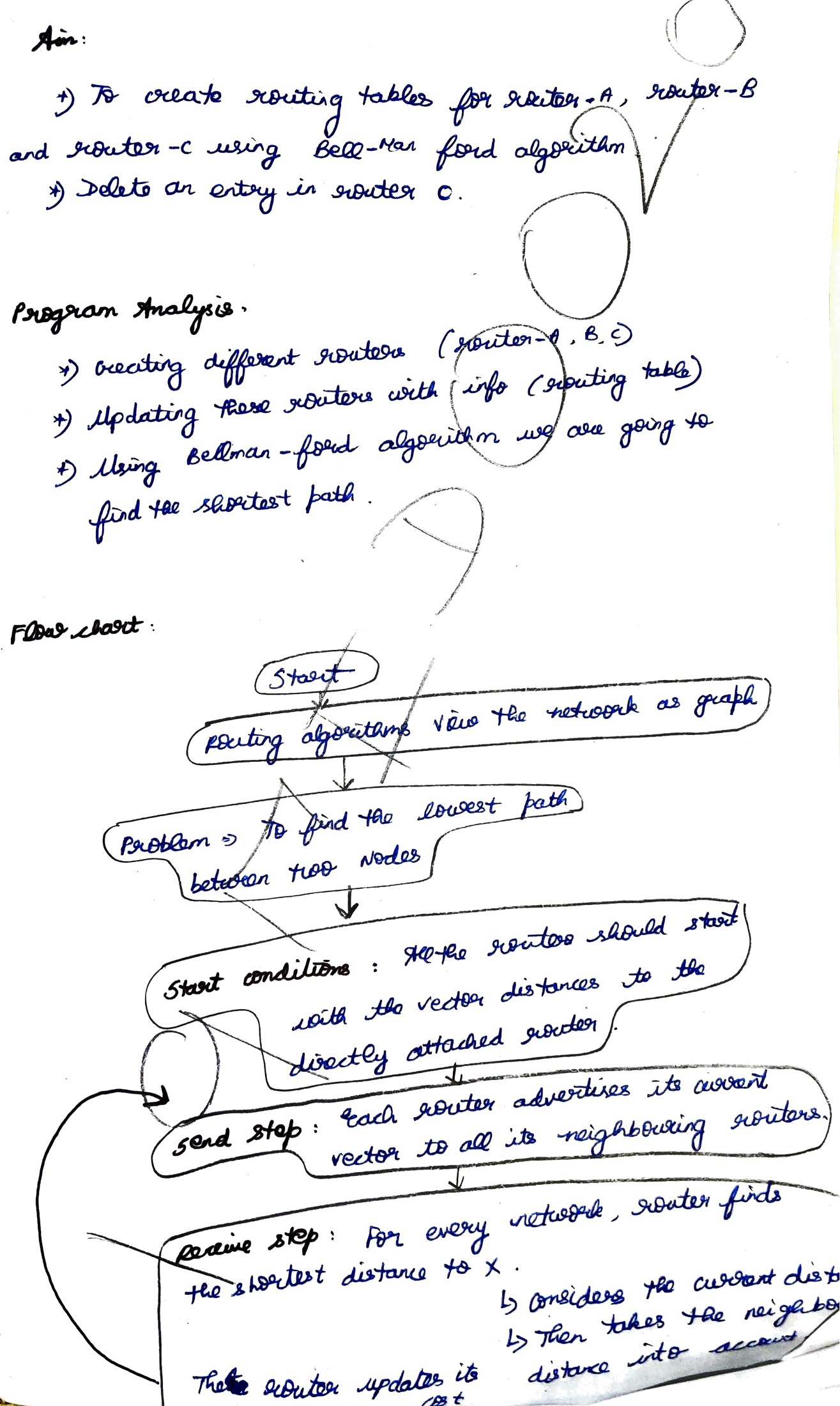
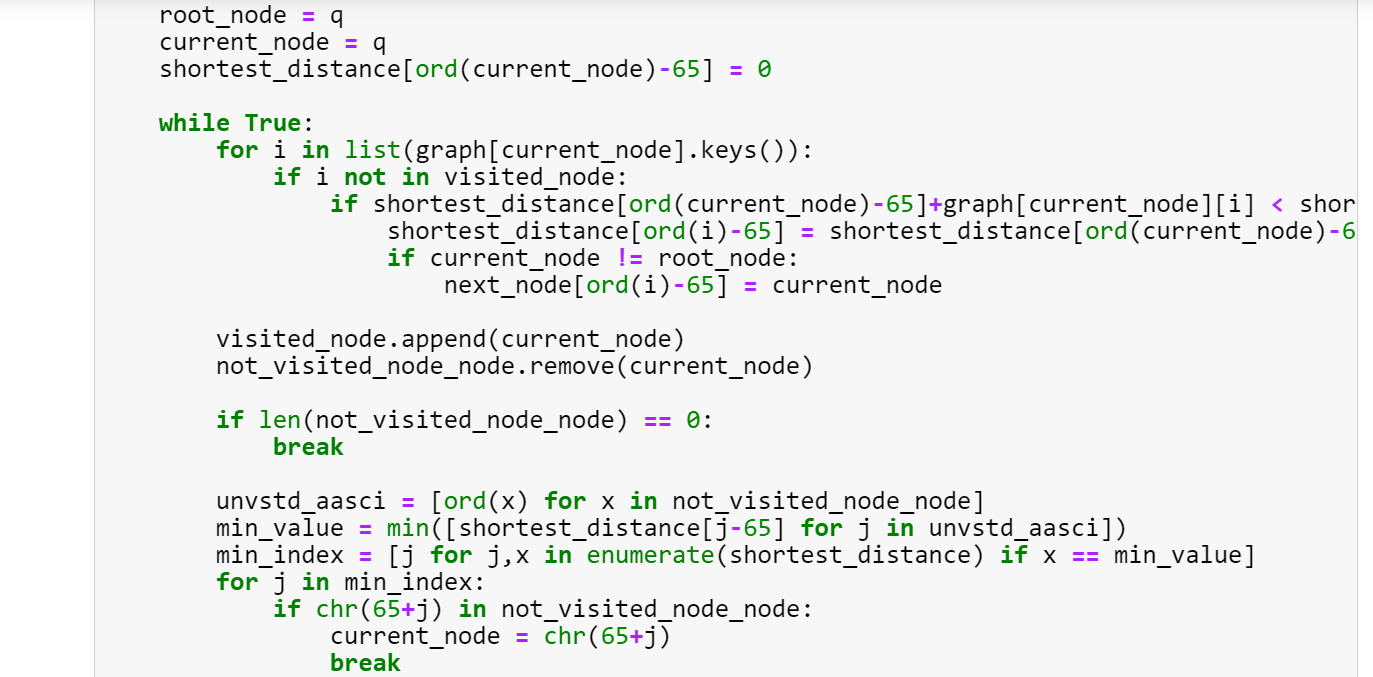
