

Code:

```
#include <conio.h>
#include <stdio.h>
#include <stdlib.h>
int a[4][4];
int visited[4];
int r = 4;
void DFS(int i)
{
    visited[i] = 1;
    printf("%c ", i + 65);
    for (int j = 0; j < r; j++)
    {
        if (a[i][j] == 1 && visited[j] == 0)
        {
            DFS(j);
        }
    }
}
void BFS(int start)
{
    int queue[10];
    int front = 0, rear = 0;
    visited[start] = 1;
    queue[rear++] = start;
    while (front < rear)
    {
        int node = queue[front++];
        printf("%c ", node + 65);

        for (int j = 0; j < r; j++)
        {
            if (a[node][j] == 1 && visited[j] == 0)
```

```

        {
            visited[j] = 1;
            queue[rear++] = j;
        }
    }
}

void main()
{
    int i, j;
    for (i = 0; i < r; i++)
    {
        for (j = 0; j < r; j++)
        {
            printf("Input element %d,%d: ", i, j);
            scanf("%d", &a[i][j]);
        }
    }

    printf("\nDFS Traversal:\n");
    for (i = 0; i < r; i++)
        visited[i] = 0;
    DFS(0);
    printf("\n\nBFS Traversal:\n");
    for (i = 0; i < r; i++)
        visited[i] = 0;
    BFS(0);
    printf("\n");
}

```

Output:

```
E:\piyu\Computer Engg\Sem 3\DSA\Binary Tree>dfsbsf
Input element 0,0: 0
Input element 0,1: 1
Input element 0,2: 1
Input element 0,3: 0
Input element 1,0: 0
Input element 1,1: 0
Input element 1,2: 0
Input element 1,3: 1
Input element 2,0: 0
Input element 2,1: 0
Input element 2,2: 0
Input element 2,3: 1
Input element 3,0: 0
Input element 3,1: 0
Input element 3,2: 0
Input element 3,3: 0

DFS Traversal:
A B D C

BFS Traversal:
A B C D
```