```
//Created By Ritwik Chandra Pandey on 6 Nov’ 21
//Bucket Sort: Max element method

/* In the Bucket Sort algorithm, all the elements to be sorted are initially distributed into number of buckets. Then each bucket is individually sorted
and then they are merged back to get the sorted order.

There are two ways of taking the buckets:
  1.  Taking the number of buckets equal to the maximum element in array to be sorted. For example for sorting 1,3,2,4,5,6,2, we would take 6
      buckets as 6 is the maximum of all the elements in the list.
  2.  Taking a fixed number of buckets and distributing the given elements into the buckets based on a logic.

This is method 1. */

#include <stdio.h>
#include <conio.h>


#include<stdlib.h>

void bucketSort(int array[], int n) {
        int i,j;
        int mymax=array[0];
        int *count;

        for(int i=1;i<n;i++){
                if(mymax<array[i]){
                        mymax = array[i];
                }
        }
        count = (int*)calloc(mymax+1,sizeof(int));

        for(i=0;i<n;i++){
                count[array[i]]++;
        }
        for(i=0,j=0;i<=mymax;i++){
                while(count[i]>0){
                        array[j] = i;
                        j++;
                        count[i]--;
                }
        }
        free(count);

}
void printArray(int arr[], int n) {
        for(int i= 0;i<n;i++)
        printf("%d\t",arr[i]);

        printf("\n");

}


int main()  {
        int size;
        int *arr, i;
        printf("Enter the size of an array: ");
   scanf("%d",&size);
        arr = (int*) malloc(size * sizeof(int));
        printf("Enter the %d array elements\n",size);
   for (i = 0; i < size; i++) {
                scanf("%d", &arr[i]);
   }
   printf("Array before sorting: \n");
        printArray(arr,size);

        bucketSort(arr,size);

        printf("Array after sorting: \n");
        printArray(arr,size);
        return 0;
}
```