args )
    {
        int[] arr = {22,31,30,1,60,100 ,200};
        int size = arr.length;

        int start = 0;
        int end = size-1;
        while(start<end)
        {
            int temp = arr[start];
            arr[start] = arr[end];
            arr[end] = temp;
            start++;
            end--;
        }

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


    }
}
```

---

3. Find the Second Largest Element

○ Find the second-largest element in the given array.

```
class Q3
{
    public static void main(String[] args)
    {

        int[] arr = {22,31,30,1,60,100,3000};
        int max1 = arr[0];
        int max2 = arr[1];

        if(arr[0] > arr[1])
        {
            max1 = arr[0];
            max2= arr[1];
        }else
        {
            max1 = arr[1];
            max2= arr[0];
        }

        for ( int i =2 ;i<arr.length;i++)
        {
            if(arr[i] > max1 )
            {
                max2 =max1;
                max1= arr[i];
            }
            else if(arr[i] > max2 )
            {
                max2 = arr[i];
            }

        }
        System.out.println(max2);
    }
}
```

4. Count Even and Odd Numbers

○ **Count the number of even and odd numbers in an array.**

```
 class Q4
 {
     public static void main(String[] args )
     {
         int[] arr = {22,31,30,1,60,100,3000};
         int evenCount = 0;
         int oddCount = 0;

         for( int x : arr )
         {
             if( x % 2==0)
             {
                 evenCount++;
             }
             else
             {
                 oddCount++;
             }
         }

         System.out.println("total even number in given array "+evenCount );
         System.out.println("total odd number in given array "+oddCount );

     }
```

```
}
```

5. Find Sum and Average

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

```
class Q5
{
    public static void main(String[] args)
    {
        int[] arr = {22,31,30,1,60,100,3000};

        int sum=0;
        for ( int x : arr )
        {
            sum+=x;
        }
        System.out.println("sum of all the elements in the array is: "+sum);
        System.out.println("Average of all the elements in the array is: "+sum/arr.length);
    }
}
```

6. Remove Duplicates from a Sorted Array

○ Remove duplicate elements from a sorted array without using extra space.

```
//Remove Duplicates from a Sorted Array

import java.util.Scanner;
import java.util.HashSet;

public class Q6 {
    public static void main(String[] args) {


        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the number of elements: ");  //howmany elements
        int n = scanner.nextInt();

        int[] arr = new int[n];
        System.out.println("Enter the sorted elements of the array:"); //array elements
        for (int i = 0; i < n; i++) {
            arr[i] = scanner.nextInt();   //take input of array each element
        }

        // Remove duplicates

        HashSet<Integer> set = new HashSet<>();

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

        System.out.println("Array after removing duplicates:");  // printing array here aftre removing duplicate element

        for (int num : set) {
            System.out.print(num + " ");
        }
    }
}
```

---

7. Rotate an Array

○ **Rotate the array to the right by k positions.**

```
//Rotate an array

import java.util.Scanner;

public class Q7{
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the number of elements: ");
        int n = scanner.nextInt();

        int[] arr = new int[n];
        System.out.println("Enter the elements of the array:");
        for (int i = 0; i < n; i++) {
            arr[i] = scanner.nextInt();
        }

        System.out.print("Enter the number of positions to rotate: ");
        int k = scanner.nextInt();
        // Rotate the array

        k = k % n;

        int[] rotated = new int[n];
        for (int i = 0; i < n; i++) {
            rotated[(i + k) % n] = arr[i];
        }

        System.out.println("Rotated Array:");
        for (int i = 0; i < n; i++) {
            System.out.print(rotated[i] + " ");
        }
    }
}
```

8. Merge Two Sorted Arrays

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

```
import java.util.Arrays;
public class Main
{
            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.arraycopy(a, 0, c, 0, a.length);
        System.arraycopy(b, 0, c, a.length, b.length);
        Arrays.sort(c);
            for(int x:c){
            System.out.print(x);
}
}
}
```

9. Find Missing Number in an Array

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

```
//Find Missing number in an array

import java.util.Scanner;

public class Q9 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the number of elements: "); //howmany elements
        int n = scanner.nextInt();

        int[] arr = new int[n - 1];
        System.out.println("Enter the elements of the array (1 to " + n + " excluding one number):");
        for (int i = 0; i < n - 1; i++) {
            arr[i] = scanner.nextInt();
        }

        int totalSum = n * (n + 1) / 2;  //here we are counting and did some of number of elements
we entered before
        int aSum = 0;
        for (int num : arr) {
            aSum += num;
        }

        int missingNum = totalSum - aSum;

