



The image shows a screenshot of a C code editor. The editor has a toolbar at the top with buttons for Run, Debug, Stop, Share, Save, and Beautify. The language is set to C. The code is for a Breadth-First Search (BFS) algorithm. It includes standard headers, defines constants, and implements functions for queue operations and BFS.

```
9 #include<stdio.h>
10 #define size 20
11 #define true 1
12 #define false 0
13 int a[20][20],reach[20],n;
14 int queue[size],visit[20],rear=-1,front=0;
15 int n,s,adj[20][20],flag=0;
16 void insertq(int v)
17 {
18     queue[++rear]=v;
19 }
20
21 int deleteq()
22 {
23     return(queue[front++]);
24 }
25
26 int qempty()
27 {
28
29     if(rear<front)
30         return 1;
31
32     else
33         return 0;
34 }
35
36 void bfs(int v)
37 {
38
```

```
main.c
36 void bfs(int v)
37 {
38     int w;
39     visit[v]=1;
40     insertq(v);
41
42     while(!qempty())
43     {
44         v=deleteq();
45         for(w=1;w<=n;w++)
46
47             if((adj[v][w]==1) && (visit[w]==0))
48             {
49                 visit[w]=1;
50                 flag=1;
51                 printf("v%d\t",w);
52                 insertq(w);
53             }
54     }
55 }
56 void dfs(int v){
57     int i;
58     reach[v]=1;
59     for(i=1;i<=n;i++)
60     {
61         if(a[v][i]&&!reach[i]){
62             printf("\n%d->%d",v,i);
63             dfs(i);
64         }
65     }
```


The screenshot shows a web browser window with the URL `onlinegdb.com/online_c_compiler`. The browser's address bar and tabs are visible. Below the browser window is a toolbar with buttons for `Run`, `Debug`, `Stop`, `Share`, `Save`, `Beautify`, and a download icon. The language is set to `C`. The main editor area shows a C program named `main.c` with the following code:

```
66 void main()
67 {
68     int ch,v,w,i,j,count=0;
69     printf("1.Print all the nodes reachable from a given starting node in a digraph using BFS method\n2. Check whether a given
70     printf("Your choice:");
71     scanf("%d",&ch);
72     do{
73         switch(ch){
74             case 1:printf("Enter the no.of vertices:\n");
75             scanf("%d",&n);
76             printf("Enter adjacency matrix:\n");
77             for(v=1;v<=n;v++)
78             {
79                 for(w=1;w<=n;w++)
80                     scanf("%d",&adj[v][w]);
81             }
82             printf("Enter the start vertex:");
83             scanf("%d",&s);
84             printf("Reachability of vertex %d\n",s);
85             for(v=1;v<=n;v++)
86                 visit[v]=0;
87
88             bfs(s);
89
90             if(flag==0)
91             {
92                 printf("No path found!!\n");
93             }
94             break;
95 }
```

The Windows taskbar is visible at the bottom, showing the search bar and several application icons. The system clock in the bottom right corner displays the time `18:13` and the date `10-05-2021`.

```
main.c
92     printf("No path found!!\n");
93 }
94 break;
95 case 2: printf("\nEnter no of vertices :");
96 scanf("%d",&n);
97 for(i=1;i<=n;i++)
98     for(j=1;j<=n;j++){
99         reach[i]=0;
100         a[i][j]=0;
101     }
102 printf("\nEnter adjacency matrix : \n");
103 for(i=1;i<=n;i++)
104     for(j=1;j<=n;j++)
105         scanf("%d",&a[i][j]);
106 dfs(1);
107 for(i=1;i<=n;i++)
108     if(reach[i])
109         count++;
110 if(count==n)
111     printf("\nGraph is connected.");
112 else
113     printf("\nGraph is disconnected.");
114 break;
115 }
116 printf("\nYour Choice:");
117 scanf("%d",&ch);
118 }while(ch!=3);
119 }
120 }
```



```
onlinegdb.com/online_c_compiler
input
1.Print all the nodes reachable from a given starting node in a digraph using BFS method
2. Check whether a given graph is connected or not using DFS method.
3.Exit
Your choice:1
Enter the no.of vertices:
4
Enter adjacency matrix:
0 1 1 0
0 0 0 1
0 0 1 0
1 0 0 1
Enter the start vertex:1
Reachability of vertex 1
v2      v3      v4
Your Choice:2

Enter no of vertices :4
Enter adjacency matrix :
0 1 1 0
0 0 0 1
1 0 0 0
0 0 0 0

1->2
2->4
1->3
Graph is connected.
```

```
input
v2      v3      v4
Your Choice:2

Enter no of vertices :4

Enter adjacency matrix :
0 1 1 0
0 0 0 1
1 0 0 0
0 0 0 0

1->2
2->4
1->3
Graph is connected.
Your Choice:2

Enter no of vertices :4

Enter adjacency matrix :
0 1 0 0
0 0 0 0
0 0 0 1
0 0 0 0

1->2
Graph is disconnected.
Your Choice:
```