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

```
#include <stdio.h>
#include <stdlib.h>
int source,V,E,time,visited[200],G[200][200];
void DFS(int i)
{
   int j;
   visited[i]=1;
   printf(" %d->",i+1);
   for(j=0;j<V;j++)
   {
      if(G[i][j]==1&&visited[j]==0)
         DFS(j);
   }
}
int main()
{
   int i,j,v1,v2;
   printf("\t\t\tGraphs\n");
   printf("Enter the no of edges:");
   scanf("%d",&E);
   printf("Enter the no of vertices:");
   scanf("%d",&V);
   for(i=0;i<V;i++)
   {
      for(j=0;j<V;j++)
```

```
            G[i][j]=0;
    }
    /*    creating edges :P    */
    for(i=0;i<E;i++)
    {
       printf("Enter the edges (format: V1 V2) : ");
       scanf("%d%d",&v1,&v2);
       G[v1-1][v2-1]=1;

    }

    for(i=0;i<V;i++)
    {
       for(j=0;j<V;j++)
           printf(" %d ",G[i][j]);
       printf("\n");
    }
    printf("Enter the source: ");
    scanf("%d",&source);
       DFS(source-1);
    return 0;
}
```

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

```
#include<stdio.h>
int G[120][120],q[120],visited[120],n,front = 1, rear = 0 ;
void bfs(int v)
{
   int i;
   visited[v] = 1;
 for(i=1;i<=n;i++)
  if(G[v][i] && !visited[i])
   q[++rear]=i;
   if(front <= rear)
    bfs(q[front++]);
 }


int main()
{
 int v,i,j;

 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",&G[i][j]);
 printf("\n Enter the starting vertex:");
```

```
 scanf("%d",&v);
 bfs(v);
 printf("\n The nodes which are reachable are:\n");
 for(i=1;i<=n;i++)
  if(visited[i])
   printf("%d\t",i);
  else
   printf("\n %d is not reachable",i);

return 0;
}
```