Assignment 3

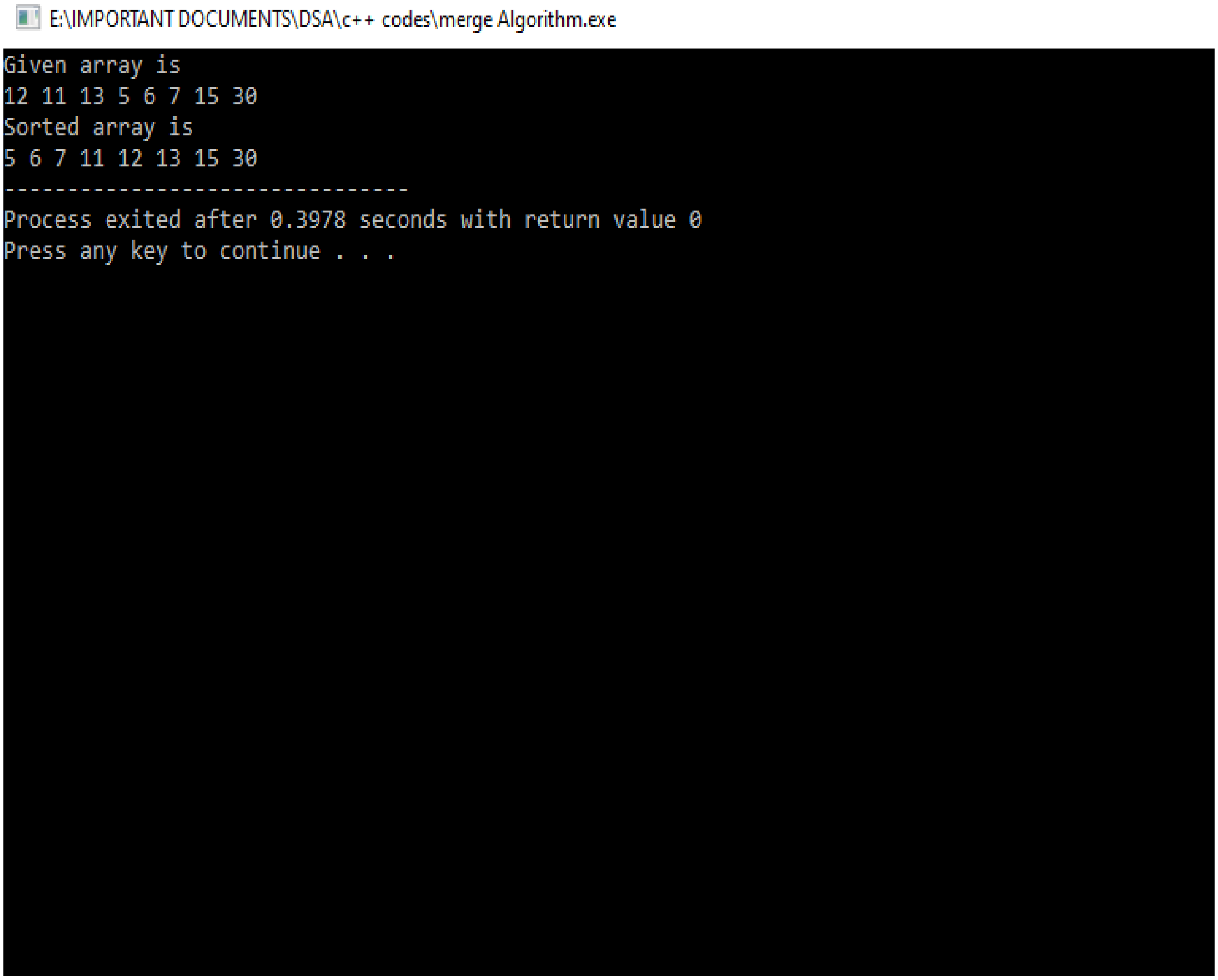
Question 1

Write a piece of code on **merge-sort** algorithm. Using RAM diagrams show the dynamic nature of this algorithm just on the first iteration.

```
#include <iostream>
using namespace std;
//A function for dividing array into two halves
void merge(int arr[], int l, int m, int r)
{
int n1 = m - l + 1;
int n2 = r - m;

int left[n1], right[n2];

for (int i = 0; i < n1; i++)
left[i] = arr[l + i];
for (int j = 0; j < n2; j++)
right[j] = arr[m + 1 + j];


int i = 0;

int j = 0;

int k = l;
}

while (i < n1 && j < n2) {
if (left[i] <= right[j]) {
arr[k] = left[i];
i++;
}
else {
arr[k] = right[j];
j++;
}
k++;
}

while (i < n1) {
arr[k] = left[i];
i++;
k++;
}

while (j < n2) {
arr[k] = right[j];
j++;
k++;
}
}
```

