tree = {

'A':[('B',12),('C',4)],

'B':[('D',7),('E',3)],

'C':[('F',8),('G',2)],

'D':[],

'E':[('H',0)],

'F':[('H',0)],

'G':[('H',0)],

'H':[]

}

start = input("Enter Start Node : ")

goal = input("Enter Goal Node : ")

def bestfirstsearch(start,goal,tree,open=[],close=[]) :

if start == goal :

print(start)

return

if start not in close :

print(start,end="->")

close.append(start)

neighbour=tree[start]

for i in neighbour:

if i[0][0] not in open :

open.append(i)

open.sort(key=lambda x : x[1])

if open[0][0] == goal :

print(open[0][0],end="")

else:

node=open[0]

open.remove(node)

bestfirstsearch(node[0],goal,tree,open,close)

if goal in tree :

print("Path : ",end="")

bestfirstsearch(start,goal,tree,open=[],close=[])

else :

print("Goal does not exist ")



