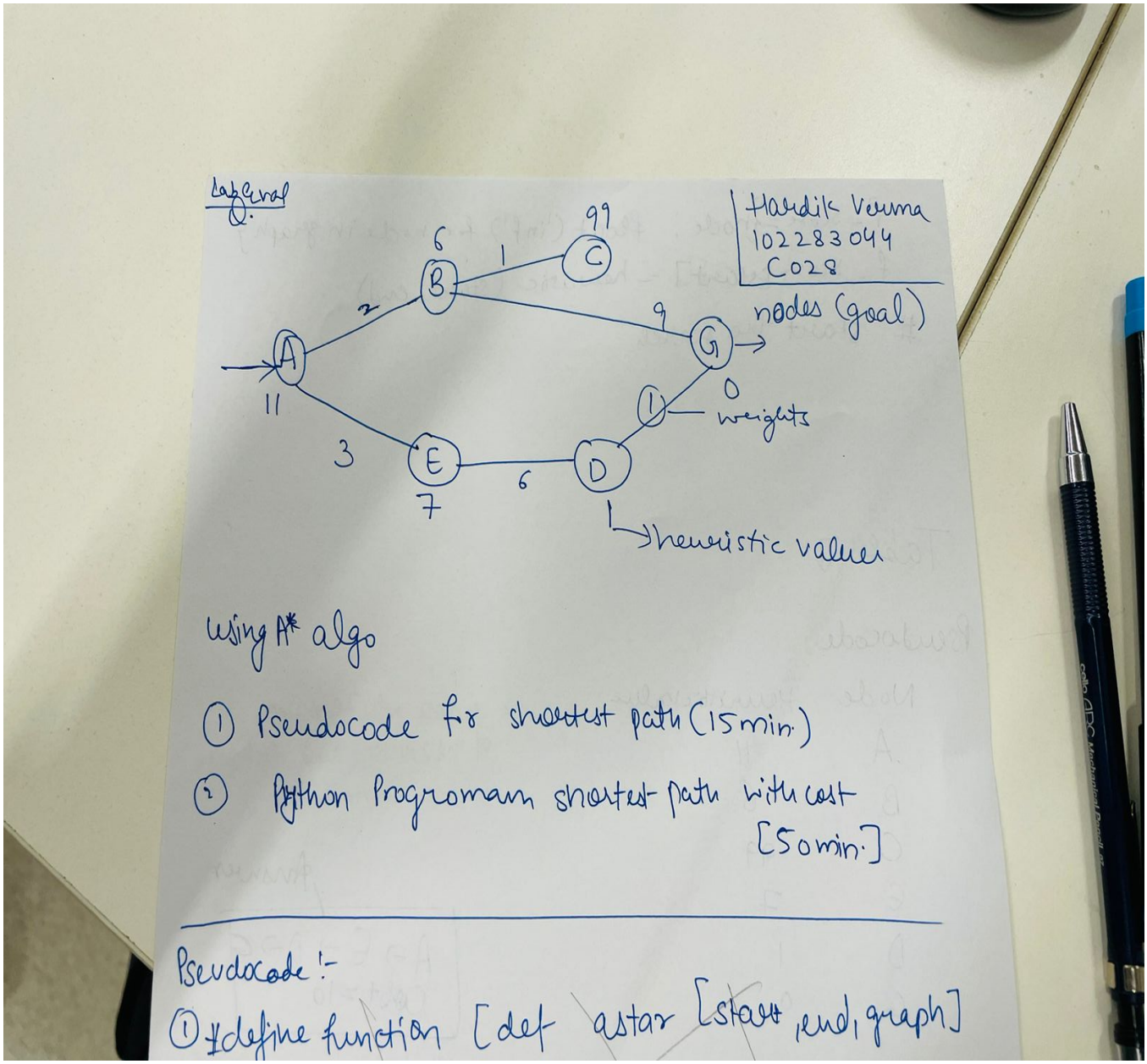
Artificial Intelligence Lab Evaluation Pre-EST **7th April,2023**

Sumbitted by:- Hardik Verma, 102283044, CO28

Question:- Python Program for given Graph using A* Algorithm. Output should represent shortest path with cost.



CODE:-

```
# Define the graph with nodes
graph = {
   'A': {'B': 2, 'E': 3},
   'B': {'C': 1, 'G': 9, 'A':2},
   'E': {'A':3, 'D':6},
   'C': {'B':1},
   'D': {'E':6, 'G':1},
   'G': {'B':9, 'D':1}
}

# Define the heuristic values
heuristic = {
   'A': 10,
   'B': 6,
   'C': 99,
   'D': 1,
   'E': 7,
   'G': 0
}

# Define the A* algo
def astar(graph, start, goal, heuristic):
   open_list = [(0, start)]
   closed_list = set()
   distances = {node: float('inf') for node in graph}
   distances[start] = 0
   paths = {start: []}
   while open_list:
      current_cost, current_node = min(open_list)
      if current_node == goal:
         return paths[current_node] + [current_node], distances[current_node]
      closed_list.add(current_node)
      open_list.remove((current_cost, current_node))
      for neighbor, cost in graph[current_node].items():
```

```
            if neighbor in closed_list:
                continue
            tentative_distance = distances[current_node] + cost
            if tentative_distance < distances[neighbor]:
                distances[neighbor] = tentative_distance
                paths[neighbor] = paths[current_node] + [current_node]
                fx = tentative_distance + heuristic[neighbor]
                open_list.append((fx, neighbor))
    return None, None
path, cost = astar(graph, 'A', 'G', heuristic)
if path is None:
    print("Erro! No path found")
else:
    print("Shortest path:", path)
    print("Cost:", cost)
```

OUTPUT:-