Name:-Dikesh Ganboi
Roll NO:- A36

```cpp
#include <iostream>
#include <vector>
#include <chrono>

using namespace std;
using namespace std::chrono;

// Function to partition the array
int partition(vector<int>& arr, int low, int high) {
    int pivot = arr[high]; // Choose the last element as pivot
    int i = low - 1; // Index of smaller element

    for (int j = low; j <= high - 1; j++) {
        // If current element is smaller than or equal to pivot
        if (arr[j] <= pivot) {
            i++; // Increment index of smaller element
            swap(arr[i], arr[j]);
        }
    }
    swap(arr[i + 1], arr[high]);
    return i + 1;
}

// Function to perform Quick Sort
void quickSort(vector<int>& arr, int low, int high) {
    if (low < high) {
        // Partition the array
        int pi = partition(arr, low, high);

        // Recursively sort elements before and after partition
        quickSort(arr, low, pi - 1);
        quickSort(arr, pi + 1, high);
    }
}

// Function to print array
void printArray(const vector<int>& arr) {
    for (int num : arr) {
        cout << num << " ";
    }
    cout << endl;
}

int main() {
    int n;
    cout << "Enter the number of elements: ";
    cin >> n;

    vector<int> arr(n);
    cout << "Enter " << n << " elements: ";
    for (int i = 0; i < n; ++i) {
        cin >> arr[i];
    }

    // Measure time taken by Quick Sort
    auto start = high_resolution_clock::now();
    quickSort(arr, 0, n - 1);
    auto stop = high_resolution_clock::now();
    auto duration = duration_cast<microseconds>(stop - start);

    cout << "Sorted array: ";
    printArray(arr);

    cout << "Time taken by Quick Sort: " << duration.count() << "
microseconds" << endl;

    return 0;
}
```

OUTPUT:-

```
PS C:\Users\HP\Desktop\DAA EXperiment> cd "c:\Users\HP\Desktop\DAA
EXperiment\" ; if ($?) { g++ Quicksort.cpp -o Quicksort } ; if ($?)
{ .\Quicksort }

Enter the number of elements: 6
Enter 6 elements: 34 15 19 20 17 9
Sorted array: 9 15 17 19 20 34
Time taken by Quick Sort: 0 microseconds
```