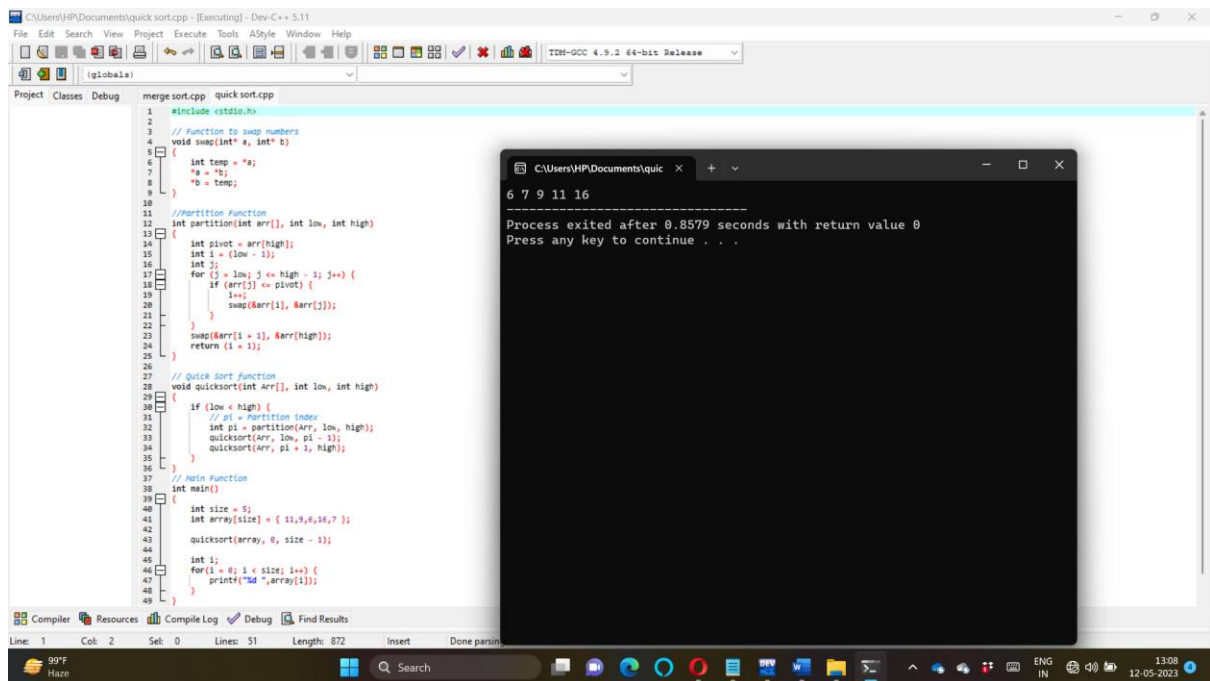


NAME:P.HARI HASSAN

REG.NO:192210633

DATE:12/05/23

## 1.C PROGRAMM FOR QUICK SORT



The screenshot displays a C++ IDE with a source code editor on the left and a console window on the right. The source code implements a Quick Sort algorithm. It includes a swap function, a partition function that selects the last element as a pivot and rearranges the array, and a recursive quicksort function. The main function initializes an array {11, 9, 8, 14, 7} and prints its elements after sorting. The console window shows the output '6 7 9 11 16' and a message indicating the process exited after 0.8579 seconds.

```
1 #include <stdio.h>
2
3 // function to swap numbers
4 void swap(int* a, int* b)
5 {
6     int temp = *a;
7     *a = *b;
8     *b = temp;
9 }
10
11 //Partition Function
12 int partition(int arr[], int low, int high)
13 {
14     int pivot = arr[high];
15     int i = (low - 1);
16     int j;
17     for (j = low; j <= high - 1; j++) {
18         if (arr[j] <= pivot) {
19             i++;
20             swap(&arr[i], &arr[j]);
21         }
22     }
23     swap(&arr[i + 1], &arr[high]);
24     return (i + 1);
25 }
26
27 // Quick Sort function
28 void quicksort(int arr[], int low, int high)
29 {
30     if (low < high) {
31         // pi = partition index
32         int pi = partition(arr, low, high);
33         quicksort(arr, low, pi - 1);
34         quicksort(arr, pi + 1, high);
35     }
36 }
37
38 // Main function
39 int main()
40 {
41     int size = 5;
42     int array[size] = { 11, 9, 8, 14, 7 };
43     quicksort(array, 0, size - 1);
44
45     int i;
46     for (i = 0; i < size; i++) {
47         printf("%d ", array[i]);
48     }
49 }
```

6 7 9 11 16  
-----  
Process exited after 0.8579 seconds with return value 0  
Press any key to continue . . .