**Lab Sheet-IV**

**INSTRUCTIONS**

Given below are the lab questions for the day. For each question, you should prepare a document with following titles.

1. PROBLEM
2. ALGORITHM
3. CODE (with formal coding standard)
4. TESTING
   1. INPUT SET
   2. OUTPUT

**QUESTIONS**

**1.** A printer is connected to 4 computers (computer 1, computer 2, computer 3, and computer 4). It can accept 10 printing requests at a time. Write program which uses a **simple queue** to implement the following scenarios.

a. Assume at a particular time the printing request reaching the printer was from the computers 2, 1, 3, 2, 4, 2, 3. Display the current status of the queue.

b. Consider that 3 printing processes are completed. Display the current status of the queue.

c. New Printing request arrived from computers 2, 3, 4, 1. Display the status

**2.** In a ticket counter assume 8 people are waiting in queue for getting the ticket. Only 10 people can stand in the queue at a time. Write program which uses a simple queue to implement the following scenarios.

a. Currently the persons standing in the queue are Albert, Raj, Rohan, Daya, Ritu, Angel, Vihar (8 Persons). Display the current status of the queue. First Person Is Albert.

b. Rohan got the ticket. Print the status of the queue.

c. New people arrived at the queue are Laxman, Akash, Bob,Tinku,Ben,Riya and Neha . Display the current status of the queue.

d. Assume Angel got the ticket now .Display the current status.

**3.** Implement a circular queue using array

**4.** Implement a circular queue.

1. Implement “Round Robin” scheduling algorithm using circular queue.

**[** Round robin is the scheduling algorithm used by the CPU during execution of the processes. Round robin is designed specifically for time sharing systems. Processes along with its total time for execution are kept in a queue. Each process will be taken from the queue and is executed for a fixed amount of time, called “quantum. Process execution will be done cyclically if it is not complete in one “quantum.” Meanwhile new processes are added to the tail of the queue.

Example: Following are the set of jobs and execution time. Let quantum= 5 seconds

|  |  |
| --- | --- |
| Process | Execution Time |
| P1 (front) | 10 seconds |
| P2 | 5 Seconds |
| P3 | 15 second |
| P4(rear) | 10 seconds |

P1 executes for 5 seconds ie one quantum; then P2, P3…..Whichever process completes the execution within 5 second will be pop out from queue. Others will remain and wait for their next turn. ]

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| P1 | P2 | P3 | P4 | P1 | P3 | P4 | P3 |

*Execution sequence*

**5**. Implement a deque using array

**6.** Implement a palindrome checker using deque.