Name :

Muhammad Junaid

Roll No:

222047

Submitted to:

Sir Awais

Python code

```
class City:
    def __init__(self, name, stDG, stDS, node):
        self.f_n = stDG  # straight Line Dist from Goal
        self.g_n = stDS  # straight Line Dist from Source
        self.total = self.f_n + self.g_n
        self.city_name = name
        self.node = node


class Node:
    def __init__(self, name):
        self.node_name = name
        self.next_nodes = []
        self.Start = None
```

```
def initialization(self):
    # Create Nodes
    Arad = Node("Arad")
    Sibiu = Node("Sibiu")
    RimicuVilcea = Node("RimicuVilcea")
    Pitesti = Node("Pitesti")
    Zerind = Node("Zerind")
    Timisoara = Node("Timisoara")
    Craiova = Node("Craiova")
    Bucharest = Node("Bucharest")
    Oradea = Node("Oradea")
    Fagaras = Node("Fagaras")

    # Setup Connections (Cities)
    Arad.next_nodes = [
        City("Sibiu", 253, 140, Sibiu),
        City("Zerind", 374, 75, Zerind),
        City("Timisoara", 329, 118, Timisoara)
    ]
```

```
Sibiu.next_nodes = [
   City("Arad", 366, 280, Arad),
   City("Fagaras", 176, 239, Fagaras),
   City("RimicuVilcea", 193, 220, RimicuVilcea),
   City("Oradea", 380, 291, Oradea)
]

RimicuVilcea.next_nodes = [
   City("Pitesti", 100, 317, Pitesti),
   City("Craiova", 160, 366, Craiova),
   City("Sibiu", 253, 300, Sibiu)
]

Pitesti.next_nodes = [
   City("RimicuVilcea", 193, 414, RimicuVilcea),
   City("Craiova", 160, 455, Craiova),
   City("Bucharest", 0, 418, Bucharest)
]

self.Start = Arad
print("Initialized")
```

```
    def search(self):
        dest = "ab"
        curr = self.Start
        print(curr.node_name)

        while dest != "Bucharest":
            min_total = curr.next_nodes[0].total
            psudo_curr = curr.next_nodes[0].node

            for i in range(1, len(curr.next_nodes)):
                if min_total > curr.next_nodes[i].total:
                    min_total = curr.next_nodes[i].total
                    psudo_curr = curr.next_nodes[i].node

            curr = psudo_curr
            dest = curr.node_name
            print(dest)

class Astar:
    @staticmethod
    def main():
        n = Node("dummy")
        n.initialization()
        n.search()

if __name__ == "__main__":
    Astar.main()
```