

3. Write a C program depth-first search (DFS) using an array.

Code:

```
#include<stdio.h>
#include <stdlib.h>
int a[20][20],r[20],n;
void dfs(int v)
{
    int i;
    r[v]=1;
    for(i=1;i<=n;i++)
        if(a[v][i] && !r[i])
    {
        printf("%d->%d",v,i);
        dfs(i);
    }
}
void main()
{
    int i,j,c=0;
    printf("Enter number of vertices:");
    scanf("%d",&n);
    for(i=1;i<=n;i++)
    {
        r[i]=0;
        for(j=1;j<=n;j++)
            a[i][j]=0;
    }
    printf("\nEnter 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(r[i])
            c++;
    }
    if(c==n)
        printf("Graph is connected");
    else
        printf("Graph is not connected");
}
```

4. Write a C program breadth-first search (BFS) using an array.

Code:

```
#include<stdio.h>
#include<stdlib.h>
int array[20][20], q[20], vis[20], n, i, j, f = 0, r = -1;
void bfs(int v) {
    for(i = 1; i <= n; i++)
        if(array[v][i] && !vis[i])
            q[++r] = i;
    if(f <= r) {
        vis[q[f]] = 1;
        bfs(q[f++]);
    }
}
void main() {
    int v;
    printf("Enter the number of vertices:");
    scanf("%d", &n);
    for(i=1; i <= n; i++) {
        q[i] = 0;
        vis[i] = 0;
    }
    printf("\n Enter data in matrix form:");
    for(i=1; i<=n; i++) {
        for(j=1;j<=n;j++) {
            scanf("%d", &array[i][j]);
        }
    }
    printf("\n Enter the starting vertex:");
    scanf("%d", &v);
    bfs(v);
    printf("\n The node which is reachable is:\n");
    for(i=1; i <= n; i++) {
        if(vis[i])
            printf("%dt", i);
        else {
            printf("\n Bfs is not possible. All nodes are not reachable");
            break;
        }
    }
}
```