Part I: Basic operators in List
(Singly, Doubly, and Circular)

Note: All linked lists are null-terminated and singly linked. Input lists may be empty, unless otherwise stated. If new nodes may be created, then report whether or not space was exhausted, using a boolean reference parameter.

Answer each question with a C/C++ subroutine.

1. Write a program in C to create and display Singly Linked List.

Test Data :

Input the number of nodes : 3

Input data for node 1 : 5

Input data for node 2 : 6

Input data for node 3 : 7

Expected Output :

```
Data entered in the list :
Data = 5
Data = 6
Data = 7
```

2. Write a program in C to create a singly linked list of n nodes and display it in reverse order.

Test Data :

Input the number of nodes : 3

Input data for node 1 : 5

Input data for node 2 : 6

Input data for node 3 : 7

Expected Output :

```
Data entered in the list are :
Data = 5
Data = 6
Data = 7

The list in reverse are :
Data = 7
Data = 6
Data = 5
```

3. Write a program in C to create a singly linked list of n nodes and count the number of nodes.

Test Data :

Input the number of nodes : 3

Input data for node 1 : 5

Input data for node 2 : 6

Input data for node 3 : 7

Expected Output :

```
Data entered in the list are :
Data = 5
Data = 6
Data = 7

Total number of nodes = 3
```

4. Write a program in C to insert a new node at the beginning of a Singly Linked List.

Test Data and Expected Output :

```
Input the number of nodes : 3
Input data for node 1 : 5
Input data for node 2 : 6
Input data for node 3 : 7

Data entered in the list are :
Data = 5
Data = 6
Data = 7

Input data to insert at the beginning of the list : 4

Data after inserted in the list are :
Data = 4
Data = 5
Data = 6
Data = 7
```

5. Write a program in C to insert a new node at the end of a Singly Linked List.

Test Data and Expected Output :

Input the number of nodes : 3
Input data for node 1 : 5
Input data for node 2 : 6
Input data for node 3 : 7

Data entered in the list are :
Data = 5
Data = 6
Data = 7

Input data to insert at the end of the list : 8

Data, after inserted in the list are :
Data = 5
Data = 6
Data = 7
Data = 8

6. Write a program in C to insert a new node at the middle of Singly Linked List.
Test Data and Expected Output :

Input the number of nodes (3 or more) : 4
Input data for node 1 : 1
Input data for node 2 : 2
Input data for node 3 : 3
Input data for node 4 : 4

Data entered in the list are :
Data = 1
Data = 2
Data = 3
Data = 4

Input data to insert in the middle of the list : 5
Input the position to insert new node : 3

Insertion completed successfully.

The new list are :
Data = 1
Data = 2
Data = 5
Data = 3
Data = 4

7. Write a program in C to delete first node of Singly Linked List.

Test Data :

Input the number of nodes : 3

Input data for node 1 : 2

Input data for node 2 : 3

Input data for node 3 : 4

Expected Output :

Data entered in the list are :
Data = 2
Data = 3
Data = 4

Data of node 1 which is being deleted is : 2

Data, after deletion of first node :
Data = 3
Data = 4

8. Write a program in C to delete a node from the middle of Singly Linked List.

Test Data and Expected Output :

Input the number of nodes : 3
Input data for node 1 : 2
Input data for node 2 : 5
Input data for node 3 : 8

Data entered in the list are :
Data = 2
Data = 5
Data = 8

Input the position of node to delete : 2

Deletion completed successfully.

The new list are :
Data = 2
Data = 8

9. Write a program in C to delete the last node of Singly Linked List

Test Data :
Input the number of nodes : 3
Input data for node 1 : 1
Input data for node 2 : 2
Input data for node 3 : 3
Expected Output :

```
Data entered in the list are :
Data = 1
Data = 2
Data = 3

The new list after deletion the last node are  :
Data = 1
Data = 2
```

10. Write a program in C to search an existing element in a singly linked list

Test Data and Expected Output :

```
Input the number of nodes : 3

Input data for node 1 : 2
Input data for node 2 : 5
Input data for node 3 : 8

Data entered in the list are :
Data = 2
Data = 5
Data = 8

Input the element to be searched : 5
Element found at node 2
```

11. Write a program in C to create and display a doubly linked list
Test Data :
Input the number of nodes : 3
Input data for node 1 : 2
Input data for node 2 : 5

Input data for node 3 : 8

Expected Output :

```
Data entered on the list are :
node 1 : 2
node 2 : 5
node 3 : 8
```

12. Write a program in C to create a doubly linked list and display in reverse order.

Test Data :

Input the number of nodes : 3

Input data for node 1 : 2

Input data for node 2 : 5

Input data for node 3 : 8

Expected Output :

```
Data in reverse order are :
Data in node 1 : 8
Data in node 2 : 5
Data in node 3 : 2
```

13. Write a program in C to insert a new node at the beginning in a doubly linked list.

Test Data and Expected Output :

```
Input the number of nodes : 3
Input data for node 1 : 2
Input data for node 2 : 5
Input data for node 3 : 8

Data entered in the list are :
node 1 : 2
node 2 : 5
node 3 : 8
Input data for the first node : 1

After insertion the new list are :
node 1 : 1
node 2 : 2
node 3 : 5
node 4 : 8
```

14. Write a program in C to insert a new node at the end of a doubly linked list.
Test Data and Expected Output :

Input the number of nodes : 3
Input data for node 1 : 2
Input data for node 2 : 5
Input data for node 3 : 8

Data entered in the list are :
node 1 : 2
node 2 : 5
node 3 : 8
Input data for the last node : 9

After insertion the new list are :
node 1 : 2
node 2 : 5
node 3 : 8
node 4 : 9

15. Write a program in C to insert a new node at any position in a doubly linked list
Test Data and Expected Output :

Input the number of nodes (3 or more): 3
Input data for node 1 : 2
Input data for node 2 : 4
Input data for node 3 : 5

Data entered in the list are :
node 1 : 2
node 2 : 4
node 3 : 5
Input the position (2 to 2) to insert a new node : 2
Input data for the position 2 : 3

After insertion the new list are :
node 1 : 2
node 2 : 3
node 3 : 4
node 4 : 5

16. Write a program in C to insert a new node at the middle in a doubly linked list.
Test Data and Expected Output :

Doubly Linked List : Insert new node at the middle in a doubly linked list :

```
Input the number of nodes (3 or more ): 3
Input data for node 1 : 2
Input data for node 2 : 4
Input data for node 3 : 5

Data entered in the list are :
node 1 : 2
node 2 : 4
node 3 : 5
Input the position ( 2 to 2 ) to insert a new node : 2
Input data for the position 2 : 3

After insertion the new list are :
node 1 : 2
node 2 : 3
node 3 : 4
node 4 : 5
```

17. Write a program in C to delete a node from the beginning of a doubly linked list.

Test Data and Expected Output :

```
Input the number of nodes (3 or more ): 3
Input data for node 1 : 1
Input data for node 2 : 2
Input data for node 3 : 3

Data entered in the list are :
node 1 : 1
node 2 : 2
node 3 : 3

After deletion the new list are :
node 1 : 2
node 2 : 3
```

18. Write a program in C to delete a node from the last of a doubly linked list.

Test Data and Expected Output :

```
Input the number of nodes (3 or more ): 3
Input data for node 1 : 1
Input data for node 2 : 2
```

Input data for node 3 : 3

Data entered in the list are :
node 1 : 1
node 2 : 2
node 3 : 3

After deletion the new list are :
node 1 : 1
node 2 : 2

19. Write a program in C to delete a node from any position of a doubly linked list.
Test Data and Expected Output :

Doubly Linked List : Delete node from any position of a doubly linked list :

Input the number of nodes (3 or more): 3
Input data for node 1 : 1
Input data for node 2 : 2
Input data for node 3 : 3

Data entered in the list are :
node 1 : 1
node 2 : 2
node 3 : 3
Input the position (1 to 3) to delete a node : 3

After deletion the new list are :
node 1 : 1
node 2 : 2

20. Write a program in C to delete a node from the middle of a doubly linked list.
Test Data and Expected Output :

Input the number of nodes (3 or more): 3
Input data for node 1 : 1
Input data for node 2 : 2
Input data for node 3 : 3

Data entered in the list are :
node 1 : 1
node 2 : 2
node 3 : 3

Input the position (1 to 3) to delete a node : 2

After deletion the new list are :
node 1 : 1
node 2 : 3

21. Write a program in C to find the maximum value from a doubly linked list
Test Data :
Input the number of nodes : 3
Input data for node 1 : 5
Input data for node 2 : 9
Input data for node 3 : 1
Expected Output :

Data entered in the list are :
node 1 : 5
node 2 : 9
node 3 : 1

The Maximum Value in the Linked List : 9

22. Write a program in C to create and display a circular linked list
Test Data :
Input the number of nodes : 3
Input data for node 1 : 2
Input data for node 2 : 5
Input data for node 3 : 8
Expected Output :

Data entered in the list are :
Data 1 = 2
Data 2 = 5
Data 3 = 8

23. Write a program in C to insert a node at the beginning of a circular linked list.
Test Data and Expected Output :

Input the number of nodes : 3

Input data for node 1 : 2
Input data for node 2 : 5
Input data for node 3 : 8

Data entered in the list are :
Data 1 = 2
Data 2 = 5
Data 3 = 8
Input data to be inserted at the beginning : 1

After insertion the new list are :
Data 1 = 1
Data 2 = 2
Data 3 = 5
Data 4 = 8

24. Write a program in C to insert a node at the end of a circular linked list.
Test Data and Expected Output :

Input the number of nodes : 3
Input data for node 1 : 2
Input data for node 2 : 5
Input data for node 3 : 8

Data entered in the list are :
Data 1 = 2
Data 2 = 5
Data 3 = 8
Input the data to be inserted : 9

After insertion the new list are :
Data 1 = 2
Data 2 = 5
Data 3 = 8
Data 4 = 9

25. Write a program in C to insert a node at any position in a circular linked list.
Test Data and Expected Output :

Input the number of nodes : 3
Input data for node 1 : 2
Input data for node 2 : 5
Input data for node 3 : 8

Data entered in the list are :
Data 1 = 2
Data 2 = 5
Data 3 = 8
Input the position to insert a new node : 3
Input data for the position 3 : 7

After insertion the new list are :
Data 1 = 2
Data 2 = 5
Data 3 = 7
Data 4 = 8

26. Write a program in C to delete node from the beginning of a circular linked list.

Test Data :

Input the number of nodes : 3

Input data for node 1 : 2

Input data for node 2 : 5

Input data for node 3 : 8

Expected Output :

Data entered in the list are :
Data 1 = 2
Data 2 = 5
Data 3 = 8

The deleted node is -> 2
After deletion the new list are :
Data 1 = 5
Data 2 = 8

27. Write a program in C to delete a node from the middle of a circular linked list.

Test Data and Expected Output :

Input the number of nodes : 3
Input data for node 1 : 2
Input data for node 2 : 5
Input data for node 3 : 8

Data entered in the list are :
Data 1 = 2
Data 2 = 5
Data 3 = 8

Input the position to delete the node : 3

The deleted node is : 8
After deletion the new list are :
Data 1 = 2
Data 2 = 5

28. Write a program in C to delete the node at the end of a circular linked list.
Test Data and Expected Output :

Input the number of nodes : 3
Input data for node 1 : 2
Input data for node 2 : 5
Input data for node 3 : 8

Data entered in the list are :
Data 1 = 2
Data 2 = 5
Data 3 = 8

The deleted node is : 8
After deletion the new list are :
Data 1 = 2
Data 2 = 5

29. Write a program in C to search an element in a circular linked list
Test Data and Expected Output :

Circular Linked List : Search an element in a circular linked list :

Input the number of nodes : 3
Input data for node 1 : 2
Input data for node 2 : 5
Input data for node 3 : 9

Data entered in the list are :
Data 1 = 2
Data 2 = 5
Data 3 = 9
Input the element you want to find : 5
Element found at node 2

Part II: Advanced operators in List

Note: All these following exercises should be done with three type of linked lists separately.

1. **Insert in order** Given a linked list of integers sorted from smallest (at the head end) to largest, and a pointer to a single node containing an integer, insert the node in the linked list so that it remains sorted.
2. **Cumulative sum** Given a null-terminated linked list, in, create a new null-terminated linked, list out, of the same length, such that node *i* of out contains the sum of the data in in's nodes up to and including node *i*of list in. Detect heap exhaustion and report it by setting a boolean variable.
3. **Delete last node** Given a nonempty list, delete the last node and set the new last link to null.
4. **Deal** Given a null terminated linked list, rearrange its nodes into two lists: <first node, third node, fifth node, ...> and <second node, fourth node, sixth node, ...>. Do not allocate any new nodes.
5. **Rifle Shuffle** Given two null terminated linked lists, combine their nodes so that the nodes of the new list alternate between those of the two original nodes: <first node of first list, first node of second list, second node of first list, second node of second list, ... >. Do not allocate any new nodes.
6. **Catenate** Given two null-terminated linked lists headed by left and right, set the last link of the left list to point to the right list, thus joining them into one list. Do not allocate any new nodes.
7. **Reverse** Given a null-terminated linked lists headed reverse the order of its nodes. Do not allocate any new nodes.
8. **Read a list** Given a nonnegative integer *i*, read *i* integers from the input and build a null-terminated linked list such that the first integer in the input is the first node and so forth. Detect heap exhaustion and report it by setting a boolean variable.
9. **List to int** Given a null-terminated linked list of integers from 0 to 9 (inclusive) representing a nonnegative integer in decimal (least significant digit at the head), compute into an unsigned variable, the integer that it represents. (The empty list

represents 0.) You may assume that the list represents a number that can be represented by an unsigned.

10. **Int to list** Given an non-negative integer , create a null-terminated linked list of integers between 0 and 9 representing the integer (least significant digit first). (0 is represented by an empty list). Set a boolean variable to indicate whether or not space has been exhausted. Hint. In C++ i/10, where i is an integer expression, gives the integer quotient of i divided by 10, and i%10 gives the remainder.