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

```
#include<stdio.h>
int G[10][10],n,visited;
void DFS(int i)
{
   int j;
   printf("%d",i);
   visited[i]=1;
   for(j=0;j<n;j++)
   {
      if(!visited[j]&&G[i][j]==1)
      {
         DFS(j);
      }
   }
}
void main()
{
   int i,j;
   printf("Enter the no.of vertices: ");
   scanf("%d",&n);
   printf("\nEnter the adjancey matrix of the graph: ");
   for(i=0;i<n;i++)
   {
      for(j=0;j<n;j++)
      {
         scanf("%d",&G[i][j]);
      }
   }
   for(i=0;i<n;i++)
   {
      visited[i]=0;
      DFS(0);
   }
}
```

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

```
#include<stdio.h>
int a[50][50],b[50],visited[50],n,i,j,x=0,y=-1;
void bfs(int z)
{
   for(i=0;i<=n;i++)
   {
      if(a[z][i]&&!visited[i])
      b[++y]=i;
      if(x<=y)
```

```
        {
            visited[b[y]]=1;
            bfs[b[y++]];
        }
    }
}
void main()
{
    int z;
    printf("Enter the no.of vertices: ");
    scanf("%d",&n);
    for(i=1;i<n;i++)
    {
        b[i]=0;
        visited[i]=0;
    }
    printf("Enter the graph data in the matrix form: ");
    for (i=1;i<=n;i++){
    for (j=1;j<=n;j++){
    scanf("%d",&a[i][j]);
    printf("Enter the starting vertex:");
    scanf("%d",&z);
    bfs(v);
    printf("The node which are reachable are:\n");
    for (i=1;i<=n;i++){
    if(visited[i]){
    printf("%d\t",i);}
    else{
    printf("Bfs is not possible");}
    }
    }
    }
}
}
```