

Implementation Of Single Linked List:

Definition:

In computer science, a linked list is a data structure that consists of a sequence of data records such that in each record there is a field that contains a reference (i.e., a link) to the next record in the sequence.

(or)

Linked list consists of series of nodes. Each node contains the element and a pointer to its successor node. The pointer of the last node points to NULL. Each record of a linked list is often called an element or node. The remaining fields may be called the data, information, value, or payload fields. The field of each node that contains address of the next node is usually called the next link or next pointer.

Types of Linked list:

1. Singly linked list.
2. Doubly linked list.
3. Circular linked list.

Advantages:

1. Length of the list need not be known in advance
2. Optimum use of memory
3. Insertion of element into the list is easier
4. Deletion of element from the list is easier
5. Concatenation of two linked lists is easier
6. Memory is not wasted if you remove an element from the list as it is freed

Singly linked list:

The simplest kind of linked list is a singly-linked list, which has one link per node. This link points to the next node in the list, or to a null value or empty list if it is the final node.

A singly linked list's node is divided into two parts. The first part holds or points to information about the node, and second part holds the address of next node. A singly

linked list travels one way.

Operations on singly linked list:

- Insert the element in the list.
- Delete the element in the list
- Forward traverse.

Advantages of singly linked list:

- Dynamic data structure.
- We can perform deletion and insertion any where in the list.
- We can merge two list easily.

Disadvantages of singly linked list:

Backward traversing is not possible in singly linked list.
Insertion is easy but deletion take some additional time, because disadvantage of backward traversing.

Algorithm steps:

Step1: Create nodes first,last,next,prev and cur then set the value as NULL.

Step 2: Read the list operation type.

step 3: If operation type is create then process the following steps.

1. Allocate memory for node cur.
2. Read data in cur's data area.
3. Assign cur link as NULL.
4. Assign first=last=cur.

Step 4: If operation type is Insert then process the following steps.

1. Allocate memory for node cur.
2. Read data in cur's data area.
3. Read the position the Data to be insert.
4. Availability of the position is true then assign cur's link as first and first=cur.

5. If availability of position is false then do following steps.
 1. Assign next as cur and count as zero.
 2. Repeat the following steps until count less than position.
 - 1 .Assign prev as next
 2. Next as prev of link.
 3. Add count by one.
 4. If prev as NULL then display the message INVALID POSITION.
 5. If prev not equal to NULL then do the following steps.
 1. Assign cur's link as prev's link.
 2. Assign prev's link as cur.

Step 5: If operation type is delete then do the following steps.

1. Read the position .
2. Check list is Empty .If it is true display the message List empty.
3. If position is first.
 1. Assign cur as first.
 2. Assign First as first of link.
 3. Reallocate the cur from memory.
4. If position is last.
 1. Move the current node to prev.
 2. cur's link as Null.
 3. Reallocate the Last from memory.
 4. Assign last as cur.

5. If position is enter Mediate.
 1. Move the cur to required position.
 2. Move the Previous to cur's previous position
 3. Move the Next to cur's Next position.
 4. Now Assign previous of link as next.
 5. Reallocate the cur from memory.

step 6: If operation is traverse.

1. Assign current as first.
2. Repeat the following steps until cur becomes NULL

=====

Implementation Of Double Linked List:
 Definition:

A variant of a linked list in which each item has a link to the previous item as well as the next. This allows easily accessing list items backward as well as forward and deleting any item in constant time.

Also known as two-way linked list, symmetrically linked list.

A doubly linked list is a linked list in which each node has three fields namely data field, forward link (Flink) and backward link (Blink). Flink points to the successor node in the list whereas Blink points to the predecessor node.

Advantages:

- Deletion process is easier
- Additional pointer for previous node is not necessary.
- Backward and forward traverse are possible.

Disadvantages :

More Memory space is required since it has two pointers.

Algorithm steps:

Step 1: Read the list operation.

Step 2: If operation is Create then process the following steps.

