

Exam: DS Lab Exam

Submitted By: Tanmay Vig

Roll Num: 19BCS061

Ans1:

Source code:

```
#include <stdio.h>

int main()
{
    int n, array[1000], i, j, t, flag = 0;

    printf("Enter number of elements\n");
    scanf("%d", &n);

    printf("Enter %d integers\n", n);

    for (i = 0; i < n; i++)
        scanf("%d", &array[i]);

    for (i = 1 ; i <= n - 1; i++) {
        t = array[i];

        for (j = i - 1 ; j >= 0; j--) {
            if (array[j] > t) {
                array[j+1] = array[j];
                flag = 1;
            }
            else
                break;
        }
    }
```

```

    if (flag)
        array[j+1] = t;
    }

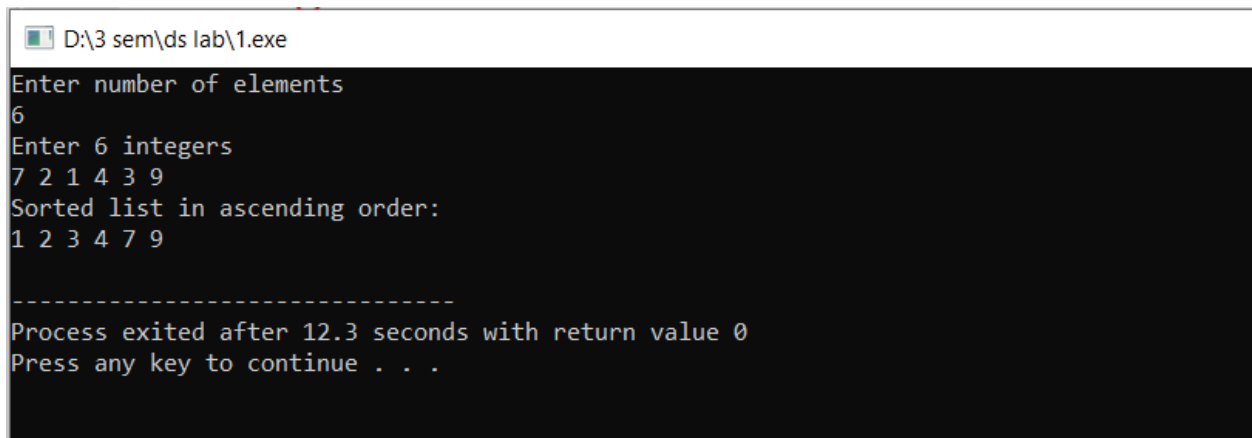
    printf("Sorted list in ascending order:\n");

    for (i = 0; i <= n - 1; i++) {
        printf("%d ", array[i]);
    }printf("\n");

    return 0;
}

```

Output:



```

D:\3 sem\ds lab\1.exe
Enter number of elements
6
Enter 6 integers
7 2 1 4 3 9
Sorted list in ascending order:
1 2 3 4 7 9

-----
Process exited after 12.3 seconds with return value 0
Press any key to continue . . .

```

Ans2:

Source Code:

```

#include <stdio.h>

#include <stdlib.h>

#include <stdbool.h>

```

```

void BFS(int src, int V, int graph[V][V]){
    bool visited[V];

    int i=0;
    for (i = 0; i < V; ++i) visited[i] = false;


    int queue[V], front = 0, back = 0;


    visited[src] = true;
    queue[back++] = src;


    while (front < back) {
        int first = queue[front];
        printf("%d ",queue[front++]);


        for (i = 0; i < V; ++i) {
            if (graph[first][i] && !visited[i]) {
                visited[i] = true;
                queue[back++] = i;
            }
        }
    }
}

```

```

int main() {
    int V,i=0,j=0;

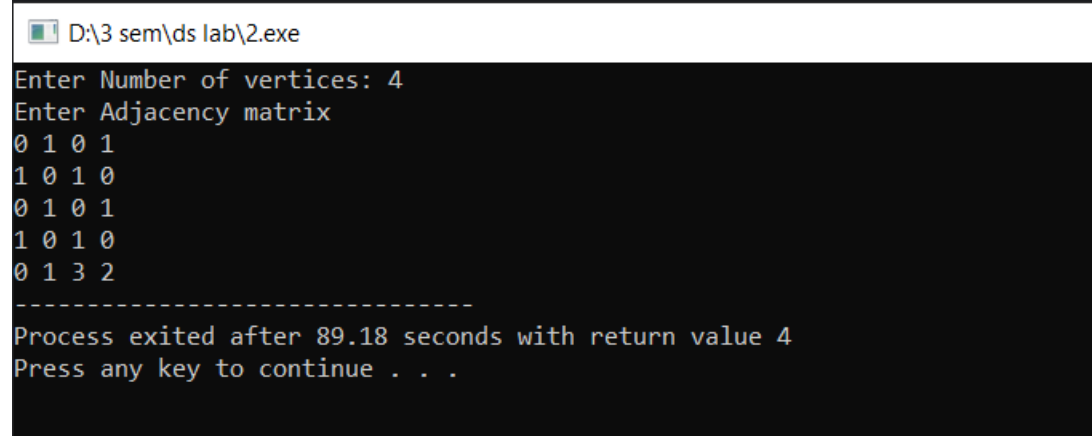
    printf("Enter Number of vertices: ");
    scanf("%d",&V);

    int graph[V][V];

```

```
printf("Enter Adjacency matrix\n");  
for (i = 0; i < V; ++i) {  
    for (j = 0; j < V; ++j) {  
        scanf("%d",&graph[i][j]);  
    }  
}  
BFS(0,V,graph);  
return 0;  
}
```

Output:



```
D:\3 sem\ds lab\2.exe  
Enter Number of vertices: 4  
Enter Adjacency matrix  
0 1 0 1  
1 0 1 0  
0 1 0 1  
1 0 1 0  
0 1 3 2  
-----  
Process exited after 89.18 seconds with return value 4  
Press any key to continue . . .
```