

Programs-1,2,5,6(previously we have completed)

/* 3. Write a C program depth first search (DFS) using array*/

```
#include<stdio.h>
#include<conio.h>
int a[20][20],reach[20],n;
void dfs(int v)
{
    int i;
    reach[v]=1;
    for(i=1;i<=n;i++)
        if(a[v][i] && !reach[i])
        {
            printf("n %d->%d",v,i);
            dfs(i);
        }
}
void main()
{
    int i,j,count=0;

    printf("\n Enter number of vertices:");
    scanf("%d",&n);
    for(i=1;i<=n;i++)
    {
        reach[i]=0;
        for(j=1;j<=n;j++)
            a[i][j]=0;
    }
    printf("\n Enter the adjacency matrix:");
    for(i=1;i<=n;i++)
        for(j=1;j<=n;j++)
            scanf("%d",&a[i][j]);
    dfs(1);
    printf("n");
    for(i=1;i<=n;i++)
    {
        if(reach[i])
            count++;
    }
    if(count==n)
```

```

    printf("\n Graph is connected");
else
    printf("\n Graph is not connected");
    getch();
}

```

OUTPUT:

Enter number of vertices:4

Enter the adjacency matrix:1

2
3
4
5
6
7
8
9
1
2
3
4
5
6
7

n 1->2n 2->3n 3->4n

Graph is connected

/*4. Write a C program breath first search (BFS) using array*/

```

#include<stdio.h>
#include<conio.h>
int a[20][20],q[20],visited[20],n,i,j,f=0,r=-1;
void bfs(int v) {
for (i=1;i<=n;i++)
    if(a[v][i] && !visited[i])
        q[++r]=i;
if(f<=r) {
    visited[q[f]]=1;
    bfs(q[f++]);
}
}

```

```

}
}
void main() {
int v;

printf("\n Enter the number of vertices:");
scanf("%d",&n);
for (i=1;i<=n;i++) {
q[i]=0;
visited[i]=0;
}
printf("\n Enter graph data in matrix form:\n");
for (i=1;i<=n;i++)
for (j=1;j<=n;j++)
scanf("%d",&a[i][j]);
printf("\n Enter the starting vertex:");
scanf("%d",&v);
bfs(v);
printf("\n The node which are reachable are:\n");
for (i=1;i<=n;i++)
if(visited[i])
printf("%d\t",i); else
printf("\n Bfs is not possible");
getch();
}

```

OUTPUT:

Enter the number of vertices:3

Enter graph data in matrix form:

1
2
3
4
5
6
7
8
9

Enter the starting vertex:1

The node which are reachable are:

1 2 3