/*Q.1) Print unique sorted array \Accept data in sorted order having duplicate value. You need to print unique array using single loop . Unique sorted array using 1 loop

Input 1 1 2 2 2 5 output 1 2 5

*/

```
import java.util.Arrays;
public class SET1Q1 {
   public static void printuniquearr(int arr[]){
      Arrays.sort(arr);
      int privious =Integer.MIN_VALUE;
      for(int i=0;i<arr.length;i++){
         if(arr[i]!=privious){
            System.out.print(arr[i]+" ");
            privious=arr[i];
         }
      }
   }
      public static void main(String[] args)
   {
      int arr[]={1,1,2,2,2,3,5};
      printuniquearr(arr);
   }
}
```

```
/*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*/
public class SET1Q2 {

   public static int maxSum(int[] arr, int k) {
      int n = arr.length;

       if (n < k) {
          return -1;
       }

       int windowSum = 0;
       for (int i = 0; i < k; i++) {
          windowSum += arr[i];
       }
       int maxSum = windowSum;



       for (int i = k; i < n; i++) {
          windowSum = windowSum + arr[i] - arr[i - k];
          maxSum = Math.max(maxSum, windowSum);
       }
       return maxSum;
```

```
    }
    public static void main(String[] args) {
        int[] arr = {100, 200, 300, 400};
        int k = 2;
        System.out.println(maxSum(arr, k));
    }
}
```