Lab Assignment 1

# Name – Akshat Chandrapatle Roll no. – 45 Div. – CS-A PRN - 12111449

# **WAP to implement Insertion and Quick sort on 1D array of Faculty structure (contains faculty\_name, faculty\_ID, subject\_codes, class\_names), with key as faculty\_ID. And count the number of swap performed**

* Quick Sort

**Code:**

#include <stdio.h>

struct Faculty {

char name[32];

int ID;

char subject\_codes[32];

char class\_names[32];

};

struct Faculty arr[] = {

{"Steve", 44, "cs101","class1"},

{"Bob", 21, "math101","class3"},

{"Dustin", 33, "physics101", "class5"},

{"Mike", 14, "history101", "class7"},

{"Nancy", 5, "english101", "class10"}

};

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

int swap\_count;

int partition(int low, int high) {

int pivot = arr[low].ID;

int i = low + 1;

int j = high;

while (1) {

while (i <= j && arr[i].ID <= pivot)

i++;

while (arr[j].ID > pivot)

j--;

if (i <= j) {

struct Faculty temp = arr[i];

arr[i] = arr[j];

arr[j] = temp;

i++;

j--;

swap\_count++;

} else {

struct Faculty temp = arr[j];

arr[j] = arr[low];

arr[low] = temp;

swap\_count++;

return j;

break;

}

}

}

void quick\_sort(int low, int high) {

if (low < high) {

int pi = partition(low, high);

quick\_sort(low, pi - 1);

quick\_sort(pi + 1, high);

}

}

int main() {

printf("Original list:\n");

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

printf("%s %d\n", arr[i].name, arr[i].ID);

}

swap\_count = 0;

quick\_sort( 0, n - 1);

printf("\nSorted Array :\n");

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

printf("%s %d\n", arr[i].name, arr[i].ID);

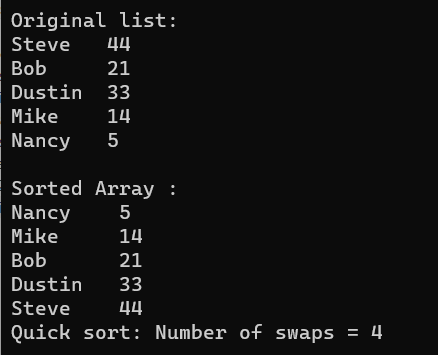
}

printf("Quick sort: Number of swaps = %d\n", swap\_count);

return 0;

}

**Output :**

****

* Insertion Sort:

**Code:**

#include <stdio.h>

struct Faculty {

char faculty\_name[30];

int faculty\_ID;

char subject\_codes[30];

char class\_names[30];

};

struct Faculty arr[5] = {

{"Steve", 44, "physics101","class1"},

{"Bob", 21, "maths101","class3"},

{"Dustin", 33, "physics101", "class5"},

{"Mike", 14, "history101", "class7"},

{"Nancy", 5, "english101", "class10"}

};

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

void insertionSort(struct Faculty arr[], int n) {

int i, j, key, swap\_count = 0;

for (i = 1; i < n; i++) {

key = arr[i].faculty\_ID;

j = i-1;

while (j >= 0 && arr[j].faculty\_ID > key) {

arr[j + 1].faculty\_ID = arr[j].faculty\_ID;

j--;

swap\_count++;

}

arr[j + 1].faculty\_ID = key;

}

printf("\nNumber of swaps performed: %d\n", swap\_count);

}

int main() {

printf("Before sorting \n");

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

printf("%s\t%d\t%s\t%s\n", arr[i].faculty\_name, arr[i].faculty\_ID, arr[i].subject\_codes, arr[i].class\_names);

}

insertionSort(arr, n);

printf("\nSorted Array :\n");

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

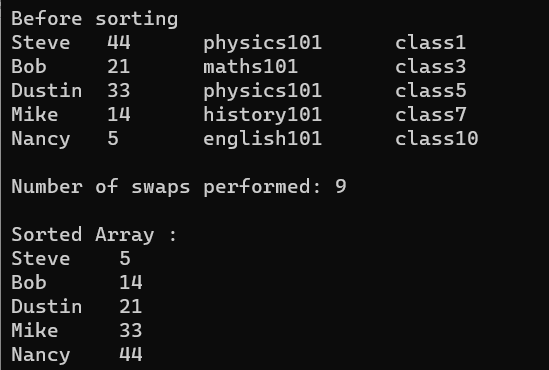
printf("%s\t %d\n", arr[i].faculty\_name, arr[i].faculty\_ID);

}

return 0;

}

**Output:**

****