

**DSA Lab(Data Structured and Algorithms Lab)**

**Assignment # 11**

**Semester**: 3ndSemester

**Section**: C

**Submitted To:**

**Mr. Abdullah Shahrose**

**Submitted By:**

**Name**: Abdul Ahad

**Roll No**: 22-CS-071

**Task 1:**

#include <iostream>

struct Student {

    int id;

    std::string name;

    // Add other fields as needed

};

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

    int pivot = arr[high].id;

    int i = low - 1;

    for (int j = low; j <= high - 1; j++) {

        if (arr[j].id < pivot) {

            i++;

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

        }

    }

    std::swap(arr[i + 1], arr[high]);

    return i + 1;

}

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

    if (low < high) {

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

        quickSort(arr, low, pi - 1);

        quickSort(arr, pi + 1, high);

    }

}

int main() {

    // Test the quickSort function

    const int size = 5;

    Student students[size] = {

        {3, "Alice"},

        {1, "Bob"},

        {4, "Charlie"},

        {2, "David"},

        {5, "Eve"}

    };

    std::cout << "Before sorting:" << std::endl;

    for (int i = 0; i < size; i++) {

        std::cout << students[i].id << ": " << students[i].name << std::endl;

    }

    quickSort(students, 0, size - 1);

    std::cout << "After sorting:" << std::endl;

    for (int i = 0; i < size; i++) {

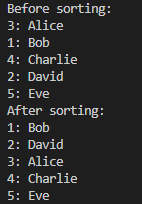
        std::cout << students[i].id << ": " << students[i].name << std::endl;

    }

    return 0;

}

**Output:**

****

**Task 2:**

#include <iostream>

#include <vector>

struct Employee {

    std::string name;

    int salary;

};

int partition(std::vector<Employee>& arr, int low, int high) {

    int pivot = arr[high].salary;

    int i = low - 1;

    for (int j = low; j <= high - 1; j++) {

        if (arr[j].salary >= pivot) {

            i++;

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

        }

    }

    std::swap(arr[i + 1], arr[high]);

    return i + 1;

}

void quickSort(std::vector<Employee>& arr, int low, int high) {

    if (low < high) {

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

        quickSort(arr, low, pi - 1);

        quickSort(arr, pi + 1, high);

    }

}

int main() {

    std::vector<Employee> employees = {

        {"John", 5000},

        {"Alice", 3000},

        {"Bob", 7000},

        {"David", 4000},

        {"Eva", 6000}

    };

    int n = employees.size();

    quickSort(employees, 0, n - 1);

    std::cout << "Sorted employees based on salary in descending order:\n";

    for (const auto& employee : employees) {

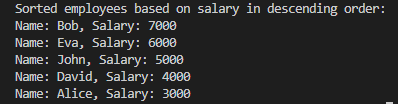
        std::cout << "Name: " << employee.name << ", Salary: " << employee.salary << std::endl;

    }

    return 0;

}

**Output:**

****