        System.out.println("Missing Number: " + missingNum);
    }
}
```

---

10. Find Intersection and Union of Two Arrays

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

```
//Find Intersection and Union of Two Arrays

import java.util.*;

public class Q10 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the number of elements for the first array: ");
        int n1 = scanner.nextInt();
        int[] arr1 = new int[n1];
        System.out.println("Enter the elements of the first array:");
        for (int i = 0; i < n1; i++) {
            arr1[i] = scanner.nextInt();
        }

        System.out.print("Enter the number of elements for the second array: ");
        int n2 = scanner.nextInt();
        int[] arr2 = new int[n2];
        System.out.println("Enter the elements of the second array:");
        for (int i = 0; i < n2; i++) {
            arr2[i] = scanner.nextInt();
        }

        Set<Integer> set1 = new HashSet<>();
        for (int num : arr1) {
            set1.add(num);
        }

        Set<Integer> set2 = new HashSet<>();
        for (int num : arr2) {
            set2.add(num);
        }

        Set<Integer> intersection = new HashSet<>(set1);
        intersection.retainAll(set2);

        Set<Integer> union = new HashSet<>(set1);
        union.addAll(set2);

        System.out.println("Intersection: " + intersection);
        System.out.println("Union: " + union);
    }
}
```

---

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.Arrays;
Import java.util.Scanner;
public class Main
{
    static void sub(int[]a){
       Scanner sc  =new Scanner(System.in);
            int S = sc.nextInt();
        for(int i =0;i<a.length;i++){
                            int sum=0;
                            for(int j=i;j<a.length;j++){
                                sum+=a[j];
                                if(sum==S){
                                    int[] subarray = Arrays.copyOfRange(a, i, j+1);
                                    for(int x:subarray){
                                        System.out.print(x+" ");
                                    }System.out.println();
                                }
                            }
                        }
    }
            public static void main(String[] args) {
                        int[]a ={ 1, 3, -7, 3, 2, 3, 1, -3, -2, -2};
                        int[]b = { 1, 2, -3, 4, 5, 6 };
                        int[]c =  { 1, 2, -2, 3, 4, 5, 6 };
                        Main.sub(a);
                        Main.sub(b);
                        Main.sub(c);


            }
}
```

---

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;
class Q12
{
            public static void main(String[] args )
            {
                        int[] arr = new int[20];
                        Scanner sc = new Scanner(System.in);

                        int evenCount=0;
                        int oddCount = 0;
                        int multipleOf3 = 0;
                        for( int i = 0; i<arr.length;i++ )
                        {
```

```
            arr[i] = sc.nextInt();
            if ( arr[i] %2 ==0)
            {
                evenCount++;
                if ( arr[i] % 3 ==0)
                {
                    multipleOf3++;
                }
            }
            else
            {
                oddCount++;
                if ( arr[i] % 3 ==0)
                {
                    multipleOf3++;
                }
            }

        }
        System.out.println(evenCount);
        System.out.println(oddCount);
        System.out.println(multipleOf3);


    }
}
```

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;

class Q13 {
    public static void main(String[] args) {
        int[] physicsMarks = new int[20];
        int[] chemistryMarks = new int[20];
        int[] mathsMarks = new int[20];

        Scanner sc = new Scanner(System.in);
        for (int i = 0; i < 5; i++) {
            System.out.println("Enter marks for Student " + (i + 1) + ":");

            System.out.print("Physics: ");
            physicsMarks[i] = sc.nextInt();

            System.out.print("Chemistry: ");
            chemistryMarks[i] = sc.nextInt();

            System.out.print("Maths: ");
            mathsMarks[i] = sc.nextInt();
        }
                        int greater75 = 0;
                        int lesser40 = 0;
                        for (int i = 0; i < 5; i++) {
            System.out.println("Student " + (i + 1) + ": Physics = " + physicsMarks[i]
                                + ", Chemistry = " + chemistryMarks[i]
                                + ", Maths = " + mathsMarks[i]);
                                    double percentage =
(double)(physicsMarks[i]+chemistryMarks[i]+mathsMarks[i])/3;
                                    if (percentage>=75)
                                    {
                                                greater75++;
                                    }
                                    if (percentage<=40)
                                    {
                                                lesser40++;
                                    }
        }

                        System.out.println("Students securing with 75% and above is: "+greater75);
                        System.out.println("Students securing with 40% and less is: "+lesser40);
    }
}
```

