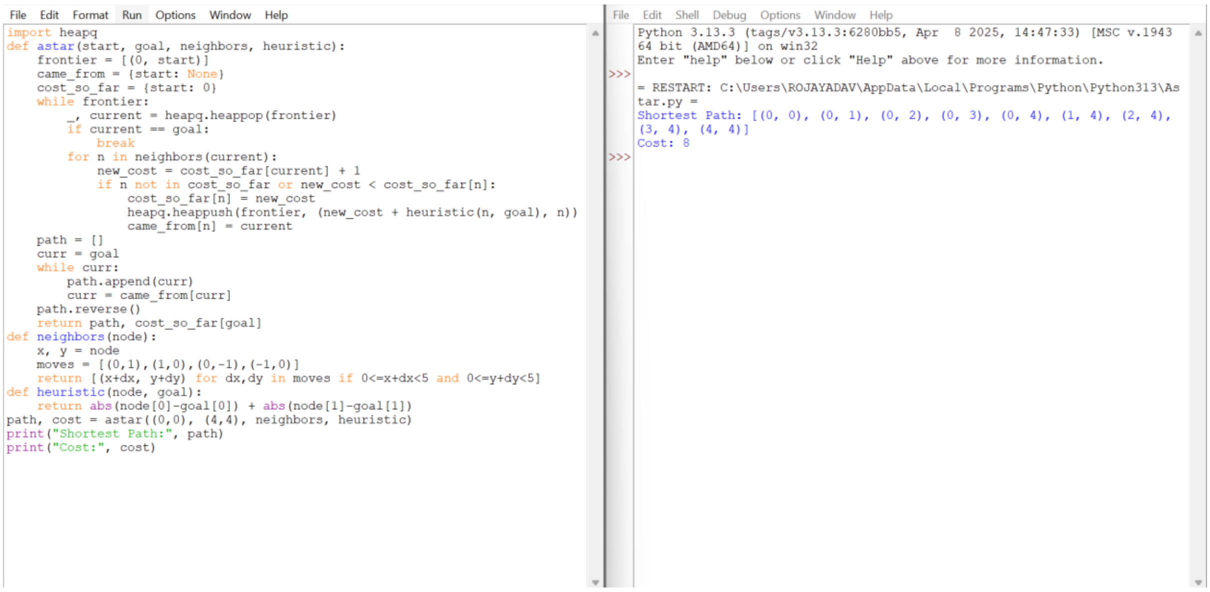
Write the python program to implement A* algorithm.



```python
import heapq
def astar(start, goal, neighbors, heuristic):
    frontier = [(0, start)]
    came_from = {start: None}
    cost_so_far = {start: 0}
    while frontier:
        _, current = heapq.heappop(frontier)
        if current == goal:
            break
        for n in neighbors(current):
            new_cost = cost_so_far[current] + 1
            if n not in cost_so_far or new_cost < cost_so_far[n]:
                cost_so_far[n] = new_cost
                heapq.heappush(frontier, (new_cost + heuristic(n, goal), n))
                came_from[n] = current
    path = []
    curr = goal
    while curr:
        path.append(curr)
        curr = came_from[curr]
    path.reverse()
    return path, cost_so_far[goal]
def neighbors(node):
    x, y = node
    moves = [(0,1),(1,0),(0,-1),(-1,0)]
    return [(x+dx, y+dy) for dx,dy in moves if 0<=x+dx<5 and 0<=y+dy<5]
def heuristic(node, goal):
    return abs(node[0]-goal[0]) + abs(node[1]-goal[1])
path, cost = astar((0,0), (4,4), neighbors, heuristic)
print("Shortest Path:", path)
print("Cost:", cost)
```

```
    Python 3.13.3 (tags/v3.13.3:6280bb5, Apr  8 2025, 14:47:33) [MSC v.1943
    64 bit (AMD64)] on win32
    Enter "help" below or click "Help" above for more information.
>>>
    = RESTART: C:\Users\ROJAYADAV\AppData\Local\Programs\Python\Python313\As
    tar.py =
    Shortest Path: [(0, 0), (0, 1), (0, 2), (0, 3), (0, 4), (1, 4), (2, 4),
    (3, 4), (4, 4)]
    Cost: 8
>>>
```