**Assignment-2**

1.Pseudocode and Flowchart for Sorting Algorithm – Write pseudocode and create a flowchart for a bubble sort algorithm. Provide a brief explanation of how the algorithm works and a simple array of integers to demonstrate a dry run of your algorithm.

**Pseudocode:**

Bubble Sort(a[],n)

For i=0 to n-1

Swap=false

For j=i+1 to n

If a[j-1]>a[j]

Swap(a[j-1],a[j])

Swap=true

Break if not swapped

**Code:**

#include <stdio.h>

void swap(int\* x, int\* y)

{

int temp = \*x;

\*x = \*y;

\*y = temp;

}

void bubble\_sort(int arr[], int n)

{

int i, j;

for (i = 0; i < n - 1; i++)

for (j = 0; j < n - i - 1; j++)

if (arr[j] > arr[j + 1])

swap(&arr[j], &arr[j + 1]);

}

int main()

{

int arr[] = { 12, 46, 34, 82, 10, 9, 28 };

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

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

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

printf("%d ", arr[i]);

bubble\_sort(arr, n);

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

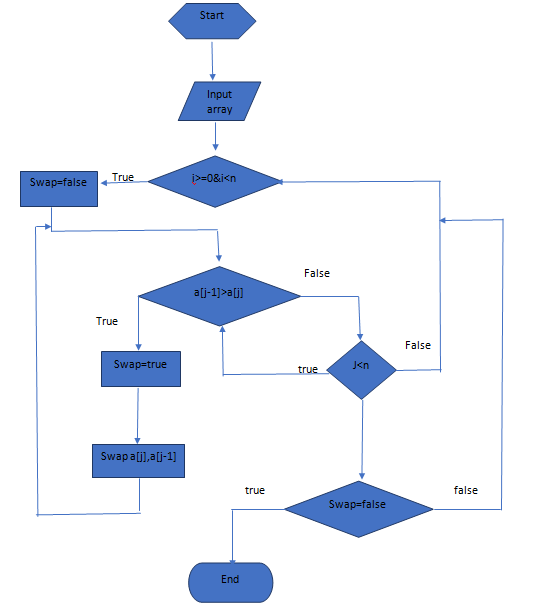
for (int i = 0; i < n; i++)

printf("%d ", arr[i]);

return 0;

}

**Flow Chart:**



Bubble sort is the simplest sorting algorithm and is useful for small amounts of data, Bubble sort implementation is based on swapping the adjacent elements repeatedly if they are not sorted. Bubble sort's time complexity in both of the cases (average and worst-case) is quite high. For large amounts of data, the use of Bubble sort is not recommended.

The basic logic behind this algorithm is that the computer selects the first element and performs swapping by the adjacent element if required based on the kind of sorting i.e. ascending and descending till it reaches the last element this is known as a pass. The computer performs multiple such passes till all the elements are sorted or no more swapping is possible.