**DAA PRACTICAL**

**Name:Parth Borse Roll No.: 07**

**QUICK SORT PROGRAM**

#include<iostream>

using namespace std;

void quick(int a[], int l, int up); // Function to perform quicksort

int partition(int a[], int l, int up);

int main() {

int n;

cout << "Enter the size of the array: "; // Prompt user to enter the size of the array

cin >> n;

int arr[n];

cout << "Enter array elements:" << endl;

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

cin >> arr[i];

}

int low = 0;

int up = n - 1;

// Call quicksort function to sort the array

quick(arr, low, up);

// Display sorted elements

cout << "Sorted elements are:" << endl;

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

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

}

cout << endl;

return 0;

}

// Function to perform quicksort

void quick(int a[], int l, int up) {

// If the lower index is greater than or equal to the upper index, return

if (l >= up) {

return;

}

// Partition the array and get the pivot location

int pvtloc = partition(a, l, up);

// Recursively call quicksort on left sublist

quick(a, l, pvtloc - 1);

// Recursively call quicksort on right sublist

quick(a, pvtloc + 1, up);

}

// Function to partition the array

int partition(int a[], int l, int up) {

if (l >= up) {

return l;

}

int temp, pvt;

int i = l + 1;

int j = up;

pvt = a[l];

// Partitioning loop

while (i <= j) {

while (a[i] < pvt) {

i++;

}

while (a[j] > pvt) {

j--;

}

// Swap elements if i < j

if (i < j) {

temp = a[i];

a[i] = a[j];

a[j] = temp;

i++;

j--;

} else {

i++;

}

}

// Swap pivot with element at position j

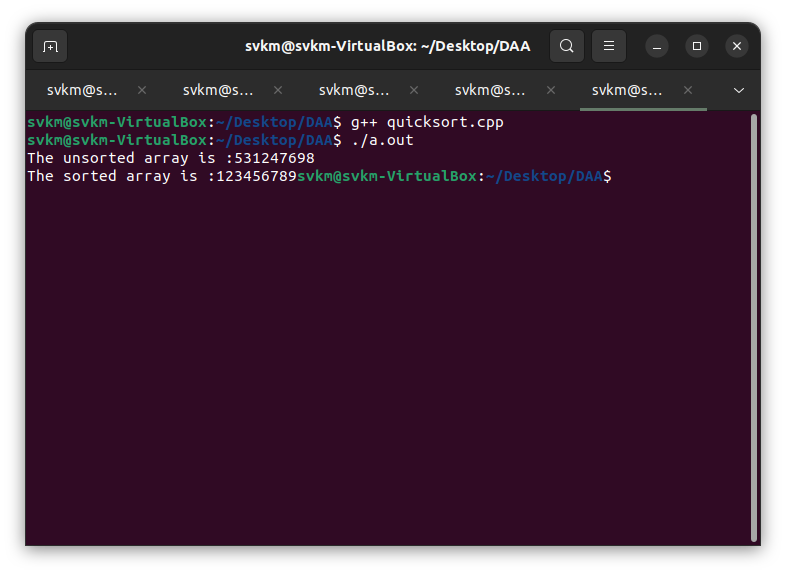
temp = a[l];

a[l] = a[j];

a[j] = temp;

return j; // Return pivot location

}

//OUTPUT****