

Linked List in C Language

Linked List is a non- primitive, linear data structure

What is Linked List in Data Structure: Data के एक Linear Collection को हम सामान्य जीवन में List कहते हैं। किसी भी List में एक या एक से अधिक Data Items होते हैं। हम हमारी आवश्यकतानुसार नए Item List में Add करते हैं और किसी Item को List से Delete करते हैं या किसी Item को Modify करते हैं।

उदाहरण के लिए मानलो कि हमें कुछ सामान खरीदने के लिए एक List बनाते हैं जिनमें निम्न Items हैं जो हमें खरीदने हैं-

Pen
Floppy
CD Bag
Walkman
CD Player
Speaker Set

यदि हम चाहें तो इसमें कुछ और Items भी Add कर सकते हैं जिन्हें हम खरीदना चाहते हैं। निम्न List को देखिए जिसमें तीन नए Items Add किए गए हैं-

Pen
Floppy
CD Bag
Home Theater
Walkman
CD Player
Color TV
Speaker Set
Diary

आप देख सकते हैं कि इस List में हमने तीन नए Items Add किए हैं। मानलो कि हमें इस List में से लिखे गए। CD Player को Change करके DVD Player लेना चाहते हैं तो हमारी इस List को हमें Modify करना होगा। हम इस List को Modify करते हैं तो List निम्नानुसार हो जाती है-

Pen

Floppy

CD Bag

Home Theater

Walkman

DVD Player

Color TV

Speaker Set

Diary

किसी List में Organized Data को Store व Process करना **Data Processing** कहलाता है। Data को Computer Memory में Store करने का एक तरीका Array है। जिस तरह से हम एक Array Linear तरीके से विभिन्न Data Items जो कि आपस में Related होते हैं, Store करते हैं, उसी तरह से Data Items को Memory में Physically Related रूप में Store कर सकते हैं।

किसी **Array** में हर Data Item अगले Index Number पर Store होता है। यानी Data Items Physically भी Memory में Related होते हैं। लेकिन Array की कुछ कमियां हैं।

- किसी Array में यदि किसी Data Item को Delete करना हो या कोई Data Item Insert करना हो, तो Array के लगभग सभी Data Items को Left या Right में Shift करना पड़ता है। ये किसी भी Array Data Structure की पहली कमी है।
- किसी Array में हमें जितने Data Items Insert करने होते हैं हमें Array की उतनी Size को पहले ही Define करना पड़ता है। ऐसे में यदि हमें आवश्यकतानुसार अधिक Data Insert करने हों तो हम Array की Size को बढ़ा नहीं सकते हैं। यानी Array में Store किए जाने वाले Data Permanent होते हैं, इसलिए Array को **Static Data** Structure या **Permanent Data** Structure या **Dense List** कहते हैं। ये Array की दूसरी कमी है।

Related Data Items की किसी List को Memory में Store करने का एक दूसरा तरीका और है जिसे **Linked List** कहते हैं। Linked List में हर Data Items के साथ अगले Data Item का Link होता है। यानी हर अगले Data Item का Address या Pointer पिछले Data Item के साथ होता है। हालांकि **Linked List** भी एक **Linear Data Structure** है, लेकिन Linked List के Data Items क्रमागत Memory Location पर Store नहीं होते हैं। इसलिए किसी List Item को Add या Delete करना सरल होता है।

इस Data Structure में हर Data Item अपने अगले Data Item को Point करता है। यानी हर अगले Data Item का Pointer या Link या Address पिछले Data Item में होता है। इसलिए सभी Data Items आपस में Linked होते हैं। इस तरह के Data Structure को Linked List कहा जाता है। इस Data Structure की अपनी कुछ विशेषताएं और कमियां हैं।

Linked List एक **One - Way** List होती है, जो कि Data Items का **Linear Collection** कहा जाता है।
Linked List के विभिन्न Data Items को **Node** कहा जाता है।

किसी Linked List में हमेशा दो भाग होते हैं। पहले भाग को **Information Part** कहा जाता है और दूसरे Part को **Pointer Part** या **Link Part** कहा जाता है। Information Part में हम विभिन्न Data Items के एक Records को Store कर सकते हैं जबकि Linked Part में हमेशा एक Pointer होता है, जिसमें अगले Record का Address या Pointer Variable होता है। इस Pointer Variable को **START** कहा जाता है। जहां पर List Item का अन्त होता है उस Last Link Field में **NULL** होता है। जब किसी List में कोई Node नहीं होता है तो उसे **NULL List** या **Empty List** कहते हैं और इस स्थिति में **START** में **NULL** होता है।

linked list, नोड्स के समूह से मिलकर बना होता है। प्रत्येक node के दो भाग होते हैं पहला भाग data का होता है और दूसरा pointer होता है। linked list का pointer भाग अगले node के address को hold किये रहता है।

nodes का प्रयोग डेटा को संग्रहित करने के लिए किया जाता है।

linked list एक ऐसा डेटा स्ट्रक्चर होता है जिसकी length को run-time में बढ़ाया या घटाया जा सकता है। अर्थात यह dynamic होता है।

linked list का प्रयोग tree तथा graph को बनाने के लिए किया जाता है।

Types of linked list in hindi in hindi:- Linked list निम्नलिखित तीन प्रकार के होते हैं:-

1:-Single linked list:- इसमें one-way direction होता है तथा single linked list के प्रत्येक node में दो fields होते हैं:-