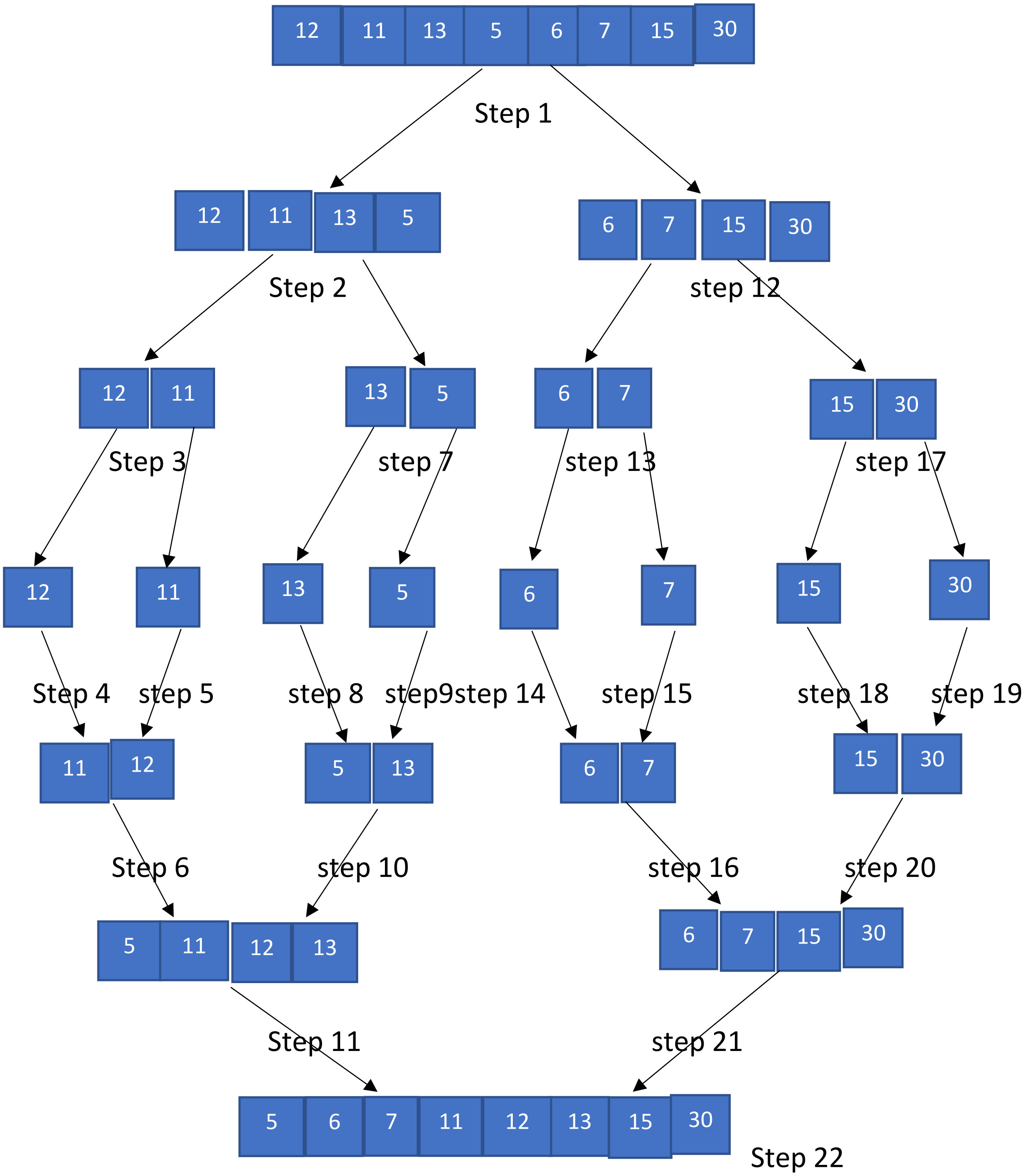
```
// this function merge sorts the array elements in ascending order
void mergeSort(int arr[],int l,int r){
if(l>=r){
return;
}
int m = (l+r-1)/2;
mergeSort(arr,l,m);
mergeSort(arr,m+1,r);
merge(arr,l,m,r);
}

//A function for displaying both array before sorting and after sorting
void printArray(int array[], int size)
{
for (int i = 0; i < size; i++)
cout << array[i] << " ";
}

//Here we call both functions
int main()
{
int arr[] = { 12, 11, 13, 5, 6, 7, 15, 30 };
int arr_size = sizeof(arr) / sizeof(arr[0]);
cout << "Given array is "<<endl;

printArray(arr, arr_size);

mergeSort(arr, 0, arr_size - 1);

cout << "\nSorted array is "<<endl;

printArray(arr, arr_size);

return 0;

}
```

PROGRAM OUTPUT:



```
Given array is
12 11 13 5 6 7 15 30
Sorted array is
5 6 7 11 12 13 15 30
--------------------------------
Process exited after 0.3978 seconds with return value 0
Press any key to continue . . .
```

IMPLEMENTATION IN RAM DIAGRAMS


12 11 13 5 6 7 15 30
Step 1
12 11 13 5
6 7 15 30
Step 2
step 12
12 11
13 5
6 7
15 30
Step 3
step 7
step 13
step 17
12
11
13
5
6
7
15
30
Step 4
step 5
step 8
step9step 14
step 15
step 18
step 19
11 12
5 13
6 7
15 30
Step 6
step 10
step 16
step 20
5 11 12 13
6 7 15 30
Step 11
step 21
5 6 7 11 12 13 15 30
Step 22