1. Create the new node and allocate memory for the new node.
2. Read the data in the new node.
3. Assign Flink and Blink as NULL.
4. Assign current as new.

Step 3: If operation is Insertion do the following steps.

- i) Check insertion at beginning is true then
 1. Create the new node and allocate memory for the new node
 2. Read the data in the new node.
 3. Move the current node to start position
 4. Assign new->Blink =Null
 5. Assign new->Flink=current
 6. Now assign current->Blink=new
- ii) Insertion at between nodes is true then
 1. Create the new node and allocate memory for the new node
 2. Read the data in the new node.
 3. Move the current node required position
 4. Assign new node's Blink=current.
 5. Assign new node's Flink=current->Flink
 6. Assign current node's Flink =new
- iii) Insertion at between nodes is true then
 1. Create the new node and allocate memory for the new node.
 2. Read the data in the new node.
 3. Move the current node to end position.
 4. Assign current node's Flink =new.

5. Assign new's Blink as current.

6. Assign new's Flink as NULL.

7. Move end pointer to new node.

Step 4: If operation is deletion then do the following steps.

i). Check deletion at beginning is true then.

1. Move current pointer to start position.

2. Move the start pointer next node in the link.

3. Assign start's Blink as NULL.

4. Reallocate current node from memory.

ii) Check deletion between two nodes is true

1. Move the current pointer to required position.

2. Assign current's->Blink->Flink as current's Flink.

3. Assign current's->Flink->Blink as current's Blink.

4. Reallocate current from memory.

iii) Check deletion at end is true

1. Move the current pointer to end position.

2. Assign current's->Blink->Flink as Null.

3. Reallocate current from memory.

4. Reallocate current from memory.

Step 5: If the operation is traversing.

i) Check deletion at end is true

1. Move the current pointer to end position.

2. Repeat the following steps until current becomes NULL.

3. Display the data.

4. Current=current->Flink.

ii) If Backward traversing then

1. Move the current to end position.

2. Repeat the following steps until current becomes NULL.

3. Display the data.

4. Current=current->Flink.

=====

Implementation of Circular Linked List

Circular Linked List:

In circular linked list the pointer of the last node points to the first node.

It can be implemented as singly linked list and doubly linked list.

Advantages of Circular Linked List:

1. It allows traversing the list starting at any point.

2. It allows quick access to the first and last records.

3. Circularly doubly linked list allows traversing the list in either direction.

Algorithm Steps:

Step 1: Create the pointers prev, cur ,last.

Step 2: Read the operation type of the list.

Step 3: If the operation type is Insertion then

i) If insertion at beginning is true

1.Create the new node and allocate memory for that node.

2.Read the data in new nodes data part.

3.Assign cur->link = last->link (now cur->points first node)

4.Assign first=cur (move the first to cur)

5.Assign last->link =first (now last -> link points new first node)

ii) If insertion at between any two nodes is true

1.Move the prev pointer to required position

2.Create the new node and allocate memory for that node.

3.Read the data in new nodes data part.

4.Assign cur->link = prev-link (give the link to next node)

5.Assign prev->link =cur.

iii) If insertion at end is true

1.Move the prev is to last position ().

2.Create the new node and allocate memory for that node.

3.Read the data in new nodes data part.

4.Assign cur->link =prev->link

5.Assign prev=cur.

Step 4: If the operation type is deletion then

i) If deletion at beginning is true then

1.Assign first=last->link->link (Move the first to second position)

2.Assign cur=last->link

3.Assign last->link=first

4.Reallocate the cur from memory.

ii) If deletion between any two nodes is true

1.Move the cur to required position.

2.Move the prev to cur predecessor's position.

3.Assign prev->link=cur->link

4.Reallocate cur from memory.

iii). If deletion at ends is true

1.Move the cur to last position.

2.Move the prev to cur's predecessor position

3.Assign prev->link=cur->link

4.Reallocate cur from memory

Step 5: If operation type is traverses

1. Assign cur=first

2. Repeat the process until cur becomes last Cur=cur->link