

Lab - Miscellaneous Tasks

Task 1: Arrays

Scenario:

You are building a simple task management application that stores tasks in an array. Each task has a title, description, and status (e.g., "To Do," "In Progress," "Done"). Design and implement the array operations needed for this application.

Tasks:

1. Create a structure or class representing a task with title, description, and status fields.
2. Implement a function to add a new task to the array.
3. Write a function to display all tasks in the array.
4. Develop a function to mark a task as "Done" by updating its status.
5. Implement a function to delete a task from the array by its title.

Task 2: Linked Lists

Scenario:

You are working on a music playlist application where each song is represented by a node in a linked list. Each node contains the song's title, artist, and duration. Design and implement the linked list operations required for managing the playlist.

Tasks:

1. Create a class or structure for a song node with title, artist, and duration fields.
2. Implement a function to add a new song to the playlist.
3. Write a function to display the entire playlist.
4. Develop a function to remove a song from the playlist by its title.
5. Implement a function to find the total duration of the playlist.

Task 3: Stacks

Scenario:

You are developing a text editor that needs to keep track of the history of user actions (e.g., typing, deleting, undoing). Design and implement a stack-based system to manage the user actions.

Tasks:

1. Create a stack to store user actions (e.g., strings representing actions).
2. Implement a function to push a new user action onto the stack.
3. Write a function to pop the last user action from the stack (undo operation).
4. Develop a function to display the entire history of user actions.
5. Implement a function to clear the entire history stack.

Task 4: Queues

Lab - Miscellaneous Tasks

Scenario:

You are building a print job management system for a printer. Print jobs are represented as nodes in a queue. Each node contains information about the document to be printed. Design and implement the queue operations needed for the print job management.

Tasks:

1. Create a class or structure for a print job node with document information.
2. Implement a function to enqueue a new print job.
3. Write a function to dequeue the next print job in the queue.
4. Develop a function to display the details of all print jobs in the queue.
5. Implement a function to clear all print jobs from the queue.