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.Scanner;
class Q14
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        int[] arr= new int[20];
        int[] arreven= new int[20];
        int[] arrodd= new int[20];
        int x = 0;
        int y = 0;

        for(int i = 0; i<arr.length;i++)
        {
            arr[i] = sc.nextInt();
        }
        System.out.println("---------");
        for(int i = 0; i<arr.length;i++)
        {
            System.out.print(arr[i]+" ");
            if( arr[i]%2==0)
            {
                arreven[x] = arr[i];
                x++;
            }
            else
            {
                arrodd[y] = arr[i];
                y++;
            }
        }

        System.out.println("even arry");
        for(int i = 0 ;i<x;i++)
        {
            System.out.println(arreven[i]);
        }
```

```
            System.out.println("0dd arry");
            for(int i = 0; i<y;i++)
            {
                  System.out.println(arrodd[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.Arrays;
public class Main
{
  static void sub(int[]a){
    for(int i =0;i<a.length;i++){
              int sum=0;
              for(int j=i;j<a.length;j++){
                sum+=a[j];
                if(sum==0){
                  int[] subarray = Arrays.copyOfRange(a, i, j+1);
                  for(int x:subarray){
                    System.out.print(x+" ");
                  }System.out.println();
                }
              }
            }
  }
      public static void main(String[] args) {
            int[]a ={ 1, 3, -7, 3, 2, 3, 1, -3, -2, -2};
            int[]b = { 1, 2, -3, 4, 5, 6 };
```

```
        int[]c =  { 1, 2, -2, 3, 4, 5, 6 };
        Main.sub(a);
        Main.sub(b);
        Main.sub(c);


    }
}
```

---

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 Main
{
    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.arraycopy(a, 0, c, 0, a.length);
   System.arraycopy(b, 0, c, a.length, b.length);
   Arrays.sort(c);
   System.arraycopy(c,0,a,0,a.length);
   System.arraycopy(c,a.length,b,0,b.length);
        for(int i=0;i<a.length;i++){
         System.out.print(a[i]+" ");
        }
        System.out.println();
        for(int i=0;i<b.length;i++){
         System.out.print(b[i]+" ");
        }


    }
}
```

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

```
import java.util.Arrays;
public class Main
{
        public static void main(String[] args) {
                int[]a ={2, 3, 5, 7, -7, 5, 8, -5};
                Arrays.sort(a);
                System.out.println(a[a.length-1]*a[a.length-2]);
        }
}
```

18. Print a Matrix
○ Given an m x n matrix, print all its elements row-wise.

```
class Q18
{
        public static void main( String[] args)
        {
                int[][] arr = new int[3][3];
                arr[0] = new int[] {1, 2, 3};
                arr[1] = new int[] {4, 5, 6};
                 arr[2] = new int[] {7, 8, 9};

                for (int x[] : arr)
                {
                        for (int x1 : x)
                        {
                                System.out.print(x1+" ");
                        }
                        System.out.println();
                }
```

```
            }
}
```

---

19. Transpose of a Matrix
○ Given a matrix, return its transpose (swap rows and columns).

```
class Q19
{

    public static void main(String[] args)
            {
                        int[][] arr = new int[3][3];
                        arr[0] = new int[] {1, 2, 3};
                        arr[1] = new int[] {4, 5, 6};
                         arr[2] = new int[] {7, 8, 9};


                         System.out.println("Original Matrix:");
                         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();
                         }

                        // Transpose the matrix
                        int[][] transpose = new int[3][3];

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


                        System.out.println("\nTransposed Matrix:");
                         for (int i = 0; i < transpose.length; i++)
                        {
```

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

}
```

---

20. Sum of Two Matrices
○ Given two matrices of the same size, compute their sum.

```
class Q20
{
    public static void main(String[] args )
    {
        int[][] arr1 = new int[3][3];
         arr1[0] = new int[] {1, 2, 3};
        arr1[1] = new int[] {4, 5, 6};
         arr1[2] = new int[] {7, 8, 9};

        int[][] arr2 = new int[3][3];
        arr2[0] = new int[] {11, 22, 33};
        arr2[1] = new int[] {44, 55, 66};
        arr2[2] = new int[] {77, 88, 99};

        int[][] result = new int[3][3];

        for (int i = 0; i < result.length; i++)
        {
             for (int j = 0; j < result[i].length; j++)
            {
                result[i][j] = arr1[i][j] + arr2[i][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.

```
import java.util.*;
public class Main
{
        public static void main(String[] args) {
           Scanner sc = new Scanner(System.in);
                 System.out.println("Enter the dimension of a matrix ");
           int M=sc.nextInt();
           int N=sc.nextInt();
           int[][]A=new int [M][N];
           int max=0;
           int[]row = new int[M];
           int col[] = new int[N];

           for(int i=0;i<M;i++){
              for(int j =0;j<N;j++){
                 System.out.println("Enter the value of A["+i+"]["+j+"]");
                 A[i][j]=sc.nextInt();
                 row[i]+=A[i][j];
              }
           }
           for(int i=0;i<N;i++){
              for(int j =0;j<M;j++){

                 col[i]+=A[j][i];
              }
           }
           System.out.println("Sum of rows");
           for(int x:row){
              System.out.print(x+" ");
           }
           System.out.println();
           System.out.println("Sum of columns");
           for(int x:col){
              System.out.print(x+" ");
           }

        }
}
```

