

Trace the following breadth first algorithm

```
1 def breadth_first_traversal(node)
2     queue = []
3     queue.append(node)
4     visited.append(node)

5     while queue is not empty
6         node = queue.pop(0)
7         print (node, end = " ")
8         for i in graph[node]:
9             if i not in visited
10                queue.append(i)
11                visited.append(i)

12 graph={'A':['D','B'],
13        'B':['A','E','C','F'],
14        'C': ['B','F'],
15        'D': ['A','E'],
16        'E':['D','B'],
17        'F':['B','C']}
```


Trace the following depth first algorithm

```
1 def depth_first_traversal(node):
2     visited.append(node)
3     for i in graph[node]:
4         if i not in visited:
5             depth_first_traversal(i)
6     return

7 graph={"A":["D","B"],
        "B":["A","E","C","F"],
        "C":["B","F"],
        "D":["A","E"],
        "E":["D","B"],
        "F":["B","C"]}

8 visited = []
9 depth_first_traversal("A")
10 print(visited)
```

© 2025 Exam Papers Practice. All Rights Reserved.