1.

3. write a program depth first searching using array.

```
#include <stdio.h>
#include <conio.h>
int a 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);
    }
}
int i, j, count = 0;
printf ("\n Enter the no. of vertices");
scanf ("%d", &n");
for (i=1; i<=n; i++) {
    reach[i] = 0;
    for (j=1; i<=n; j++)
        a[i][j] = 0;
}
printf ("\n Enter the adjancey matrix: \n");
for (i = 1; i<=n; i++)
for (j = 1; j<=n, j++)
    scanf ("%d", &a[i][i]);
```

```
dfs (1);
printf ("\n");
for (i=1; i<=n; i++) {
    if (reach[i])
        count++;
}
if (count==n)
    printf ("\n matrix is connected");
else
    printf ("\n matrix is not connected");
getch();
}
```

~~Enter number of vertices : 2~~

~~Enter~~ the adjacency matrix;

output

Enter the number of vertices : 2

Enter the adjacency matrix:

1
1
1
0

1 → 2

matrix is connected.

program finished with exit ~~code~~ 215

Code 215.

④ wrote c program

```
#include <stdio.h>
#include <conio.h>
int a[20][20], q[20], visited[20], n, i, 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 graph data in
```

```
printf ("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 ", i); else
    printf ("Bfs is not possible");
  getch ();
}
```

output :

Enter the number of vertices : 3

Enter the graph data in matrix form :

```
1
0
0
0
1
0
0
0
1
```

Enter the starting vertex 2.

The node which are reasonable are

Bfs is not possible 2.

Bfs is not possible