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

```
package logic;

public class uniqueArray {

public static void printArray(int[] arr) {
int[] newArray = new int[arr.length];
int a = 0;
int b = Integer.MIN_VALUE;
System.out.println("Unique Array: ");
for (int i = 0; i < arr.length; i++) {
if (arr[i] != b) {
System.out.print(arr[i] + " ");
newArray[a++] = arr[i];
b = arr[i];
}
}
}

public static void main(String[] args) {
int[] arr = { 1, 1, 2, 2, 2, 5 };
printArray(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

```
package logic;

public class subArraySum {

public static int Subarray(int[] arr, int a, int b) {
if (a <= b) {
System.out.println("Invalid");
return -1;
}
int max_sum = 0;
for (int i = 0; i < b; i++) {
max_sum += arr[i];
}
int window_sum = max_sum;
for (int i = b; i < a; i++) {
window_sum += arr[i] - arr[i - b];
max_sum = Math.max(max_sum, window_sum);
}
return max_sum;
}


public static void main(String[] args) {

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

int k = 2;

int len = arr.length;

System.out.println("Sum is: " + Subarray(arr, len, k));

}

}
```