Practical 1:

#BFS Algorithm

```
adj={1:[2,3],2:[1],3:[1,4],4:[3]}

def BFS(adj,startnode,numofnodes):
    visi=[0]*(numofnodes+1)
    que=[]
    visi[startnode]=1
    que.append(startnode)
    while(len(que)!=0):
        adjele=que.pop(0)
        print(adjele,end=" ")
        getlist=adj.get(adjele)
        for i in getlist:
            if(visi[i]==0):
                visi[i]=1
                que.append(i)

if __name__=='__main__':
    BFS(adj,1,4)
```

Output:

```
(base) ubuntu@ubuntu-OptiPlex-3090:~$ python BFS.py
1 2 3 4
```

#BFS Algorithm

```
adj={1:[2,3],2:[1,5],3:[1,4],4:[3],5:[2]}
visi=[0]*(6)

def DFS(adj,startnode):
    visi[startnode]=1
    print(startnode,end=" ")

    for i in adj.get(startnode):
        if(visi[i]==0):
            DFS(adj,i)


if __name__=='__main__':
    DFS(adj,1)
```

Output:

```
(base) ubuntu@ubuntu-OptiPlex-3090:~$ python DFS.py
1 2 5 3 4
```