LAB 07

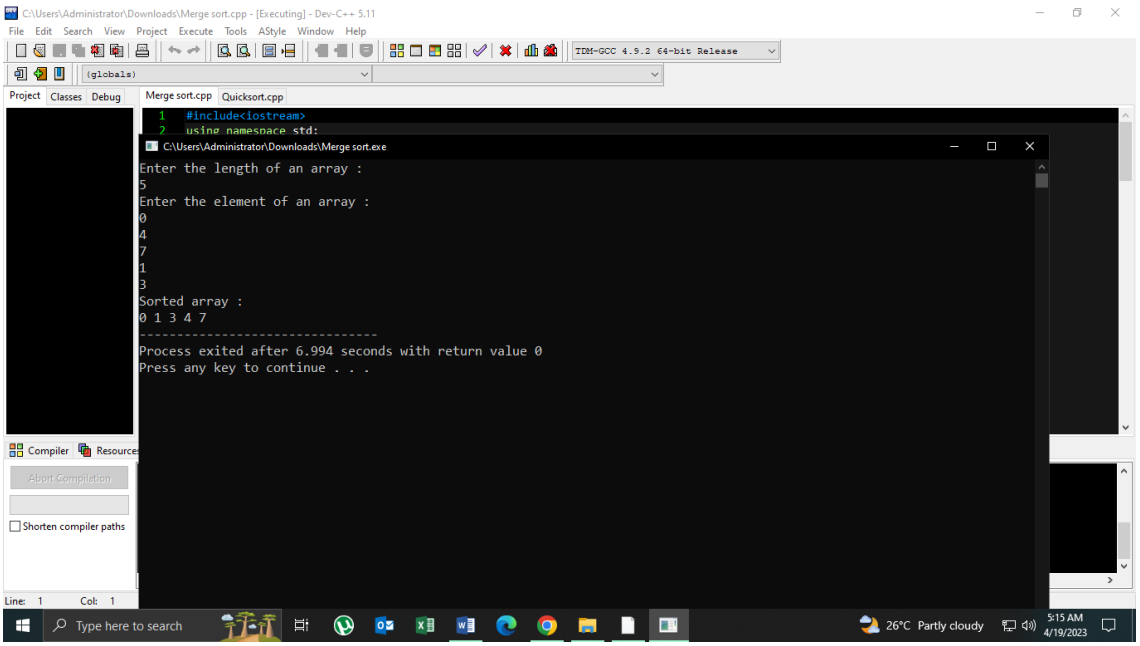
M.HUZAIFA MUSTAFA
SP22-BSCS-0046
AM SECTION

1. MERGE SORT.
   SOURCE CODE:

```
#include<iostream>
using namespace std;


void Merge(int arr[], int start, int middle, int end){
        int sizeOfFirstArray = middle - start + 1;
        int sizeOfSecondArray = end - middle;

        int FirstArray[sizeOfFirstArray];
        int SecondArray[sizeOfSecondArray];

        for(int i = 0; i < sizeOfFirstArray; i++){
                FirstArray[i] = arr[start + i];
        }

        for(int i = 0; i < sizeOfSecondArray; i++){
                SecondArray[i] = arr[middle + 1 + i];
        }

        int j = 0;
        int k = 0;
        int l = start;

        while(j < sizeOfFirstArray && k < sizeOfSecondArray){
                if(FirstArray[j] <= SecondArray[k]){
                        arr[l++] = FirstArray[j++];
                }
                else{
                        arr[l++] = SecondArray[k++];
                }
        }
```

```
	while(j < sizeOfFirstArray){
		arr[l++] = FirstArray[j++];
	}

	while(k < sizeOfSecondArray){
		arr[l++] = SecondArray[k++];
	}
}


void Mergesort(int arr[], int start, int end){
	if(start < end){
		int middle = start + (end - start)/2;
		Mergesort(arr, start, middle);
		Mergesort(arr, middle + 1, end);
		Merge(arr, start, middle, end);
	}
}


int main(){

	int length;

	cout<<"Enter the length of an array : "<<endl;
	cin>>length;

	int arr[length];

	cout<<"Enter the element of an array : "<<endl;

	for(int i = 0;i < length; i++){
		cin>>arr[i];
	}

	Mergesort(arr, 0, length - 1);
		cout<<"Sorted array : "<<endl;
		for(int i = 0; i < length ; i++){
			cout<<arr[i]<<" ";
		}
	return 0;
}
```

PICTURE:



2. QUICK SORT.

SOURCE CODE:

```
#include<iostream>
using namespace std;
int partition(int arr[], int start, int end){
   int pivot = arr[end];
   int firstpointer = start - 1;
   for (int secondpointer=start;secondpointer<end;secondpointer++){
      if (arr[secondpointer] < pivot){
         firstpointer++;
         int temp = arr[firstpointer];
         arr[firstpointer] = arr[secondpointer];
         arr[secondpointer] = temp;
      }
   }
   int temp = arr[firstpointer + 1];
   arr[firstpointer + 1] = arr[end];
   arr[end] = temp;
   return (firstpointer + 1);
}
void quicksort(int arr[], int start, int end){
   if (start<end){
      int middle = partition(arr, start, end);
      quicksort(arr, start, middle - 1);
      quicksort(arr, middle + 1, end);
   }
```

```
}
int main(){
   int n;
   cout << "Enter the size of the array: " << endl;
   cin >> n;
   int arr[n];
   cout << "Enter the elements in the array: " << endl;
   for(int i=0;i<n;i++){
      cin >> arr[i];
   }
   quicksort(arr, 0, n - 1);
   cout << "The sorted array is: " << endl;
   for(int i=0;i<n;i++){
      cout << arr[i] << " ";
   }
}
```

PICTURE: