

~\Downloads\df.c

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

typedef struct
{
    int *a;
    int top;
}stack;

void input(int **, int);
void initialize(stack *, int);
void push(stack *, int);
void pop(stack *);
void dfs(int **, int n);

void main()
{
    int i, l;
    printf("\nEnter the number of vertices: ");
    scanf("%d", &l);
    int **g = (int **)calloc(l, sizeof(int *));
    for(i = 0; i < l; i++)
        g[i] = (int *)calloc(l, sizeof(int));

    input(g, l);
    dfs(g, l);
}

void input(int **m, int n)
{
    int o, d, c;
    while(1)
    {
        printf("\nEnter origin and destination vertices(with space in between): ");
        scanf("%d %d", &o, &d);
        m[o][d] = m[d][o] = 1;
        printf("\nPress 1 to continue and 0 to exit: ");
        scanf("%d", &c);
        if(c == 0)
            break;
    }
}

void initialize(stack *s, int n)
{
    s->a = (int *)calloc(n, sizeof(int));
    s->top = -1;
}

void push(stack *s, int e)
{
    s->top++;
    s->a[s->top] = e;
}
```

```

void pop(stack *s)
{
    s->top--;
}

void dfs(int **m, int n)
{
    int i, v;
    stack s;
    initialize(&s, n);
    int *st = (int *)calloc(n, sizeof(int));
    push(&s, 0);
    st[0] = 1;
    printf("DFS Traversal of the graph: A ");
    while(s.top != -1)
    {
        v = s.a[s.top];
        for(i = 0; i < n; i++)
        {
            if(st[i] == 0 && m[v][i] != 0)
            {
                push(&s, i);
                st[i] = 1;
                printf("%c ", i + 65);
                break;
            }
        }

        if(i == n)
            pop(&s);
    }

    printf("\n");
}

```