

MD. ABU TOWSIF

ID:22-47019-1

SECTION: D

```
//Program to implement QUICKSORT in c++
#include <bits/stdc++.h>
using namespace std;

inputArr(int *arr, int s)
{
    cout << "Please enter the elements of the array: ";
    for (int i = 0; i < s; i++)
    {
        cin >> arr[i];
    }
    // cout << endl;
}

// rearrange the elements to get the actual pivot index
int partition(int arr[], int low, int high, int pivot)
{
    int PIndex = low;

    // Traversing the array from low to high
    for (int i = low; i <= high; i++)
    {
        // If the current element is smaller than the pivot element
        if (arr[i] <= pivot)
        {
            // Swap PIndex element with current element.
            swap(arr[PIndex], arr[i]);
            // Increment the pointer.
```

```

PIndex++;
}
}

PIndex--;

return PIndex;
}

// Dividing the array into two subarrays around
// the pivot and recursively call for them separately.
void quickSort(int arr[], int low, int high)
{
    if (low < high)
    {
        int pivot = arr[high];

        // Rearranging and get the actual pivot index
        int PIndex = partition(arr, low, high, pivot);

        // solve for the left and right subarrays
        quickSort(arr, low, PIndex - 1);
        quickSort(arr, PIndex + 1, high);
    }
}

int main()
{

    int n;
    cout << "Please enter the size of the array: ";
    cin >> n;

```

```

int *arr = new int[n];

inputArr(arr, n);

// Calling the quickSort function

quickSort(arr, 0, n - 1);

cout << "The sorted array is: ";

for (int i = 0; i < n; i++)
{
    cout << arr[i] << " ";
}

cout << endl;

return 0;
}

```

Screenshot of output:

