JAVA Logic Placement Preparation Test 4

PRN : 030

Q.1) Accept data in sorted order having duplicate value. You needPrint unique sorted array to print unique array using single loop . Unique sorted array using 1 loop 1

Input 1 2 2 2 5

output 1 2 5

=>

```
package java_logic;

import java.util.Arrays;

public class Unique_Array_1 {

        public static void main(String[] args) {

                int[] arr = {1,1,2,2,5};

                uniqueArray(arr);

        }

        static void uniqueArray(int[] arr) {

                Arrays.sort(arr);

                int n = arr.length;

                int[] res = new int[n];

                int r=0;

                int prev = Integer.MIN_VALUE;

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

                        if(arr[i] != prev) {

                                res[r++]=arr[i];

                                prev=arr[i];       }           }

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

                        System.out.print(res[i]+ " ");      }

        }

}
```

```
}
```

Q.2) To find the maximum sum of all subarrays of size K: Given an array of integers of size ‘n’, Our aim is to calculate the maximum sum of ‘k’ consecutive elements in the array.

Input : arr[] = {100, 200, 300, 400}, k = 2

Output : 700

=>

```
package java_logic;

public class MaxSum_SubArray {

	public static void main(String[] args) {

		int[] arr = {100,200,300,400};

		int k=2;

		System.out.println(maxSum_SubArray(arr,k));

	}

	static int maxSum_SubArray(int[] arr, int k) {

		int n = arr.length;

		int max_sum=0;

		int window_sum=0;

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

			max_sum+=arr[i];

			window_sum=max_sum;

		}

		for(int i=k;i<n;i++) {

			window_sum+=arr[i]-arr[i-k];

			max_sum=Math.max(max_sum, window_sum);

		}

		return max_sum;

	}

}
```