---

22. Find the Maximum Element in a Matrix

○ Find the largest element in a given matrix.

```
import java.util.*;
public class Main
{
        public static void main(String[] args) {
           Scanner sc = new Scanner(System.in);
                System.out.println("Enter the dimension of a matrix ");
           int M=sc.nextInt();
           int N=sc.nextInt();
           int[][]A=new int [M][N];
           int max=0;
           for(int i=0;i<M;i++){
              for(int j =0;j<N;j++){
                System.out.println("Enter the value of A["+i+"]["+j+"]");
                A[i][j]=sc.nextInt();
                max=(A[i][j]>max)?A[i][j]:max;
              }
           }
           System.out.println("Maximum element of the matrix is : "+max);
        }
}
```

---

23. Matrix Multiplication

○ **Multiply two matrices and return the resultant matrix**.

```
import java.util.Scanner;
public class Main
{
        public static void main(String[] args) {
                Scanner sc = new Scanner(System.in);
                System.out.println("Print dimension of matrix A(MxN)");
                int am=sc.nextInt();
                int an=sc.nextInt();
                System.out.println("Print dimension of matrix B(NxO)");
                int bn=sc.nextInt();
                int bo=sc.nextInt();
                int[][]A=new int[am][an];
                int[][]B=new int[bn][bo];
                for(int i=0;i<am;i++){
                   for(int j=0;j<an;j++){
                      System.out.println("Enter element of A["+i+"]["+j+"]");
                      A[i][j]=sc.nextInt();
                   }
                }
                for(int i=0;i<bn;i++){
```

```
            for(int j=0;j<bo;j++){
                System.out.println("Enter element of B["+i+"]["+j+"]");
                B[i][j]=sc.nextInt();
            }
        }
        /*for(int q[]:A){
            for(int w:q){
                System.out.print(w+" ");
            }System.out.println();
        }
        for(int q[]:B){
            for(int w:q){
                System.out.print(w+" ");
            }System.out.println();
        }*/
        int[][]R=new int[am][bo];
        for(int i=0;i<am;i++){
            for(int j=0;j<bo;j++){
                for(int k =0;k<an;k++){
                    R[i][j]+=A[i][k]*B[k][j];
                }
            }
        }
        for(int q[]:R){
            for(int w:q){
                System.out.print(w+" ");
            }System.out.println();
        }

    }
}
```

24. Rotate a Matrix by 90 Degrees

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

```
public class Main
{
    public static void main(String[] args) {
        int [][]arr = new int[3][];
        arr[0]= new int[] {1,2,3};
        arr[1]= new int[] {4,5,6};
        arr[2]= new int [] {7,8,9};
        for(int a[]:arr){
            for(int b:a){
```

```
                        System.out.print(b+" ");
                    }System.out.println();
                }
                System.out.println("After rotation of 90deg to th right ");
                for(int i =0; i<arr.length;i++){
                    int count=2;
                    for (int j=0; j<arr[i].length;j++){
                        System.out.print(arr[count][i]+" ");
                        count--;
                    }System.out.println();

                }
        }
}
```

---

25. Find the Diagonal Sum

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

```
public class Main
{
        public static void main(String[] args) {
                int[][] arr = new int[3][3];
                arr[0]=new int [] {2,2,2};
                arr[1]=new int [] {3,3,3};
                arr[2]=new int [] {4,4,4};
                for(int x[] : arr){
                    for(int y:x){
                        System.out.print(y+" ");
                    }
                    System.out.println();
                }
                int diag1=0, diag2=0;
                for(int a=0;a<3;a++){
                    for(int b=0;b<3;b++){
                        if(a==b){
                            diag1 += arr[a][b];
                        }
                        if(a+b==2){
                            diag2 +=  arr[a][b];
                        }
                    }
                }
                System.out.println("Sum of 1st diagonal : "+diag1);
                System.out.println("Sum of 2nd diagonal : "+diag2);}}
```