...for a complete human being
Texas
International college

Lab No: 19 **Date: 2081/**

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

Insertion sort is a simple, in-place, and stable sorting algorithm that builds a sorted array one element at a time by placing each unsorted element in its correct position, much like sorting playing cards in hand. It works by dividing the list into a sorted and an unsorted section. Initially, the first element is considered sorted. Then, one by one, each unsorted element is picked and compared with the elements in the sorted section, shifting larger elements to the right until the correct position is found. This process repeats until the entire array is sorted. Though inefficient for large datasets compared to quicksort or mergesort, insertion sort performs well on small or nearly sorted data due to its adaptive nature. It has a worst-case time complexity of $O(n^2)$ but remains efficient for small or mostly sorted datasets.

IDE: Visual Studio Code

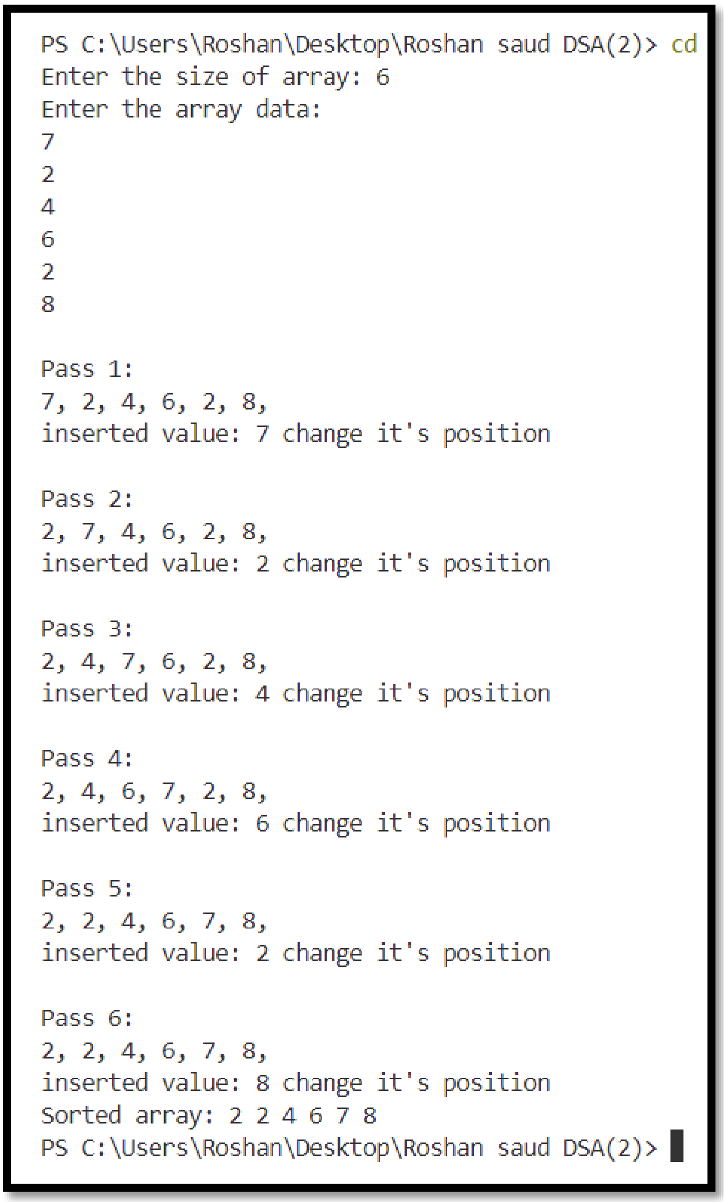
Language: C

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:



```
PS C:\Users\Roshan\Desktop\Roshan saud DSA(2)> cd
Enter the size of array: 6
Enter the array data:
7
2
4
6
2
8

Pass 1:
7, 2, 4, 6, 2, 8,
inserted value: 7 change it's position

Pass 2:
2, 7, 4, 6, 2, 8,
inserted value: 2 change it's position

Pass 3:
2, 4, 7, 6, 2, 8,
inserted value: 4 change it's position

Pass 4:
2, 4, 6, 7, 2, 8,
inserted value: 6 change it's position

Pass 5:
2, 2, 4, 6, 7, 8,
inserted value: 2 change it's position

Pass 6:
2, 2, 4, 6, 7, 8,
inserted value: 8 change it's position
Sorted array: 2 2 4 6 7 8
PS C:\Users\Roshan\Desktop\Roshan saud DSA(2)>
```