```
import heapq

def a_star(start, goal, graph, heuristic):
    open_list = []
    heapq.heappush(open_list, (0, start))

    g = {start: 0}
    parent = {start: None}
    closed_list = []

    while open_list:
        f_score, node = heapq.heappop(open_list)
        closed_list.append(node)


        if node == goal:
            path = []
            while node is not None:
                path.append(node)
                node = parent[node]
            path.reverse()
            return path


        for neighbour, w in graph[node].items():
            if neighbour in closed_list:
                continue

            new_g = g[node] + w
            f_score = new_g + heuristic[neighbour]


            if neighbour not in g or new_g < g[neighbour]:
                g[neighbour] = new_g
                parent[neighbour] = node
                heapq.heappush(open_list, (f_score, neighbour))

    return None


def main():
    graph = {}
    heuristic = {}
```

```
while True:
    print("\n--- A* Search Menu ---")
    print("1. Enter a new connection")
    print("2. Run the A* Algorithm")
    print("3. Exit")
    option = int(input("Enter your choice: "))

    if option == 1:
        node1 = input("Enter the name of node 1: ")
        node2 = input("Enter the name of node 2: ")
        weight = int(input("Enter the weight between them: "))


        if node1 not in graph:
            heu1 = int(input(f"Enter the heuristic value for {node1}: "))
            graph[node1] = {}
            heuristic[node1] = heu1


        if node2 not in graph:
            heu2 = int(input(f"Enter the heuristic value for {node2}: "))
            graph[node2] = {}
            heuristic[node2] = heu2


        graph[node1][node2] = weight
        graph[node2][node1] = weight

    elif option == 2:
        start = input("Enter the start node: ")
        goal = input("Enter the goal node: ")
        path = a_star(start, goal, graph, heuristic)

        if path is not None:
            print("Shortest path found:")
            print(" → ".join(path))
        else:
            print("Path not found!")

    elif option == 3:
        print("Exiting program...")
        break
    else:
        print("Invalid choice! Please enter 1, 2, or 3.")
```

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