**Program – 3**

**AIM - Write an algorithm and program to sort n numbers using Bubble sort technique.**

**Algorithm**

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

**Source Code -**

#include <stdio.h>

void swap(int \*xp, int \*yp)

{

int temp = \*xp;

\*xp = \*yp;

\*yp = temp;

}

voidbubbleSort(intarr[], int n)

{

inti, j;

bool swapped;

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

{

swapped = false;

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

{

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

{

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

swapped = true;

}

}

if (swapped == false)

break;

}

}

voidprintArray(intarr[], int size)

{

inti;

for (i=0; i< size; i++)

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

printf("n");

}

int main()

{

intarr[] = {64, 34, 25, 12, 22, 11, 90};

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

bubbleSort(arr, n);

printf("Sorted array: \n");

printArray(arr, n);

return 0;

}

**Implement recursion in the above question**

**Source Code –**

#include <bits/stdc++.h>

using namespace std;

voidbubbleSort(intarr[], int n)

{

if (n == 1)

return;

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

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

swap(arr[i], arr[i+1]);

bubbleSort(arr, n-1);

}

voidprintArray(intarr[], int n)

{

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

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

printf("\n");

}

int main()

{

intarr[] = {64, 34, 25, 12, 22, 11, 90};

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

bubbleSort(arr, n);

printf("Sorted array : \n");

printArray(arr, n);

return 0;

}