1:-पहला वह field होता है जहां डेटा स्टोर रहता है।

2:-दूसरा pointer या लिंक होता है।

data	next
------	------

Figure 1: Element of a singly linked list

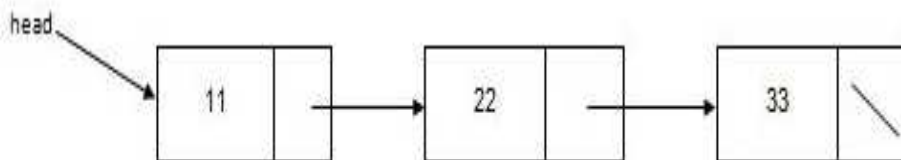
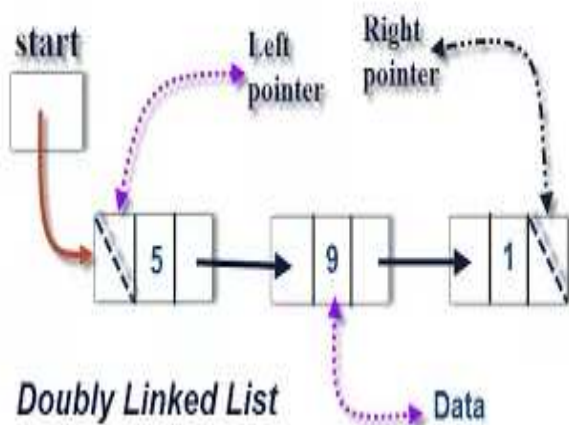


Figure 2: A singly linked list

2:-Doubly linked list:- इसमें two-way direction होता है। doubly linked list के प्रत्येक node में तीन भाग होते हैं:-

- 1:-पहले भाग में डेटा स्टोर रहता है।
- 2:-दूसरा भाग अगले नोड के लिए link होता है।
- 3:-तीसरा भाग पिछले के लिए लिंक होता है।



3:-Circular linked list:- Circular linked list में प्रत्येक नोड circle के रूप में जुड़े रहते हैं। circular Linked list के अंत में NULL वैल्यू नहीं होती है।

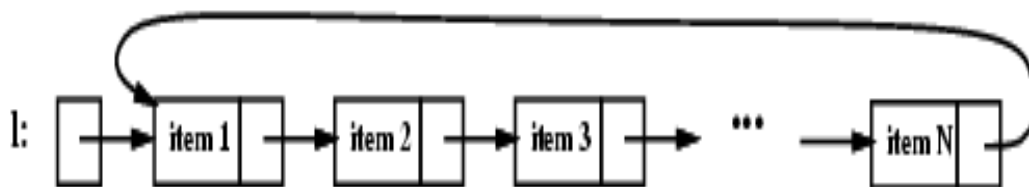
इसमें अंतिम नोड, पहले नोड के address को contain किये हुए रहता है अर्थात पहला और अंतिम नोड adjacent होते हैं।

Circular linked list के दो प्रकार होते हैं:-

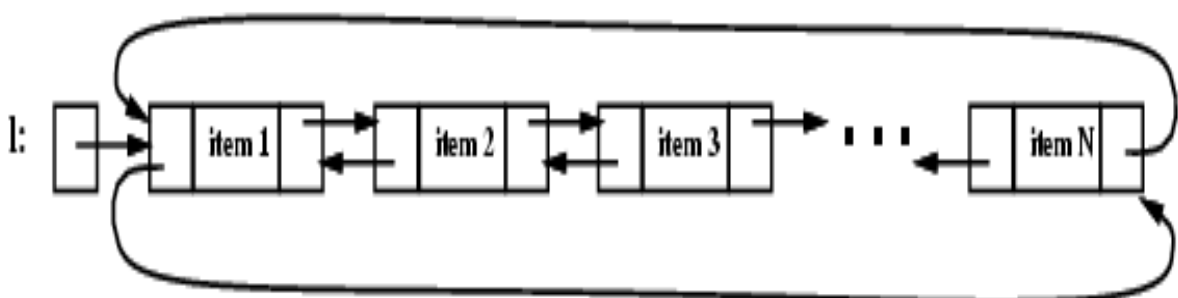
1:-Single circular linked list

2:-Doubly circular linked list.

Circular, singly linked list:



Circular, doubly linked list:



circular.gif

Advantages and Disadvantages of Linked List

Advantages of Linked List

Dynamic Data Structure

Linked list is a dynamic data structure so it can grow and shrink at runtime by allocating and de allocating memory. So there is no need to give initial size of linked list.

Insertion and Deletion

Insertion and deletion of nodes are really easier. Unlike array here we don't have to shift elements after insertion or deletion of an element. In linked list we just have to update the address present in next pointer of a node.

No Memory Wastage

As size of linked list can increase or decrease at run time so there is no memory wastage. In case of array there is lot of memory wastage, like if we declare an array of size 10 and store only 6 elements in it then space of 4 elements are wasted. There is no such problem in linked list as memory is allocated only when required.

Implementation

Data structures such as stack and queues can be easily implemented using linked list.

Disadvantages of Linked List

Memory Usage

More memory is required to store elements in linked list as compared to array. Because in linked list each node contains a pointer and it requires extra memory for itself.

Traversal

Elements or nodes traversal is difficult in linked list. We can not randomly access any element as we do in array by index. For example if we want to access a node at position n then we have to traverse all the nodes before it. So, time required to access a node is large.

Reverse Traversing

In linked list reverse traversing is really difficult. In case of [doubly linked list](#) its easier but extra memory is required for back pointer hence wastage of memory.