

1-List Data Structures

Practical exercises

Note: You can select and do some options according to your ability only. We would like to note you that the more questions you do the better for you in doing final practical and writing exams.

Question 1. Write a Java program to implement a singly linked list of integer values with the following operations:

1. void addToHead(int x) - add a node with value x at the head of a list.
2. void addToTail(int x) - add a node with value x at the tail of a list.
3. void addAfter(Node p, int x) - add a node with value x after the node p.
4. void traverse() - traverse from head to tail and display info of all nodes in the list.
5. int deleteFromHead() - delete the head and return its info.
6. int deleteFromTail() - delete the tail and return its info.
7. int deleteAfter(Node p) - delete the node after the node p and return its info.
8. void delete(int x) - delete the first node whose info is equal to x.
9. Node search(int x) - search and return the reference to the first node having info x.
10. int count() - count and return number of nodes in the list.
11. void delete(int i) - delete an i-th node on the list. Ensure that such a node exists.
12. void sort() - sort the list by ascending order of info.
13. void delete(Node p) - delete node p if it exists in the list.
14. int [] toArray() - create and return array containing info of all nodes in the list.
15. Merge two ordered singly linked lists of integers into one ordered list.
16. void addBefore(Node p, int x) - add a node with value x before the node p.
17. Attach a singly linked list to the end of another singly linked list.
18. int max() - find and return the maximum value in the list.
19. int min() - find and return the minimum value in the list.
20. int sum() - return the sum of all values in the list.
21. int avg() - return the average of all values in the list.
22. boolean sorted() - check and return true if the list is sorted, return false if the list is not sorted.
23. void insert(int x) - insert node with value x into sorted list so that the new list is sorted.
24. Reverse a singly linked list using only one pass through the list.
25. Check whether two singly linked lists have the same contents.