Data Structures (CS-2001) Assignment#1
"One-Stop Management System Using Linear Data Structures"

Objective: The objective of this assignment is to build a One-Stop Management System that uses linear data structures such as arrays, linked lists, stacks, and queues. This system will help manage student service requests, agents, and service logs.

Overview:
In this assignment, students will create a simplified One-Stop system where they can:
Add, remove, and search for service requests (tickets).
Manage support agents (one each for IT, admin, accounts, and 3 for academics) and assign them tickets.
Track ticket resolution and response logs using stacks and queues.
Sort tickets based on priority, creation time, or agent assignment using different sorting algorithms.
Generate reports and analyze system performance based on response times.

Requirements:

1. Service Request (Ticket) Management
You need to implement a data structure to manage service requests (tickets). Each ticket will have the following attributes:

Ticket ID (int)
Customer Name (String)
Priority (int) (1 = High, 2 = Medium, 3 = Low)
Service Request description (String)
Creation Time (timestamp)
Status (open/closed)
Ticket close time: When the ticket is finally closed (timestamp)

Features to Implement:

1. Add a Ticket: Add a new support ticket to the system.
2. Remove a Ticket: Remove a ticket from the system using its Ticket ID.
3. Search for a Ticket: Search for a ticket by Ticket ID or by the customer's name.
4. Sort Tickets: Implement sorting algorithms to sort tickets based on the attributes below. Users will be presented a choice about which sorting algorithm they want to use:
   Priority
   Creation Time
   Customer Name

Data Structure Choices:
Use a Linked List to manage tickets.

2. Agent Management
Each support agent will have the following attributes:

Agent ID (int)
Name (String)
Assigned Tickets (List of Ticket IDs)
Availability (boolean)

Features to Implement:

1. Add an Agent: Add a new support agent to the system.
2. Assign Ticket to Agent: Automatically assign open tickets to available agents based on the following conditions:
3. Agents should be assigned high-priority tickets first.
4. Agents with fewer open tickets should be assigned new tickets first.
5. Mark Agent Unavailable: After an agent is assigned a ticket, mark them as unavailable if they are at full capacity (e.g., max 5 open tickets).
6. Status (String): Available/ Unavailable

Data Structure Choices:
Use a Dynamic Array to manage agents.

3. Ticket Resolution Logs (Transaction Log)
To track the One-Stop's ticket transactions, implement a log system using Stacks and Queues:

1. Ticket Resolution Log (Stack): Maintain a stack of the most recent ticket resolution operations. Every time a ticket is closed, log the transaction into the stack.
2. Pending Ticket Queue: Maintain a queue of tickets waiting for agent assignment (FIFO structure). This queue should handle all incoming tickets based on their priority and creation time.
3. Log Closed Ticket: When a ticket is resolved (marked as closed), log the transaction into the resolution stack.
4. Add Ticket to Pending Queue: All incoming tickets (based on priority) will be placed in the pending ticket queue until an agent is available to handle them.
5. View Logs: Display the most recent ticket resolution logs.

Data Structure Choices:
Stacks and Queues

4. Reporting and Analytics
Create the following reports using sorting and searching algorithms:

1. List of Open Tickets: Search for open tickets using the searching algorithm Display all open tickets sorted by priority or creation time.

2. Agent Ticket Load: Display the list of agents and their currently assigned tickets. Sort agents by the number of assigned tickets.
3. Ticket Resolution History: Show the most recent ticket resolutions using the resolution log (stack).

Data Structures:
Queue for managing pending ticket assignments.
Stack for managing the resolution logs.

Sorting Algorithms:
Implement Elementary sorts (Bubble Sort, Insertion Sort, Selection Sort), Advanced sorts (Merge Sort and Quicksort) for sorting and interpolation sort and Bubble sort for searching.
There should be a **configuration file** where the user can specify which sorting and searching algorithm will be used by default. When the items to be sorted are less than any threshold (decide the threshold yourselves), the configured elementary sorting algorithm should be used. Otherwise the configured Merge Sort or Quick Sort should be used when handling larger datasets.

Efficiency:
Analyze the time complexity of the operations for managing tickets, agents, and logs. Every operation needs to have the time taken by that operation mentioned on the screen.

Deliverables:
Create a command-line or GUI interface that allows users to interact with the system (e.g., adding/removing tickets, assigning agents, closing tickets).
Code: Well-documented source code with clear comments explaining the logic.