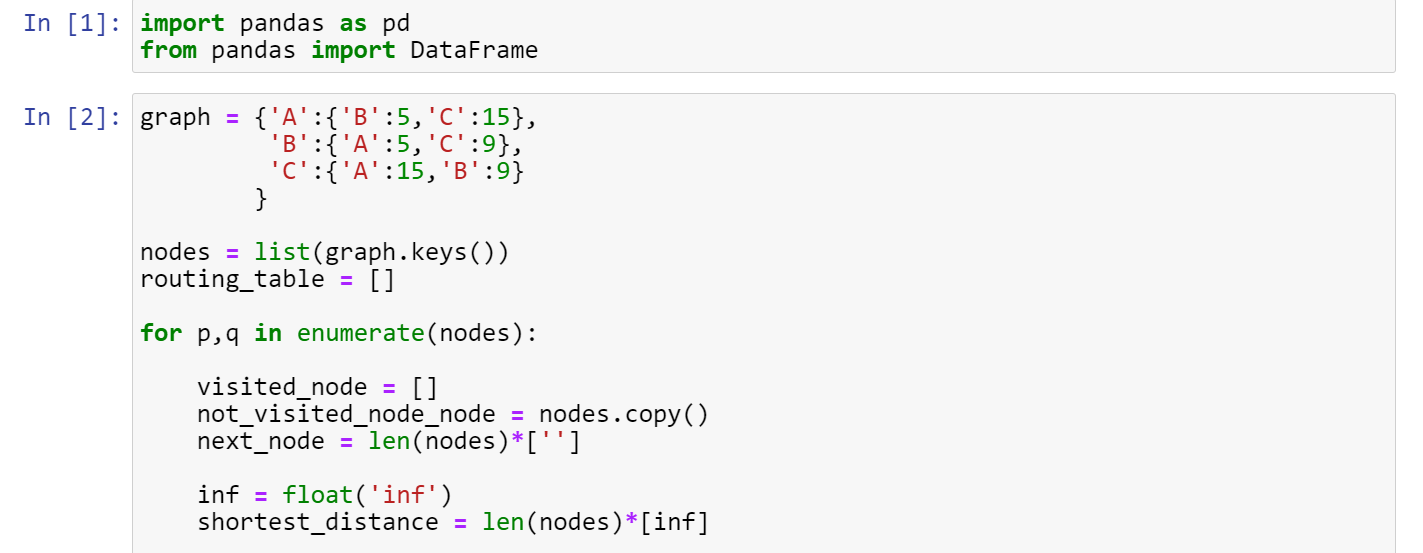
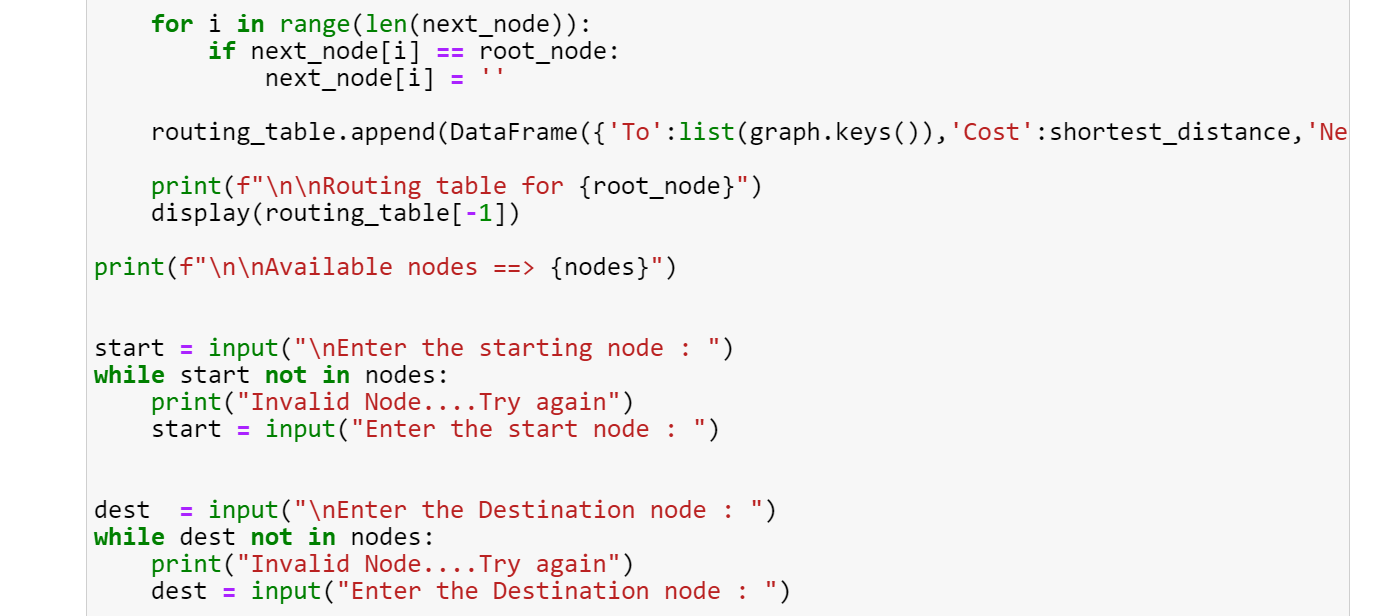
**DCN Lab-Fat Prashanth.S   
 (19MID0020)**

**Question-1**



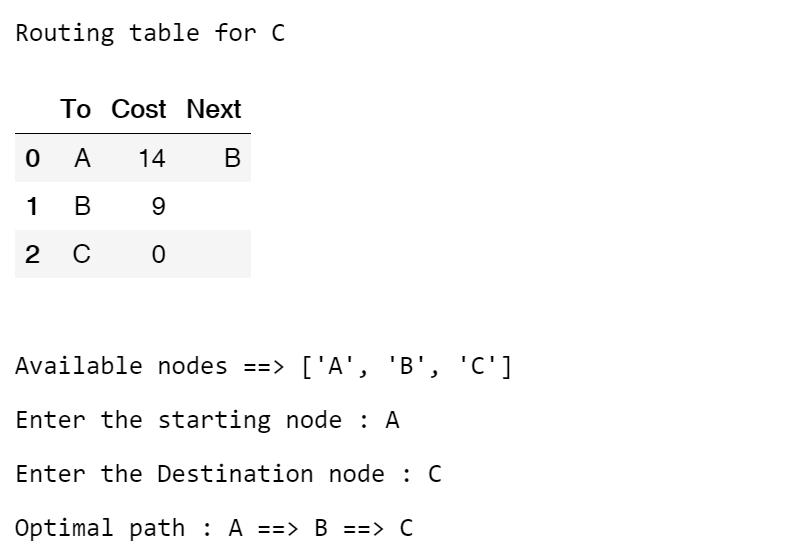
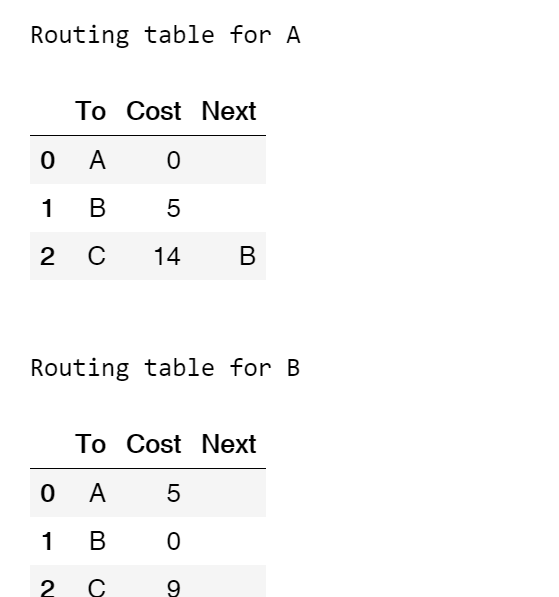
**Code Snippet**



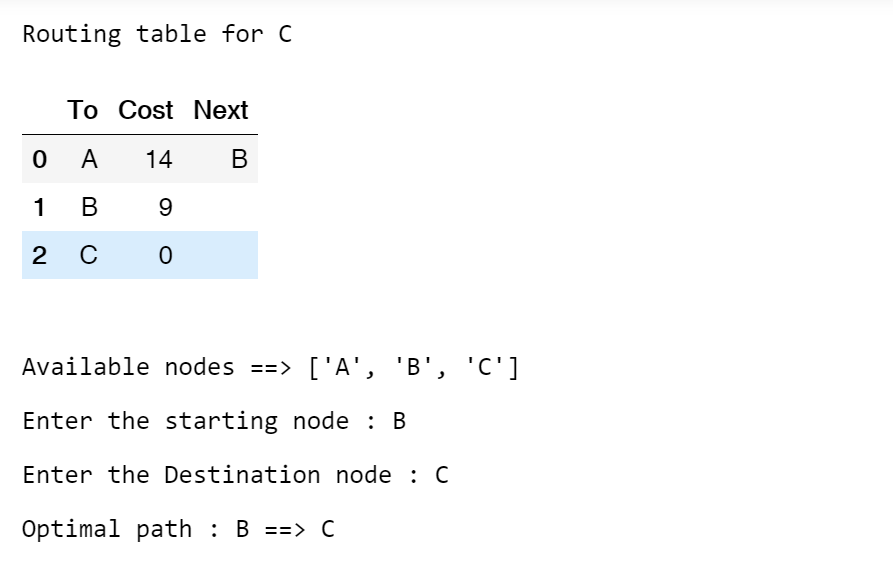
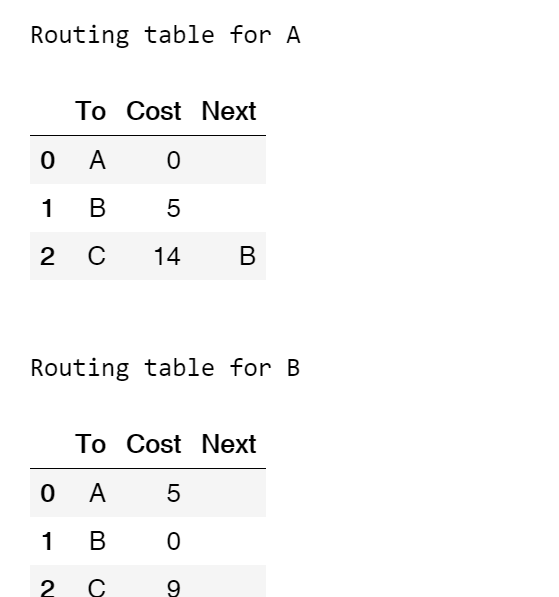




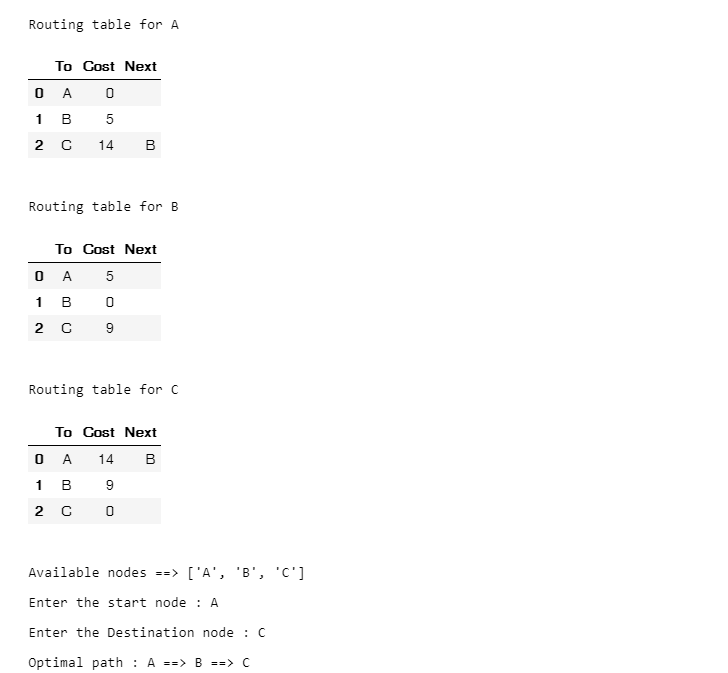
**Test-case 1**



**Test-case 2**



**Output**



**Actual Code**

import pandas as pd

from pandas import DataFrame

graph = {'A': {'B': 5, 'C': 15},

'B': {'A': 5, 'C': 9},

'C': {'A': 15, 'B': 9}

}

nodes = list(graph.keys())

routing\_table = []

for p, q in enumerate(nodes):

visited\_node = []

not\_visited\_node\_node = nodes.copy()

next\_node = len(nodes) \* ['']

inf = float('inf')

shortest\_distance = len(nodes) \* [inf]

root\_node = q

current\_node = q

shortest\_distance[ord(current\_node) - 65] = 0

while True:

for i in list(graph[current\_node].keys()):

if i not in visited\_node:

if shortest\_distance[ord(current\_node) - 65] + graph[current\_node][i] < shortest\_distance[ord(i) - 65]:

shortest\_distance[ord(

i) - 65] = shortest\_distance[ord(current\_node) - 65] + graph[current\_node][i]

if current\_node != root\_node:

next\_node[ord(i) - 65] = current\_node

visited\_node.append(current\_node)

not\_visited\_node\_node.remove(current\_node)

if len(not\_visited\_node\_node) == 0:

break

unvstd\_aasci = [ord(x) for x in not\_visited\_node\_node]

min\_value = min([shortest\_distance[j - 65] for j in unvstd\_aasci])

min\_index = [j for j, x in enumerate(

shortest\_distance) if x == min\_value]

for j in min\_index:

if chr(65 + j) in not\_visited\_node\_node:

current\_node = chr(65 + j)

break

for i in range(len(next\_node)):

if next\_node[i] == root\_node:

next\_node[i] = ''

routing\_table.append(DataFrame(

{'To': list(graph.keys()), 'Cost': shortest\_distance, 'Next': next\_node}))

print(f"\n\nRouting table for {root\_node}")

display(routing\_table[-1])

print(f"\n\nAvailable nodes ==> {nodes}")

start = input("\nEnter the starting node : ")

while start not in nodes:

print("Invalid Node....Try again")

start = input("Enter the start node : ")

dest = input("\nEnter the Destination node : ")

while dest not in nodes:

print("Invalid Node....Try again")

dest = input("Enter the Destination node : ")

index = ord(start) - 65

df = routing\_table[index]

path = []

path.append(start)

temp = dest

while df[df['To'] == temp]['Next'].values[0] != '':

path.insert(1, df[df['To'] == temp]['Next'].values[0])

temp = df[df['To'] == temp]['Next'].values[0]

print("\nOptimal path : ", end="")

for i in range(len(path)):

print(path[i], end=" ==> ")

print(dest)

**Question-2**

