

Name- Abhinav Kumar

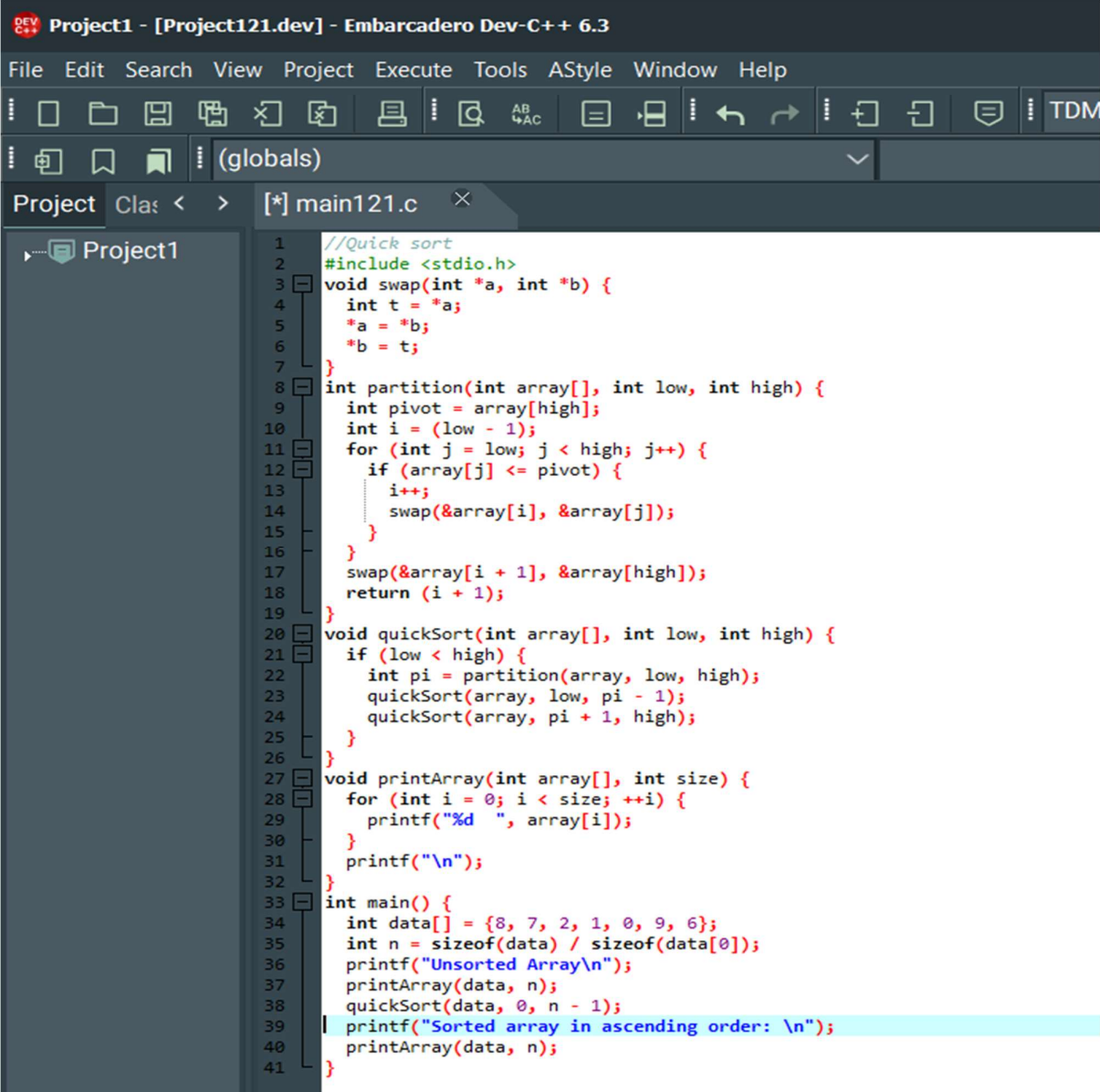
Branch- AIML-A1

PRN- 21070126006

Data Structure & Algorithm Assignment 3

Implement the following sort algorithm using a function (Separate programs):

