

```
def BFS( graph, start):
    visited = []
    queue = []

    visited.append(start)
    queue.append(start)

    while queue:
        print('-----')
        print(f"Queue : { queue }")
        s = queue.pop(0)
        print( s , end = ' ')
        print()

        for node in graph[s]:
            if node not in visited:
                print(f"Node = { node }")
                visited.append(node)
                queue.append(node)

graph = { 'A':['B','C'] , 'B':['A','D','E'] , 'C':['A','F','G'] , 'D':['B'] , 'E':['B'] , 'F':['C'] , 'G':['C'] }
BFS( graph, 'A')
```

OUTPUT

```
-----
Queue : ['A']
A
Node = B
Node = C
-----
Queue : ['B', 'C']
B
Node = D
Node = E
-----
Queue : ['C', 'D', 'E']
C
Node = F
Node = G
-----
Queue : ['D', 'E', 'F', 'G']
D
-----
Queue : ['E', 'F', 'G']
E
-----
Queue : ['F', 'G']
F
-----
Queue : ['G']
G
```

```
def BFS( graph, start):
    visited = []
    queue = []

    visited.append(start)
    queue.append(start)

    while queue:
        s = queue.pop(0)
        print( s , end = ' ')

        for node in graph[s]:
            if node not in visited:
                visited.append(node)
                queue.append(node)

graph = { 'A':['B','C'] , 'B':['A','D','E'] , 'C':['A','F','G'] , 'D':['B'] , 'E':['B'] , 'F':['C'] , 'G':['C'] }
print("BFS : ", end= " ")
BFS( graph, 'A')
```

OUTPUT

BFS : A B C D E F G