Program for breadthfirst search algorithm

In [11]:

```
from collections import deque
graph={
    'A':['B','C'],
    'B':['D','E','F'],
    'C':['G'],
    'D':[],
    'E':[],
    'F':['H'],
    'G':['I'],
    'H':[],
    'I':[],
}
def bfs(graph,node):
    visited=[]
    queue=deque()

    visited.append(node)
    queue.append(node)

    while queue:
        s=queue.popleft()
        print(s, end = " ")

        for n in graph[s]:
            if n not in visited:
                visited.append(n)
                queue.append(n)
def main():
    bfs(graph,'A')

main()
```

```
A B C D E F G H I
```

VIVA

In [10]:

```
from collections import deque
graph={
    '1':['2','3','4'],
    '2':[],
    '3':['5','6'],
    '4':['7'],
    '5':['10'],
    '6':['8','9'],
    '7':[],
    '8':['11','12'],
    '9':[],
    '10':[],
    '11':[],
    '12':[],
}
def bfs(graph,node):
    visited=[]
    queue=deque()

    visited.append(node)
    queue.append(node)

    while queue:
        s=queue.popleft()
        print(s, end = " ")

        for n in graph[s]:
            if n not in visited:
                visited.append(n)
                queue.append(n)
def main():
    bfs(graph,'1')

main()
```

```
1 2 3 4 5 6 7 10 8 9 11 12
```

In [ ]: