

DFS

```

from collections import defaultdict

class Graph:

    def __init__(self):

        self.graph = defaultdict(list)

    def addEdge(self, u, v):

        self.graph[u].append(v)

    def dfsUtil(self, v, visited):

        visited.add(v)

        print(v, end=' ')

        for neighbour in self.graph[v]:

            if neighbour not in visited:

                self.dfsUtil(neighbour, visited)

    def DFS(self, v):

        visited = set()

        self.dfsUtil(v, visited)

if __name__ == "__main__":

    n = int(input('Enter the number of vertices : '))

    print('Enter the adjacency matrix : ')

    graph = [list(map(int, input().split())) for _ in range(n)]

    g = Graph()

    for i in range(n):

        for j in range(n):

            if graph[i][j] == 1:

                g.addEdge(i, j)

    start_vertex = int(input(f'Enter the starting vertex(0 to {n}-1)'))

    print(f'Following is Depth First Search Traversal starting from vertex {start_vertex}')

    g.DFS(start_vertex)

```



