```
scanf ("%d", &n)
printf ("\n enter adjency matrix");
for (int i = 0 ; i<n ; i++){
        for (j = 0; j<n ; j++){
                scanf( "%d", &a[i][j]);

        }

}
for (source = 0; source<n; source++){
        bfs (a, n, source);

}
return 0;

}
```

```
Enter no. of vertices : 4

Enter adjency matrix:  0  1  1  0
1  0  1  0
1  1  0  1
0  0  1  0

The nodes visited from 0 :  0  1  2  3
The nodes visited from 1 :  1  0  2  3
The nodes visited from 2 :  2  0  1  3
The nodes visited from 3 :  3  2  0  1
```

dfs :

```
#include <stdio.h>
#include <stdlib.h>

void dfs( int a[10][10], int n, int u,
          int visited[]){
```

```
        int v ;
        printf ( " %d ", u) ;
        visited [u] = 1 ;
        for ( v=0; v<n ;v++ ) {
                if ( a[u][v] == 1 && !visited [v] ) {
                        dfs ( a,n, v, visited ) ;
                }
        }
}

int  main() {
        int n, a[10][10] , source , i, j;
        int visited [10] = {0};
        printf (  " Enter the no. of vertices : ");
        scanf ( "%d", &n );

        printf ( " Enter adjacency matrix." );
         for ( i=0; i<n; i++) {
                for ( j=0 ;j<n ;j++) {
                        scanf ( "%d", &a[i][j] ) ;
                }
        }
        printf ( "DFS traversal' );
        for ( source = 0 ; source <n ; source ++ ){
                if ( !visited [source] ) {
                        dfs  (a, u, source, visited);
                }
        }
}
```

utput:

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

DFS traversal :  0 1 2 3
```

```
int findBottomLeftValue (struct ~~node~~ TreeNode* root
      int ans = root → val;
      int maxLev = 0;
     f (root, 0, & ans, & maxlevel);
      return ans;
  }
```

26.02.24