

**Lab No: 18 Date: 2081/**

**Title: Write a program to sort the user input data in ascending or descending order using Selection sort**

Selection Sort is a comparison-based sorting algorithm that organizes an array by continuously identifying the smallest (or largest) element from the unsorted section and swapping it with the first unsorted element. Initially, the smallest element is located and swapped with the first securing its correct position. The process is then repeated by finding the next smallest element and placing it in the second position. This step-by-step approach continues until all elements are in their proper positions, resulting in a fully sorted array.It belongs to the class of comparison-based sorting algorithms. The

Sort is a simple sorting algorithm that works by repeatedly swapping adjacent elements if the

e so the wrong order, they are swapped. This process is repeated until the entire array is sorted.

**IDE: Visual Studio Code**

**Langauage: C**

**Source code :**

#include <stdio.h>

#include <conio.h>

void insertionSort(int arr[], int n)

{

    int least, p, i, j, k, temp, pass = 1, key;

    for (i = 0; i < n; i++)

    {

        key = arr[i];

        j = i - 1;

        printf("\nPass %d: \n", pass++);

        while (j >= 0 && arr[j] > key)

        {

            arr[j + 1] = arr[j];

            j = j - 1;

        }

        arr[j + 1] = key;

        for (k = 0; k < n; k++)

        {

            printf("%d, ", arr[k]);

        }

        printf("\n");

        printf("inserted value: %d interchange it's position\n", key);

    }

}

int main()

{

    int n, i;

    printf("Enter the size of array: ");

    scanf("%d", &n);

    int arr[n];

    printf("Enter the array data:\n"); // Taking input from user

    for (i = 0; i < n; i++)

    {

        scanf("%d", &arr[i]);

    }

    insertionSort(arr, n);    // Calling insertion sort on array arr

    printf("Sorted array: "); // Printing the final result

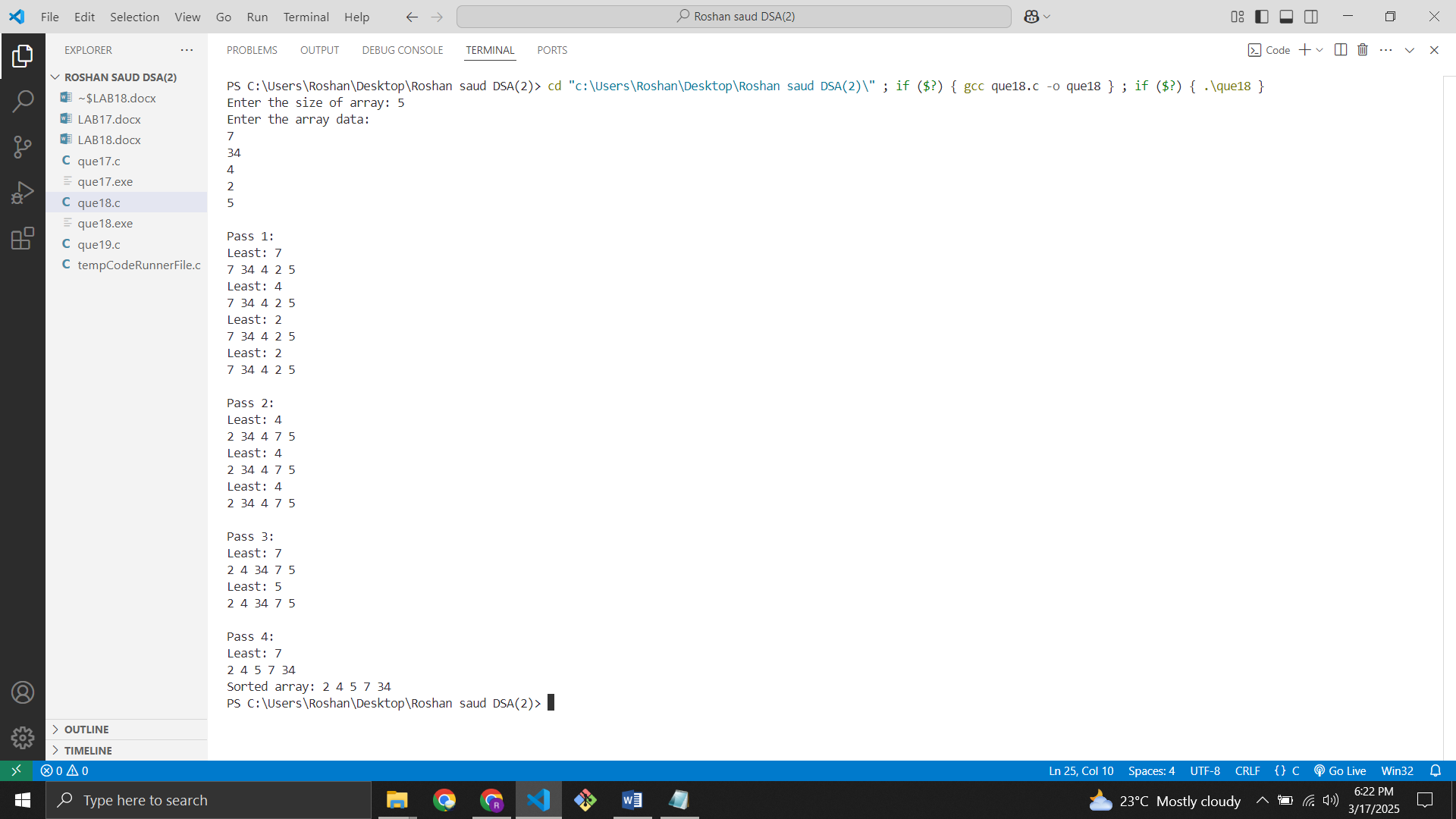
    for (int i = 0; i < n; i++)

        printf("%d ", arr[i]);

    return 0;

}

**Output:**

****