Data Structures and Algorithms
Department of Computer Science
University of Engineering and Technology, Lahore

Mid Term | Fall 2025 | Time: 60 Mintus

Name:_____ | Roll no:_____ | Section:_____

Questions	CLOs	Marks
1. The "Hot Potato" Process Scheduler You are writing a CPU scheduler for an old operating system. The system uses a "**Round Robin**" technique to manage active programs. Programs sit in a circle. The CPU gives a fixed time slot (quantum) to the current program. • If the program finishes within the time slot, it is removed from the circle. • If it doesn't finish, the CPU moves to the immediate next program in the circle. Implement this using a **Circular Linked List**. 1. Create a node **Process** with **ID** and **ExecutionTime**. 2. **cycle(time_quantum)**: Iterate through the circular list. For each node, **subtract time_quantum** from its **ExecutionTime**. Logic: If ExecutionTime becomes <= 0 print "Process [ID] Completed" and **delete** that node from the list, repairing the links so the circle remains intact. 3. Continue cycling until the list is empty.	CLO1, CLO3	10
2. The "Undo/Redo" Text Editor Engine You are building the core logic for a new text editor called "Notepad++ Lite". The most critical feature requested is a robust **Undo** and **Redo** system. Every time a user types a word, it is pushed as an action. If they hit "Undo", the action is reverted but saved in case they want to "Redo" it later. However, if the user types something *new* after an Undo, the Redo history must be cleared (standard editor behavior).	CLO1, CLO3	10

Implement a class **EditorHistory** using **Two Stacks** (UndoStack and RedoStack): 1. **typeWord(word)**: Pushes the word onto the UndoStack and **clears** the RedoStack entirely. 2. **undo()**: Pops the top word from UndoStack and pushes it onto RedoStack. 3. **redo()**: Pops the top word from RedoStack and pushes it back onto UndoStack. **Constraint:** You must verify if stacks are empty before popping to avoid crashing the program. The stacks should be implemented manually using arrays with a fixed size of 8.		