Given two **sorted** arrays `A[]` and `B[]` of size **N** and **M**. The task is to merge both the arrays into a single array in **non-decreasing** order.

Input Format

First line take an integer input from user as **N** , where **N** is the size of `A[]` .

Second line takes **N** elements as Integer input in `A[]` .

Third line take an integer input from user as **M** , where **M** is the size of `B[]` .

Next line takes **M** elements as Integer input in `B[]` .

NOTE:- After answering the question, attempt the related question in the linked resource to improve your understanding of the question . Click here

Constraints

```
1<=N<=200

1<=M<=200

-10^4 <= A[i], B[j] <= 10^4
```

Output Format

Return the merged Array.

Sample Input 0

```
4
1 3 5 7
4
2 4 6 8
```

Sample Output 0

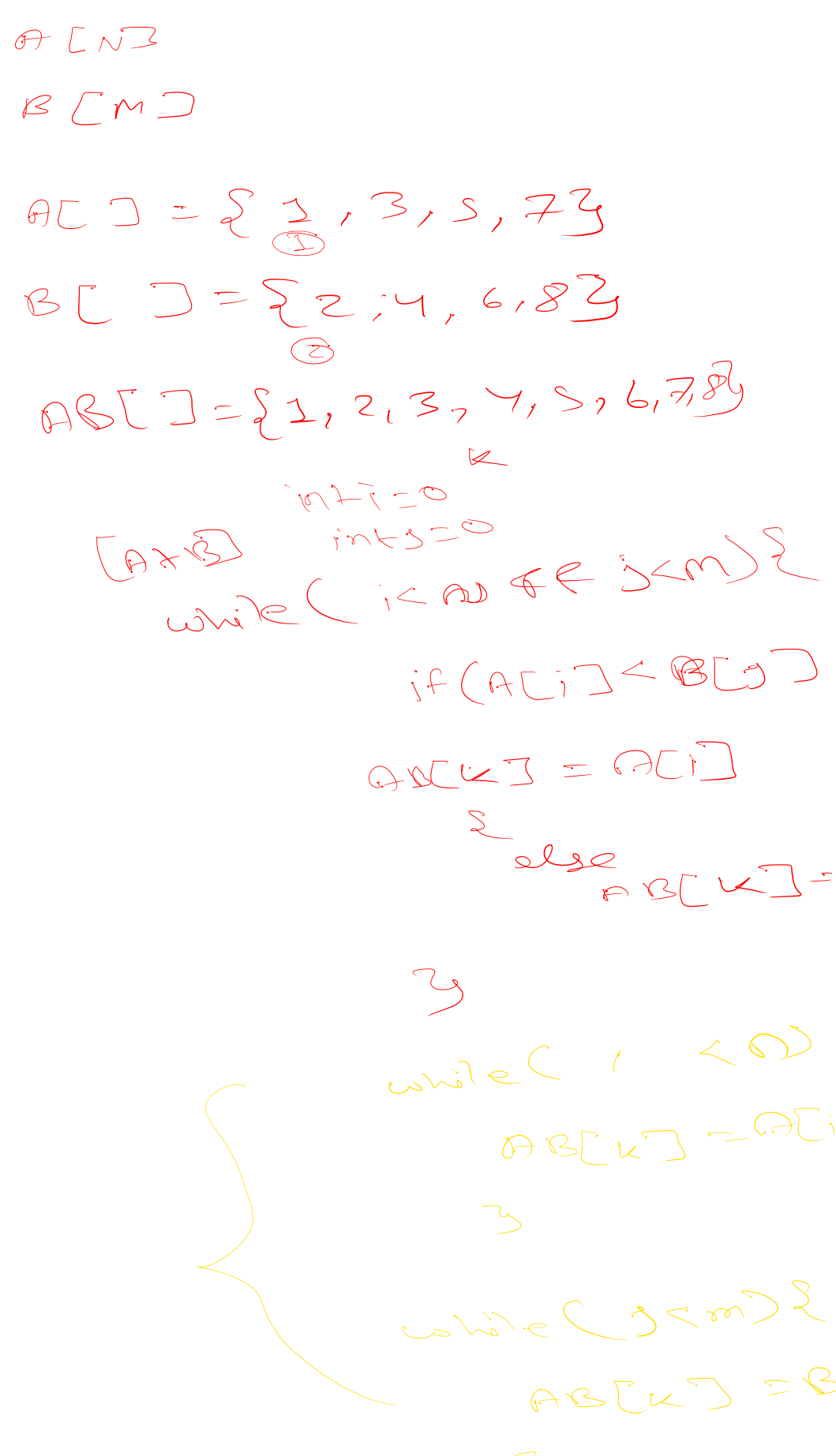
```
1 2 3 4 5 6 7 8
```

Explanation 0

The merged array contains all the elements from both arrays in sorted order.

Language: Java 15

```java
import java.io.*;
import java.util.*;

public class Solution {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        int A[]= new int[n];
        for(int i=0;i<n;i++)A[i]=sc.nextInt();

         int m = sc.nextInt();
        int B[]= new int[m];
        for(int i=0;i<m;i++)B[i]=sc.nextInt();

        int AB[] = new int[n+m];

        int i=0,j=0,k=0;

        while(i<n && j<m){
            if(A[i]<B[j]){
                AB[k]=A[i];
                i++;
            }else{
                AB[k]=B[j];
                j++;
            }
            k++;
        }

        while(i<n){
            AB[k]=A[i];
            i++;
            k++;
        }

        while(j<m){
            AB[k]=B[j];
            j++;
            k++;
        }

        for(int l=0;l<AB.length;l++)System.out.print(AB[l]+" ");
    }
}
```



```java
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();
    int[] arr = new int[n];
    for (int i = 0; i < n; i++) {
        arr[i] = scn.nextInt();
    }
    int ans = kadanesAlgorithm(arr, n);
    System.out.println(ans);
}
public static int kadanesAlgorithm(int[] arr, int n) {
    int maxSum = Integer.MIN_VALUE;
    int sumSoFar = 0;
    for (int i = 0; i < n; i++) {
        if ( sumSoFar < 0 ) {
            sumSoFar = arr[i];
        } else {
            sumSoFar += arr[i];
        }

        if (sumSoFar > maxSum) {
            maxSum = sumSoFar;
        }
    }
    return maxSum;
}
```

n^2

$O(n)$

[1, 2, 3, 4, 5]

[7, 2, 3, 4, 5]

1+2+3+4+5.

if

[-1, 2, 3, 4, 5]

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

if(sumSo<0) sumSofarEar(;);

sumSofar += arr[i];

if(sumSofar > maxSum)

maxSum = sumSofar;

}