AIM-Write a program to implement counting sort.

ALGORITHM-

```
countingSort(array, n) // 'n' is the size of array
max = find maximum element in the given array
create count array with size maximum + 1
Initialize count array with all 0's
for i = 0 to n
find the count of every unique element and
store that count at ith position in the count array
for j = 1 to max
Now, find the cumulative sum and store it in count array
for i = n to 1
Restore the array elements
Decrease the count of every restored element by 1
end countingSort
```

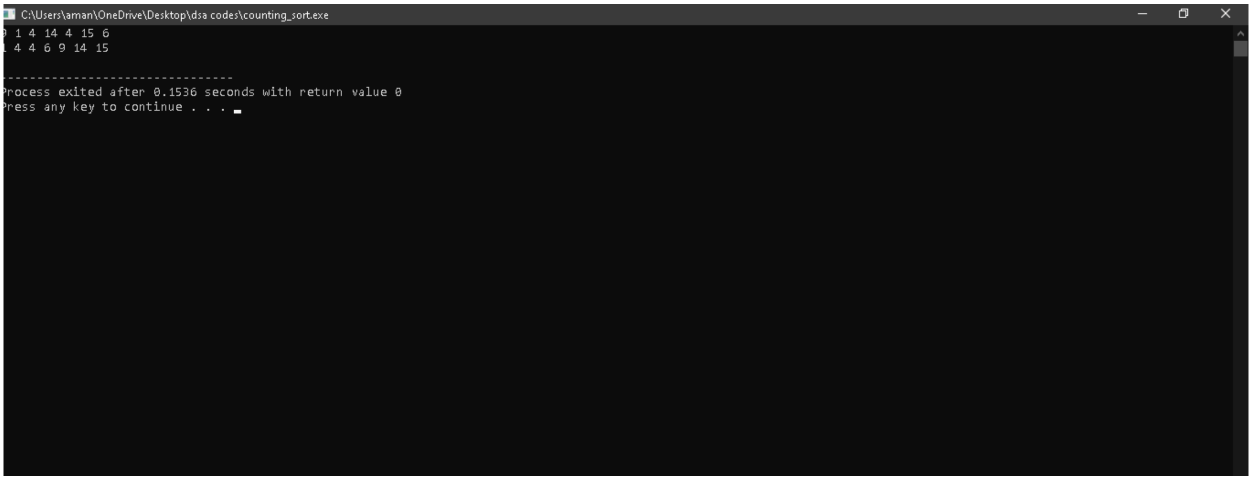
//code-

```
#include<stdio.h>
#include<limits.h>
#include<stdlib.h>
void printArray(int *A, int n){
  for (int i = 0; i < n; i++) {
    printf("%d ", A[i]);
  }
  printf("\n");
}
```

```
int maximum(int A[], int n){
    int max = INT_MIN;
    for (int i = 0; i < n; i++){
        if (max < A[i]){
            max = A[i];
        }
    }
    return max;
}
void countSort(int * A, int n){
    int i, j;
 int max = maximum(A, n);
   int* count = (int *) malloc((max+1)*sizeof(int));
   for (i = 0; i < max+1; i++) {
        count[i] = 0;
    }
for (i = 0; i < n; i++) {
        count[A[i]] = count[A[i]] + 1;
    }
    i =0;
    j =0;
  while(i<= max){
        if(count[i]>0){
            A[j] = i;
            count[i] = count[i] - 1;
            j++;
        }
        else{
            i++;
        }
    }
}

int main(){
    int A[] = {9, 1, 4, 14, 4, 15, 6};
    int n = 7;
    printArray(A, n);
    countSort(A, n);
    printArray(A, n);
    return 0;
}

//OUTPUT-
```



C:\Users\aman\OneDrive\Desktop\dsa codes\counting_sort.exe

```
9 1 4 14 4 15 6
1 4 4 6 9 14 15

--------------------------------
Process exited after 0.1536 seconds with return value 0
Press any key to continue . . .
```