2/24

WEEK-9

BFS :-

```
#include <stdio.h>

#include <stdio.bool>

#define MAX_VERTICES 100

int n, j, visited[Max vertices];
int adj[Max_Vertices][Max_Vertices];
void dfs(int v) {
    visited[v] = 1;

for(int i=0; i<n; i++) {
    if(adj[v][i] && !visited[i] {
            dfs(i)
    }
}
}

int main()
int v;
```

```
printf("Enter the number of vertices:");
scanf("%d",&n);

for(i=0,i<n;i++){
    visited[i]=0;
}

printf("Enter graph data in matrix form\n");
for(i=0;i<n;i++)
    for(j=0;j<n;j++)
    scanf("%d",&adj[i][j]);

printf("Enter staring vertex:");
scanf("%d",&v);
dfs(v);

for(i=0,i<n;i++)
{
    if(!visited[i]){
    printf("\n The graph is not connected\n");
    return 0
    }
}
printf("The graph is connected\n");
return 0

}
```

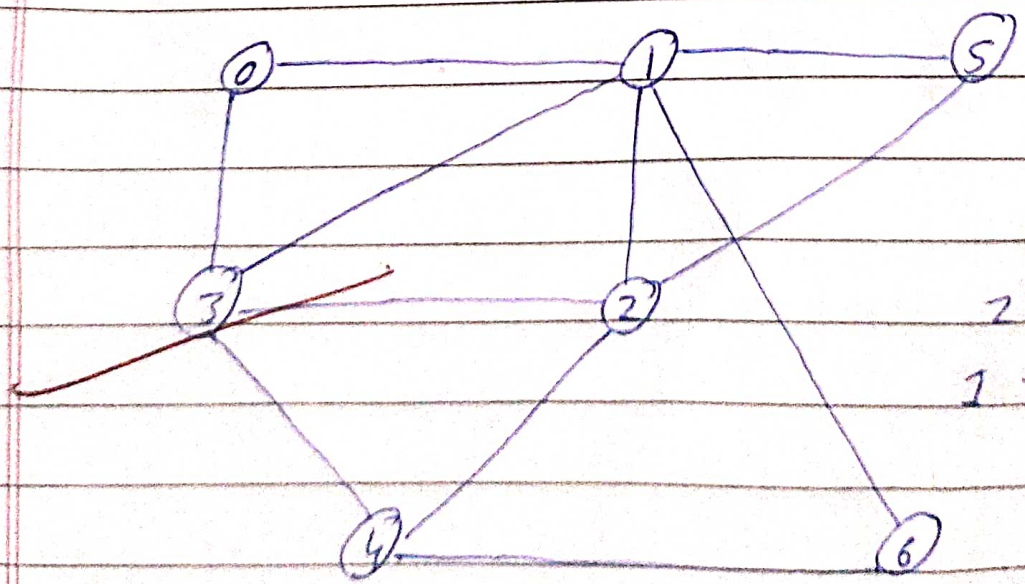
Output:-

Enter no of vertices: 5
Enter no of edges: 6

Enter graph data into matrix form:

0 1 0 1 0 0 0
1 0 1 1 0 1 1
0 1 0 1 1 0 0
0 0 1 1 0 0 1
0 1 1 0 0 0 0
0 1 0 0 1 0 0

Enter the starting vertex: 4
4 2 3 6 1 5 0



Visited:- 4, 2, 3, 6
4 → 2, 3, 6
2 → 1, 3, 4, 5, 6
1 → 0, 2, 3, 5, 6

WEEK-9

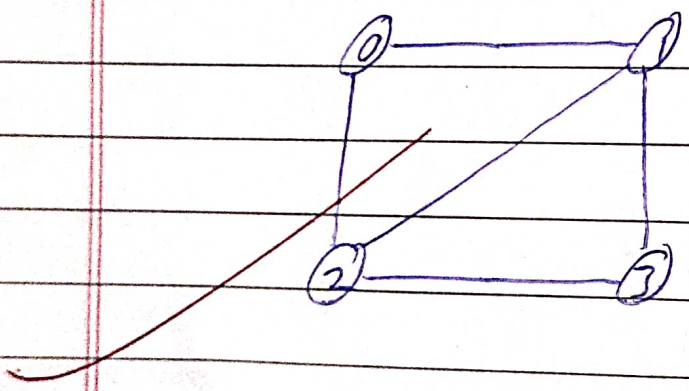
dfs:-

```
#include <stdio.h>
#include <stdlib.h>

#define MAX_VERTICES 10

int n, i, j, visited[MAX_VERTICES];
int adj[MAX_VERTICES][MAX_VERTICES];

void dfs(int v)
{
    visited[v]=1;
    for(int i=0; i<n; i++)
    {
        if(adj[v][i] && !visited[i])
        {
            dfs(i);
        }
    }
}
int main()
{
    int v;
    printf("Enter the number of vertices");
```

```
scanf("%d",&n);

for(i=0; i<n; i++)
 {
   visited[i]=0;
 }
printf("Enter graph data in matrix form\n");
   for(i=0; i<n; i++)
      for(j=0; j<n; j++)
         scanf("%d",&adj[i][j]);
printf("Enter the starting vertex:");
scanf("%d",&v);
dfs(v);

for(i=0; i<n; i++)
   if(!visited[i])
   {
      printf("\n the graph is not connected");
      exit(0);
   }

output:
```

Enter the no of vertices : 4
Enter the graph in matrix
0 1 1 0
1 0 0 1
1 0 0 0
0 1 0 0

Enter the start of vertex : 0
The graph is connected :



Sol:
22/2/24.