**AI Assignment 1**

**Code:**

def dfs(graph, start\_node):

visited = set()

def dfs\_util(node):

if node not in visited:

print(node)

visited.add(node)

for neighbour in graph[node]:

dfs\_util(neighbour)

dfs\_util(start\_node)

def bfs(graph, start\_node):

visited = set()

queue = []

visited.add(start\_node)

queue.append(start\_node)

while queue:

current\_node = queue.pop(0)

print(current\_node, end=" ")

for neighbour in graph[current\_node]:

if neighbour not in visited:

visited.add(neighbour)

queue.append(neighbour)

graph = {}

num\_edges = int(input("Enter the number of edges in the graph: "))

print("Enter the source & destination ofthe edges separated by space:")

for \_ in range(num\_edges):

edge = input().split()

source, destination = edge

if source not in graph:

graph[source] = []

if destination not in graph:

graph[destination] = []

graph[source].append(destination)

start\_node\_dfs = input("Enter the starting node for DFS: ")

print("Depth-First Search:")

dfs(graph, start\_node\_dfs)

start\_node\_bfs = input("Enter the starting node for BFS: ")

print("\nBreadth-First Search:")

bfs(graph, start\_node\_bfs)

**Output:**

Enter the number of edges in the graph: 6

Enter the source & destination of the edges separated by space:

5 3

5 7

3 2

3 4

7 8

4 8

Enter the starting node for DFS: 5

Depth-First Search:

5

3

2

4

8

7

Enter the starting node for BFS: 5

Breadth-First Search:

5 3 7 2 4 8