1. Merge sort
2. Quick sort



```
1 //Quick sort
2 #include <stdio.h>
3 void swap(int *a, int *b) {
4     int t = *a;
5     *a = *b;
6     *b = t;
7 }
8 int partition(int array[], int low, int high) {
9     int pivot = array[high];
10    int i = (low - 1);
11    for (int j = low; j < high; j++) {
12        if (array[j] <= pivot) {
13            i++;
14            swap(&array[i], &array[j]);
15        }
16    }
17    swap(&array[i + 1], &array[high]);
18    return (i + 1);
19 }
20 void quickSort(int array[], int low, int high) {
21     if (low < high) {
22         int pi = partition(array, low, high);
23         quickSort(array, low, pi - 1);
24         quickSort(array, pi + 1, high);
25     }
26 }
27 void printArray(int array[], int size) {
28     for (int i = 0; i < size; ++i) {
29         printf("%d ", array[i]);
30     }
31     printf("\n");
32 }
33 int main() {
34     int data[] = {8, 7, 2, 1, 0, 9, 6};
35     int n = sizeof(data) / sizeof(data[0]);
36     printf("Unsorted Array\n");
37     printArray(data, n);
38     quickSort(data, 0, n - 1);
39     printf("Sorted array in ascending order: \n");
40     printArray(data, n);
41 }
```

C:\Users\AK-Lenovo\Documents\DEV++\Project121.exe

— □ ×

Unsorted Array

8 7 2 1 0 9 6

Sorted array in ascending order:

0 1 2 6 7 8 9

Process exited after 0.4687 seconds with return value 0

Press any key to continue . . .

DEV Project1 - [Project121.dev] - Embarcadero Dev-C++ 6.3

File Edit Search View Project Execute Tools AStyle Window

(globals)

Project Class < > main121.c

Project1

```
1 //Merge sort
2 #include <stdio.h>
3 void merge(int arr[], int p, int q, int r) {
4     int n1 = q - p + 1;
5     int n2 = r - q;
6     int L[n1], M[n2];
7     for (int i = 0; i < n1; i++)
8         L[i] = arr[p + i];
9     for (int j = 0; j < n2; j++)
10         M[j] = arr[q + 1 + j];
11     int i, j, k;
12     i = 0;
13     j = 0;
14     k = p;
15     while (i < n1 && j < n2) {
16         if (L[i] <= M[j]) {
17             arr[k] = L[i];
18             i++;
19         } else {
20             arr[k] = M[j];
21             j++;
22         }
23         k++;
24     } while (i < n1) {
25         arr[k] = L[i];
26         i++;
27         k++;
28     } while (j < n2) {
29         arr[k] = M[j];
30         j++;
31         k++;
32     } void mergeSort(int arr[], int l, int r) {
33         if (l < r) {
34             int m = l + (r - 1) / 2;
35             mergeSort(arr, l, m);
36             mergeSort(arr, m + 1, r);
37             merge(arr, l, m, r);
38         } void printArray(int arr[], int size) {
39             for (int i = 0; i < size; i++)
40                 printf("%d ", arr[i]);
41             printf("\n");
42         } int main() {
43             int arr[] = {6, 5, 12, 10, 9, 1};
44             int size = sizeof(arr) / sizeof(arr[0]);
45             mergeSort(arr, 0, size - 1);
46             printf("Sorted array: \n");
47             printArray(arr, size);
48         }
```

C:\Users\AK-Lenovo\Documents\DEV++\Project121.exe

— □ ×

Sorted array:

1 5 6 9 10 12

Process exited after 0.08125 seconds with return value 0

Press any key to continue . . .