**Riphah International University**

**Artificial Intelligence (AI)**

**Lab 5**

**Submitted by:** Muhammad Qasim

**Sap ID:** 37137

**Section:** BSCS-6A

**Submitted To:** Mahjabeen

**Riphah School of Computing & Innovation Faculty of Computing Riphah International University, Lahore 2023**

**Lab**

import heapq

def astar(graph, heuristic, start, goal):

    open\_list = [(0, start)]

    closed = set()

    g = {node: float('inf') for node in graph}

    for node in heuristic:

        if node not in g:

            g[node] = float('inf')

    g[start] = 0

    pred = {}

    while open\_list:

        f, curr = heapq.heappop(open\_list)

        if curr == goal:

            path = []

            while curr != start:

                path.append(curr)

                curr = pred[curr]

            path.append(start)

            return list(reversed(path))

        closed.add(curr)

        for neigh, dist in graph[curr]:

            if neigh in closed:

                continue

            tentative\_g = g[curr] + dist

            if tentative\_g < g[neigh]:

                g[neigh] = tentative\_g

                f = tentative\_g + heuristic[neigh]

                heapq.heappush(open\_list, (f, neigh))

                pred[neigh] = curr

    return None

graph = {

    'A': [('B', 2), ('E', 3)],

    'B': [('C', 1), ('G', 9)],

    'E': [('D', 6)],

    'D': [('G', 1)]

}

heuristic = {'A': 11, 'B': 6, 'C': 99, 'D': 1, 'E': 7, 'G': 0}

start = 'A'

goal = 'G'

path = astar(graph, heuristic, start, goal)

print(path)

Output:

