BFS CODE-->

```
#include <stdio.h>
#include <stdlib.h>
#define MAX 10
void BFS(int vertex);
int graph[MAX][MAX], visited[MAX], total;
int main()
{
   int i, j;
   printf("\nEnter the total number of vertices in the graph (max %d): ", MAX);
   scanf("%d", &total);
   if (total > MAX || total <= 0)
   {
      printf("Invalid number of vertices.\n");
      return 1;
   }

   printf("\nEnter the adjacency matrix:\n");
   for (i = 0; i < total; i++)
   {
      for (j = 0; j < total; j++)
      {
         scanf("%d", &graph[i][j]);
      }
```

```
    }

    for (i = 0; i < total; i++)
    {
        visited[i] = 0;
    }
    printf("\nBFS traversal is:\n");
    BFS(0);
    return 0;
}
void BFS(int vertex)
{
    int queue[MAX], front = -1, rear = -1, j;

    rear++;
    queue[rear] = vertex;
    visited[vertex] = 1;
    while (front != rear)
    {
        front++;
        vertex = queue[front];
        printf("%d\t", vertex);

        for (j = 0; j < total; j++)
```

```
    {
      if (!visited[j] && graph[vertex][j] == 1)
      {
        rear++;

        queue[rear] = j;
        visited[j] = 1;
      }
    }
  }
}
```

```
/tmp/NmOs0kFkLu.o

Enter the total number of vertices in the graph (max 10): 4

Enter the adjacency matrix:
0 1 1 0 0
1 0 0 1 0
1 0 0 0 1
0 1 0 0 10 1 1 0 0

1 0 0 1 0

1 0 0 0 1

0 1 0 0 1

BFS traversal is:
0   1   2

=== Code Execution Successful ===
```

DFS CODE-->

```
#include <stdio.h>

#include <stdlib.h>

int graph[10][10], visited[10], total, arr[30];

static int k = 0, count = 0;

void DFS(int vertex);

int main()

{

   int i, j;
```

```
    printf("\nEnter the total number of vertices in the graph: ");
    scanf("%d", &total);

    printf("\nEnter the adjacency matrix:\n");
    for (i = 0; i < total; i++)
    {
        for (j = 0; j < total; j++)
        {
            scanf("%d", &graph[i][j]);
        }
    }

    for (i = 0; i < total; i++)
    {
        visited[i] = 0;
    }

    printf("\nDFS traversal is:\n");
    DFS(0);
    return 0;
}
void DFS(int vertex)
{
    int j;

    printf("%d\t", vertex);
```

```
    visited[vertex] = 1;


    for (j = 0; j < total; j++)
    {
        if (!visited[j] && graph[vertex][j] == 1)
        {
            DFS(j);
        }
    }
}
```

```
/tmp/mNdvqzyIhD.o

Enter the total number of vertices in the graph: 4

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

0 1 0 00 1 1 0

1 0 0 1

1 0 0 0



0 1 0 0

DFS traversal is:
0   1   3   2

=== Code Execution Successful ===
```