#in this code we will consider explored = 1,visited and didnot explored = 0, unvisited = -1

import heapq

import math

def function\_1():

temp = open("input\_files.txt", "r")

adj\_lst = {}

h\_val = {}

for i in temp.readlines():

i = i.split()

h\_val[i[0]] = int(i[1])

if i[0] not in adj\_lst:

adj\_lst[i[0]] = []

for j in range(2, len(i), 2):

adj\_lst[i[0]].append((i[j], int(i[j+1])))

temp.close()

return adj\_lst, h\_val

#----------------

def a\_star\_algo(adj\_lst, h\_val, s\_node, e\_node) :

save\_dictionary = {}

store = []

for k in adj\_lst.keys():

save\_dictionary[k] = -1

heapq.heappush(store, (0+h\_val[s\_node], s\_node, [s\_node], 0))

save\_dictionary[s\_node] = 0

while len(store) >0 :

flag = heapq.heappop(store)

for x, y in adj\_lst[flag[1]]:

if save\_dictionary[x] != 1:

heapq.heappush(store, ((flag[3]+y)+h\_val[x], x, flag[2]+[x], (flag[3]+y)))

save\_dictionary[x] = 0

save\_dictionary[flag[1]] = 1

if flag[1] == e\_node:

return flag[2], flag[3]

return [], -1

#----------------------

if \_\_name\_\_ == "\_\_main\_\_" :

adj\_lst, h\_val = function\_1()

s\_node = input("enter your start node: ")

e\_node = input("enter your destination node: ")

route, final\_cost = a\_star\_algo (adj\_lst, h\_val, s\_node, e\_node)

if final\_cost != -1:

print('finally the route is:', route[0], end = "")

for z in route[1:]:

print(" ->", z, end = "")

print()

print("Total distance:", final\_cost, "km")

if final\_cost == -1:

print("NO PATH FOUND")