PROGRAM :

```
graph=[
    ['A','H',7,0],['A','B',1,3],['A','C',2,4],
    ['B','D',4,2],['B','E',6,6],['C','F',3,3],['C','G',2,1],
    ['D','E',7,6],['F','H',1,0],['G','H',2,0],['D','H',5,0]
]
start=input("Enter start node : ")
goal=input("Enter goal node : ")
tmp=[]
tmp1=[]
for i in graph:
    tmp.append(i[0])
    tmp1.append(i[1])
nodes=set(tmp).union(set(tmp1))
cost=dict()
path=dict()
for i in nodes:
    cost[i]=9999
    path[i]=''
open=set()
close=set()
open.add(start)
cost[start]=0
path[start]=start

def Astar(graph,open,close,cost,curr_node):
    if curr_node in open:
        open.remove(curr_node)
    close.add(curr_node)
    for i in graph:
        if(i[0]==curr_node and cost[i[0]]+i[2]+i[3]<cost[i[1]]):
            open.add(i[1])
            cost[i[1]]=cost[i[0]]+i[2]+i[3]
            path[i[1]]=path[i[0]]+'->'+i[1]
    cost[curr_node]=9999
    small=min(cost,key=cost.get) #storing B right now
    if small not in close:
        Astar(graph, open, close, cost, small)


Astar(graph, open, close, cost, start)
print("Path is:",path[goal])
path1  = path[goal].replace('->','')
list1  = path1[1:-1]

sum = 0
for elements in list1:
    for j in graph:
        if j[1] == elements:
            sum  = sum + j[3]

print(f"Cost = {cost[goal] - sum}")
```

OUTPUT :

```
Enter the start node : A
Enter the goal node  : H
path is  A->C->G->H
Cost = 6
```