**A\* ALGORITHM**

Name : AYESHA SIKANDAR

Roll no. 044

Section : BSDS 3A

**Code Explanation:**

First, I created a dictionary named tree which stores each node along with its connected nodes and the cost between them. Another dictionary called heuristic stores the estimated cost from each node to the goal, and cost keeps track of the actual cost from the start node.

Then, I defined a **function AStarSearch()** which performs the A\* algorithm. Inside this function, two lists are used —

* **opened**: stores the nodes that need to be explored.
* **closed**: stores the nodes that have already been visited.

At the start, the program adds the starting node 'S' into the opened list. Then, in every loop, it picks the node with the smallest total cost (f = g + h). That node becomes the current node and is moved to the closed list.

Next, the algorithm checks all the connected nodes (children) of the current node. It updates their costs and heuristic values and adds them into the opened list if they haven’t been visited. This process repeats until the goal node 'G' is reached.

After reaching 'G', a trace-back loop runs to reconstruct the shortest path by checking which nodes lead to the goal with the correct cost. The final **optimal path** is then reversed and printed as the output.

**Why I Made This:**

I made this to understand how the **A\*** search algorithm works for finding the shortest path in a graph.  
It is one of the most efficient algorithms used in Artificial Intelligence, games, and navigationsystems.  
By making this program, I practiced how to use heuristics, loops, and data structures like dictionaries and lists in Python.

**How I Made It:**

I made this by first defining the **graph structure** using dictionaries for nodes and heuristics.  
Then I added logic inside the AStarSearch() function to calculate and compare total costs for each node.  
Finally, I wrote a path reconstruction loop to trace back the shortest path once the goal was reached.  
The program uses loops, lists, conditionals, and cost calculations to perform all steps of the A\* algorithm correctly.