

1. Write a C program to create a linked list having 10 nodes with random values between 0 and 100. Take the choice from user and implement the following functions:

Choice	Function
0	Print the values of the nodes of linked list
1	Add a new node at the beginning having a random value
2	Add a new node at the end having a random value
3	Add a new node between (i)th and (i+1)th node having a random value
4	Delete a node at the beginning
5	Delete a node at the end
6	Delete a node at (i)th position
7	Search an element in the list
8	Compute the average of all the elements of list

2. Write a C program having a function that takes 2 linked lists as arguments, and returns a new linked list that contains every number the two lists have in common. One or both of the input lists may be empty, or may contain numbers. Don't change either of the two input lists. Assume that each list has the unique numbers, i.e., at most only one copy of any particular number. If list1 contains the numbers 3, 2, 4, 9, 8, 1, 0, 5 and list2 contains the numbers 0, 1, 7, 6, 4, 5, 9, then the resultant list should return a list of the numbers 0, 1, 4, 5, 9 in any order.

3. Write a C program having a function that takes 2 linked lists as arguments, and returns a new linked list that contains every number the two lists do NOT have in common. Use the same assumptions as in Question 2. If list1 contains the numbers 3, 2, 4, 9, 8, 1, 0, 5 and list2 contains the numbers 0, 1, 7, 6, 4, 5, 9, then the resultant list should return a list of the numbers 2, 3, 6, 7, 8 in any order.

4. What's the order of the operations described in parts D and E? If the lists were sorted, would this order change? Explain. Rewrite the routines in D and E assuming that list1 and list2 are already sorted smallest to largest.