

Lab No: 17

Date: 2081/10/04

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

Bubble Sort is the simplest sorting algorithm that works by repeatedly swapping the adjacent elements if they are in the wrong order. This algorithm is not suitable for large data sets as its average and worst-case time complexity are quite high. We sort the array using multiple passes. After the first pass, the maximum element goes to end (its correct position). Same way, after second pass, the second largest element goes to second last position and so on. In every pass, we process only those elements that have already not moved to correct position. After k passes, the largest k elements must have been moved to the last k positions. In a pass, we consider remaining elements and compare all adjacent and swap if larger element is before a smaller element. If we keep doing this, we get the largest (among the remaining elements) at its correct position.

IDE: Visual Studio Code

Language: C

Source code:

```
#include <stdio.h>
#include <conio.h>
void bubbleSort(int arr[], int n)
{
    int pass = 1;
    for (int i = 0; i < n - 1; i++)
    {
        printf("\nPass %d: \n", pass++);
        for (int j = 0; j < n - i - 1; j++)
        {
            // To swap the value
            if (arr[j] > arr[j + 1])
            {
                int temp = arr[j];
                arr[j] = arr[j + 1];
                arr[j + 1] = temp;
            }
        }
        for (int k = 0; k < n; k++)
        {
            printf("%d, ", arr[k]);
        }
        printf("\n");
    }
}

int main()
{
    int n, i;
    printf("Enter the size of array: ");
    scanf("%d", &n);
```

```

int arr[n];
printf("Enter the array data:\n");
for (i = 0; i < n; i++)
{
    scanf("%d", &arr[i]);
}
bubbleSort(arr, n);
printf("Sorted array: ");
for (int i = 0; i < n; i++)
    printf("%d ", arr[i]);
return 0;
}

```

Output:

```

Enter the size of array: 7
Enter the array data:
18
6
2
19
4
27
8

Pass 1:
6, 18, 2, 19, 4, 27, 8,
6, 2, 18, 19, 4, 27, 8,
6, 2, 18, 19, 4, 27, 8,
6, 2, 18, 4, 19, 27, 8,
6, 2, 18, 4, 19, 27, 8,
6, 2, 18, 4, 19, 8, 27,

Pass 2:
2, 6, 18, 4, 19, 8, 27,
2, 6, 18, 4, 19, 8, 27,
2, 6, 4, 18, 19, 8, 27,
2, 6, 4, 18, 19, 8, 27,
2, 6, 4, 18, 8, 19, 27,

Pass 3:
2, 6, 4, 18, 8, 19, 27,
2, 4, 6, 18, 8, 19, 27,
2, 4, 6, 18, 8, 19, 27,
2, 4, 6, 8, 18, 19, 27,

Pass 4:
2, 4, 6, 8, 18, 19, 27,
2, 4, 6, 8, 18, 19, 27,
2, 4, 6, 8, 18, 19, 27,

Pass 5:
2, 4, 6, 8, 18, 19, 27,
2, 4, 6, 8, 18, 19, 27,

Pass 6:
2, 4, 6, 8, 18, 19, 27,
Sorted array: 2 4 6 8 18 19 27
PS C:\Users\user\OneDrive\Desktop\DSA\ManjilBajgain>

```

Lab No: 17

Date: 2081/10/04

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. It sorts an array by repeatedly selecting the **smallest (or largest)** element from the unsorted portion and swapping it with the first unsorted element. This process continues until the entire array is sorted.

- First, we find the smallest element and swap it with the first element. This way we get the smallest element at its correct position.
- Then we find the smallest among remaining elements (or second smallest) and swap it with the second element.
- We keep doing this until we get all elements moved to correct position.

IDE: Visual Studio Code

Language: C

Source code :

```
#include <stdio.h>
#include <conio.h>
void selectionSort(int arr[], int n)
{
    int least, p, i, j, k, temp, pass = 1;
    for (i = 0; i < n - 1; i++)
    {
        least = arr[i];
        p = i;
        printf("\nPass %d: \n", pass++);
        for (j = i + 1; j < n; j++)
        {
            if (arr[j] < arr[p])
            {
                least = arr[j];
                p = j;
            }
            printf("Least: %d\n", least);
            for (k = 0; k < n; k++)
            {
                printf("%d ", arr[k]);
            }
            printf("\n");
        }
        printf("\n");
        temp = arr[i];
        arr[i] = arr[p];
        arr[p] = temp;
    }
}

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]);
    }
    selectionSort(arr, n);
    printf("Sorted array: ");
    for (int i = 0; i < n; i++)
        printf("%d ", arr[i]);
    return 0;
}
```

Output:

```
Enter the size of array: 4
Enter the array data:
12
23
44
3
Pass 1:
Least: 12
12 23 44 3
Least: 12
12 23 44 3
Least: 3
12 23 44 3

Pass 2:
Least: 23
3 23 44 12
Least: 12
3 23 44 12

Pass 3:
Least: 23
3 12 44 23

Sorted array: 3 12 23 44
PS C:\Users\user\OneDrive\Desktop\DSA\ManjilBajgain>
```