//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

//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 Day5 {

	public static void main(String[] args) {
		// TODO Auto-generated method stub
		int nums[] = {1,1,2,2,5,5};
		Q1(nums);
		System.out.println(Q2(new int[]{100, 200, 300, 400}, 2));
	}
	public static void Q1(int[] nums) {
		int prev = Integer.MIN_VALUE;
		for(int i = 0 ; i<nums.length ; i++) {
			if(nums[i]>prev) {
				System.out.print(nums[i]+" ");
				prev = nums[i];
			}
		}
```

```
	}
	public static int Q2(int[] nums , int k) {
		if(nums.length < k) {
			System.out.println("Invalid Input");
			return -1;
		}
		int maxSum = 0;
		for(int i = 0 ; i < k ; i++) {
			maxSum+=nums[i];
		}
		int currSum = maxSum;
		for(int i = k ; i < nums.length ; i++) {
			currSum = currSum + nums[i] - nums[i-k];
			maxSum = Math.max(maxSum, currSum);
		}
		return maxSum;
	}
}
```