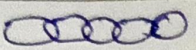
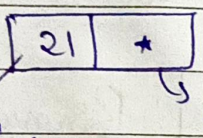
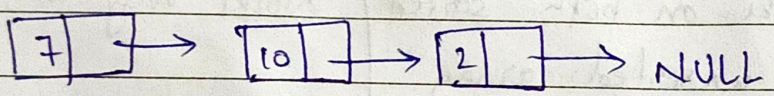
Linked List

→ chain



→ Every element in a linked list is called node and consists of two parts, the data part, and the pointer part.





Non Contigous memory locations.

Why linked lists? (vs Array)

→ Memory and the capacity of an array remain fixed, while in linked list we can keep adding and removing elements without any capacity constraint.

Drawbacks

→ Extra memory space for pointers is required.

→ Random access is not allowed as elements are not stored in contigous memory locations.

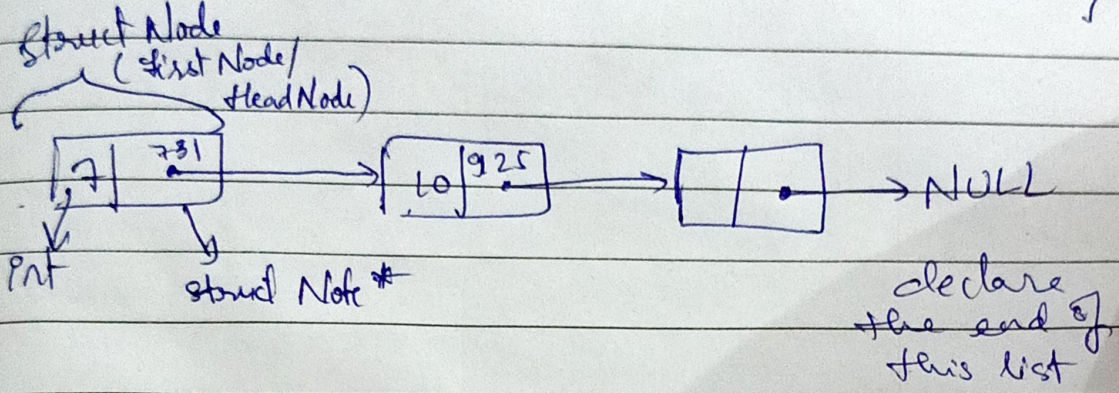
Benefit

→ Fast Insertion, Fast deletion.

Implementation :-

Using a structure in C language

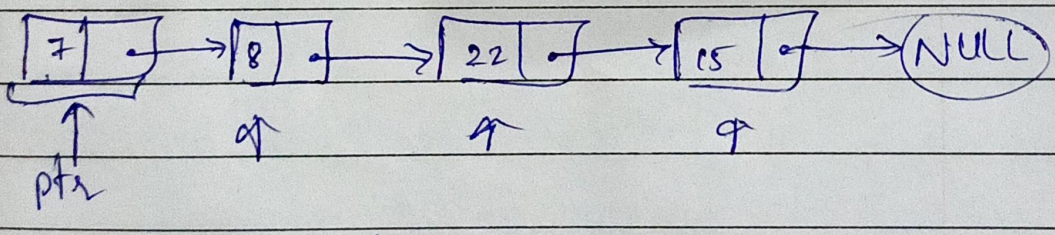
```
struct Node {
    int data;
    struct Node * next;    { Self referencing structure }
};
```



head = (struct Node*) malloc (sizeof (struct Node))
↙ struct pointer.

head → data = 7
head → next = 10

Traversal



Complexity O(n)