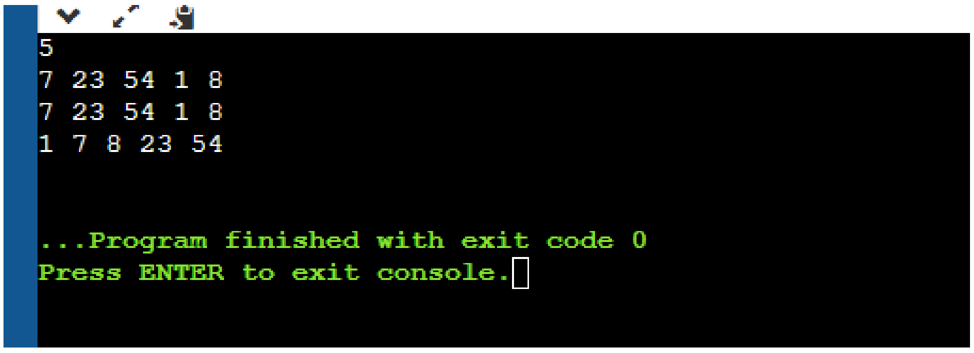
Program 1:

```
#include <iostream>

using namespace std;

void traverse(int arr[], int n){
    for(int i=0; i<n; i++){
        cout << arr[i] << " ";
    }
    cout << endl;
}

void merge(int arr[], int low, int mid, int high){
    int temp[high - low + 1];
    // cout << "size=" << (high-low+1)<< endl;
    mid = mid + 1;
    int l = low, m = mid, h = high;
    // cout << low << " " << mid << " " << high << endl;
    int i=0;
    while(l<mid && m<=high){
        if(arr[l] > arr[m]){
            // cout << "low>mid mid="<<m <<endl;
            temp[i++] = arr[m++];
        }else{
            // cout << "low<mid low="<<l <<endl;
            temp[i++] = arr[l++];
        }
    }
    while(l<mid){
        // cout << "inside norml while low="<<l <<endl;
        temp[i++] = arr[l++];
    }
    while(m<=high){
        // cout << "inside norml while mid="<<m <<endl;
        temp[i++] = arr[m++];
    }
    // cout<<"temp=";
    // traverse(temp, high - low + 1);
    int k = 0;
    for(int j=low; j<=high; j++){
        arr[j] = temp[k++];
    }
    // traverse(arr, 5);
}

void mergeSort(int arr[], int low, int high){
    if(low < high){
        int mid = (high + low)/2;
        mergeSort(arr, low, mid);
        mergeSort(arr, mid+1, high);
        merge(arr, low, mid, high);
    }
}

int main()
{
    int n;
    cin >> n;
    int arr[n];
    for(int i=0; i<n; i++){
        cin >> arr[i];
    }
    // int arr[5] = {5,4,3,2,1};
    // int n = 5;
    traverse(arr, n);
    mergeSort(arr, 0, n-1);
    traverse(arr, n);
    return 0;
}
```

Output:



```
5
7 23 54 1 8
7 23 54 1 8
1 7 8 23 54


...Program finished with exit code 0
Press ENTER to exit console.
```

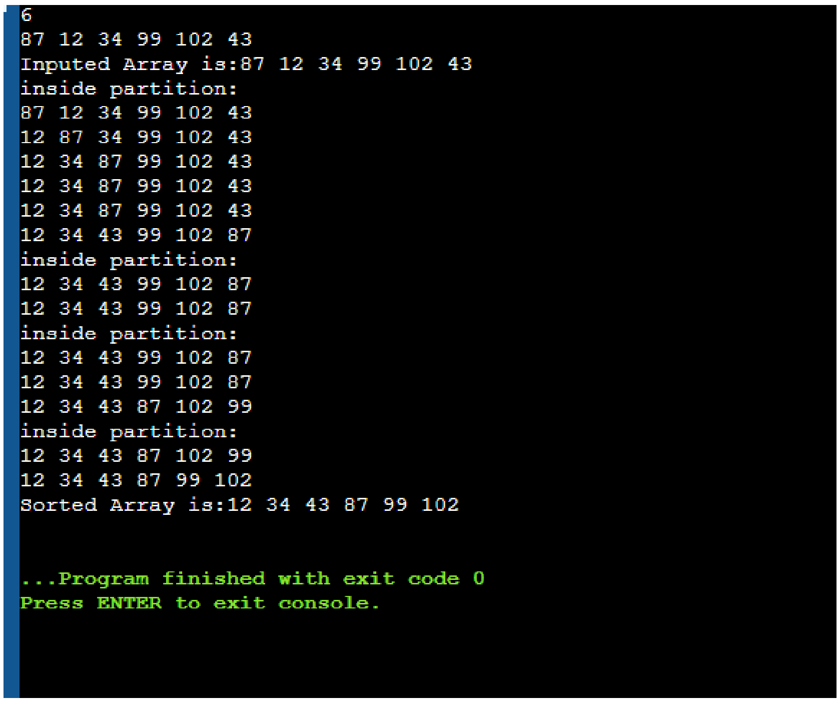
Program 2:

```
#include <bits/stdc++.h>


using namespace std;
void traverse(int arr[], int n){
    for(int i=0; i<n; i++){
        cout << arr[i] << " ";
    }
    cout << endl;
}
int partition (int arr[], int low, int high, int size)
{
    int pivot = arr[high];
    int i = (low - 1);
    cout << "inside partition: \n";
    for (int j = low; j <= high - 1; j++)
```

```
   {
      if (arr[j] < pivot)
      {
         i++;
         swap(arr[i], arr[j]);
      }
      traverse(arr, size);
   }
   swap(arr[i + 1], arr[high]);
   traverse(arr, size);
   return (i + 1);
}
void quickSort(int arr[], int low, int high, int size){
    if (low < high)
   {
      int pi = partition(arr, low, high, size);


      quickSort(arr, low, pi - 1, size);
      quickSort(arr, pi + 1, high, size);
   }
}
int main()
{
   int n;
   cin >> n;
   int arr[n];
   for(int i=0; i<n; i++){
      cin >> arr[i];
   }
   cout<<"Inputed Array is:";
   traverse(arr, n);
```

```
    quickSort(arr, 0, n-1, n);

    cout<<"Sorted Array is:";

    traverse(arr, n);


}
```

Output:



```
6
87 12 34 99 102 43
Inputed Array is:87 12 34 99 102 43
inside partition:
87 12 34 99 102 43
12 87 34 99 102 43
12 34 87 99 102 43
12 34 87 99 102 43
12 34 87 99 102 43
12 34 43 99 102 87
inside partition:
12 34 43 99 102 87
12 34 43 99 102 87
inside partition:
12 34 43 99 102 87
12 34 43 99 102 87
12 34 43 87 102 99
inside partition:
12 34 43 87 102 99
12 34 43 87 99 102
Sorted Array is:12 34 43 87 99 102


...Program finished with exit code 0
Press ENTER to exit console.
```