EXPERIMENT NO: 4 **A N V SREEVISHNU**
DATE: 26/02/2021 **RA1811003010333**

IMPLEMENTATION OF BFS & DFS

Aim: To implement BFS & DFS in AI using Python.

Procedure/Algorithm:

- First, Create a graph with number of nodes of your choice.
- Traverse through the graph with moving towards left.
- If a node is not visited pop out into the queue and print.
- If visited the node moves on and traverses to its neighbour depending on bfs or dfs it would traverse accordingly.
- Once all the nodes have been traversed, the code ends printing out all the nodes.

Code:

BFS:

```
graph = {

  'A' : ['B'],

  'B' : ['E', 'C'],

  'C' : ['D'],

  'D' : [],
```

```
 'E' : ['F'],

  'F' : []

}




visited = []

queue = []




def bfs(visited, graph, node):

    visited.append(node)

    queue.append(node)

    while queue:

        s = queue.pop(0)

        print (s, end = " ")
```

```
        for neighbour in graph[s]:

            if neighbour not in visited:

                visited.append(neighbour)

                queue.append(neighbour)



bfs(visited, graph, 'A')
```

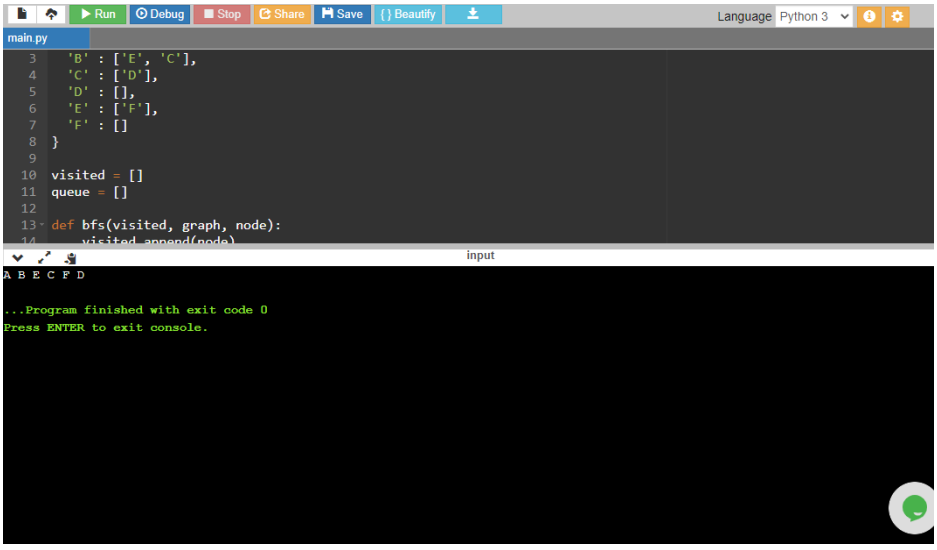
DFS:

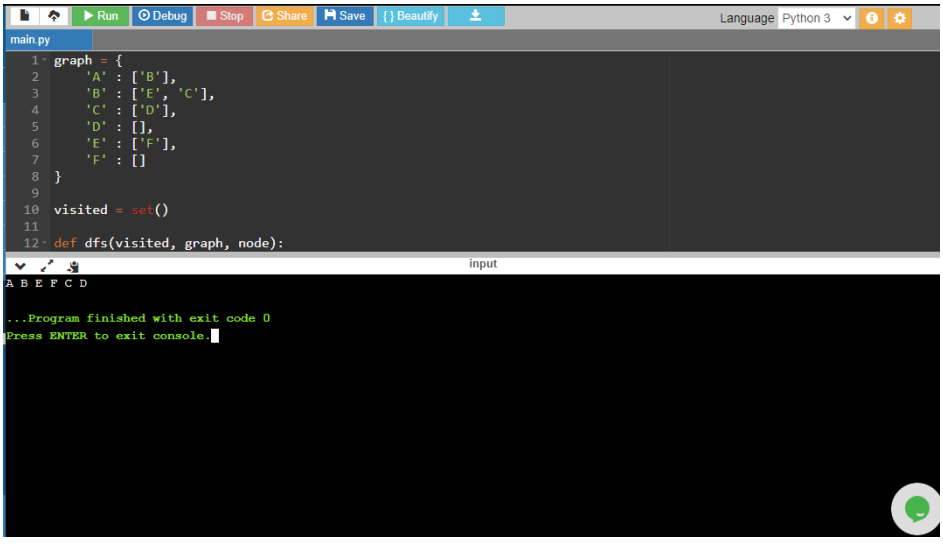
```
graph = {

   'A' : ['B'],

   'B' : ['E', 'C'],

   'C' : ['D'],

   'D' : [],
```

```
    'E' : ['F'],

    'F' : []

}


visited = set()


def dfs(visited, graph, node):

    if node not in visited:

        print (node , end=' ')

        visited.add(node)

        for neighbour in graph[node]:

            dfs(visited, graph, neighbour)
```

dfs(visited, graph, 'A')

Output:





Result: Thus,the implementation of BFS & DFS in AI using Python has been successfully done.