

**Lab 06 – 21-10-2024**

**Note 1:** This is graded lab. Code in C++. Evaluation will be lenient but no compromise on cheating or any other violation. Therefore, please do your own work.

**Task 1:** Use Linked List implementation and add following member functions:

- **addInOrder:** Assume that existing elements are in order, add new element in order.
- **removeDuplicatesElements:** Check and remove the duplicate elements
- **removeDuplicates:** Check and remove all the elements having more than one occurrences in the list

**Task 1:** Use Linked List (integer) implementation and add following member functions:

- **splitEvenElements:** Split all node with even elements into two nodes such that there will be no even element
- **mergeElements:** If there are more than one elements in the list, merge elements with previous or next element such that there will be no two nodes in the list having value less than 50

**Task 2.** Consider process simulation already shared with you. Extend the program such that there may be zero, one or two process enters into system with random time between 2-20 units/ seconds.

Further, consider there are two processors PF and PS. In every iteration, if system has two or more processes in the queue, assign first two processes to both processors. In each iteration, if there are two or more processes, both processors will execute one process each, otherwise, only first processor will execute the only process. See sample output:

```
Process Times: 15 5 13 13 7
Process 0 in processor PF for 4 seconds
Process 1 in processor PS for 4 seconds
Process 2 in processor PF for 4 seconds
Process 3 in processor PS for 4 seconds
Process 4 in processor PF for 4 seconds
Process 5 time 10 seconds
Process 0 in processor PF for 4 seconds
Process 1 in processor PS for 1 seconds
Process 2 in processor PF for 4 seconds
Process 3 in processor PS for 4 seconds
Process 4 in processor PF for 3 seconds
Process 5 in processor PS for 4 seconds
Process 6 time 3 seconds
Process 7 time 5 seconds
Process 0 in processor PF for 4 seconds
Process 2 in processor PS for 4 seconds
Process 3 in processor PF for 4 seconds
Process 5 in processor PS for 4 seconds
Process 6 in processor PF for 3 seconds
Process 7 in processor PS for 4 seconds
No new process
Process 0 in processor PF for 3 seconds
Process 2 in processor PS for 1 seconds
Process 3 in processor PF for 1 seconds
Process 5 in processor PS for 2 seconds
Process 7 in processor PF for 1 seconds
```

**Task 3.** I may add one task later