**QUICK SORT**

#include <iostream>

using namespace std;

// Swap

void swap(int& a, int& b) {

int temp = a;

a = b;

b = temp;

}

int partition(int arr[], int low, int high) {

int pivot = arr[high]; // Set pivot to the highest value

int i = low - 1; // Next location of pivot

for (int j = low; j < high; j++) {

if (arr[j] <= pivot) {

i++;

swap(arr[i], arr[j]);

}

}

swap(arr[i + 1], arr[high]);

return (i + 1); // Return the partitioning index for Pivot

}

// Quicksort function

void quickSort(int arr[], int low, int high) {

if (low < high) {

int loc = partition(arr, low, high);

// Sorting left and Right side of pivot

quickSort(arr, low, loc - 1);

quickSort(arr, loc + 1, high);

}

}

// Print the array function

void printArray(int arr[], int size) {

for (int i = 0; i < size; i++) {

cout << arr[i] << " ";

}

cout << endl;

}

// Main function

int main() {

int arr[] = {24, 49, 07, 90, 10, 3};

int n = sizeof(arr) / sizeof(arr[0]);

quickSort(arr, 0, n - 1);

cout << "Sorted array: \n";

printArray(arr, n);

return 0;

}