Linked List

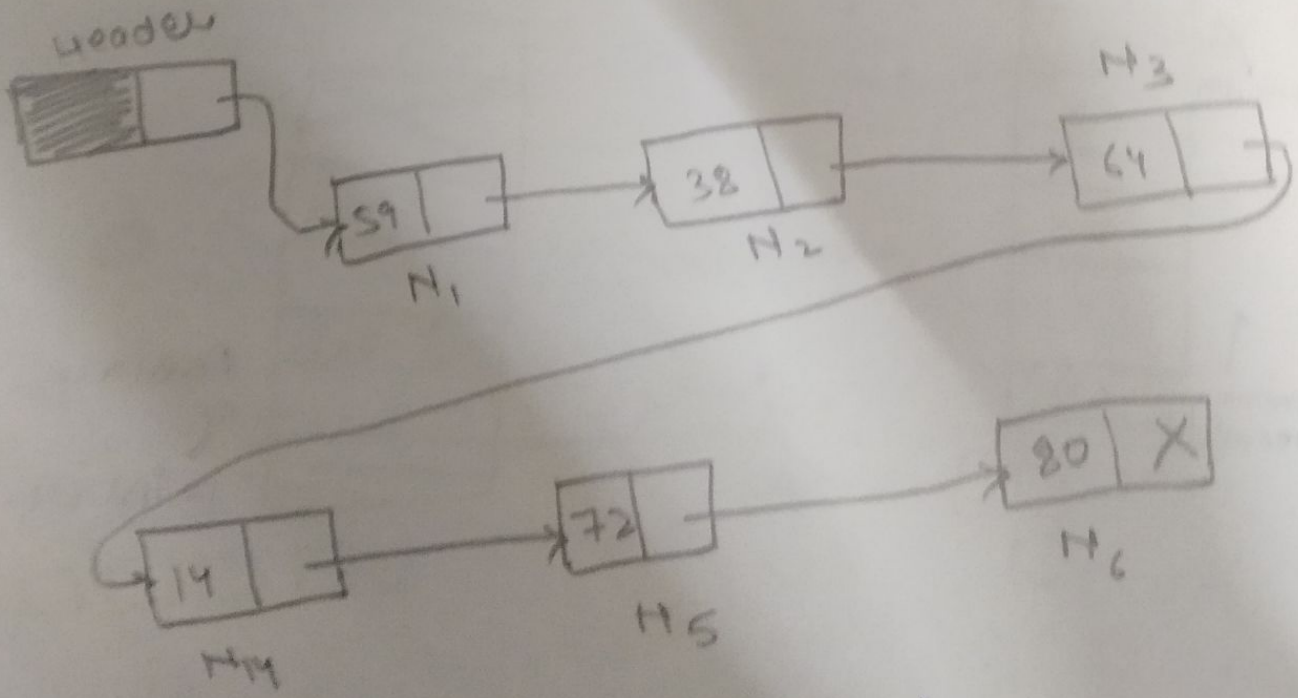
→ A linked list is an ordered collection of finite homogenous data elements called nodes wherein the linear order is maintain by means of link or pointer.

→ There are mainly 3 types of linked list

1. Single linked list
2. Circular linked list
3. Double linked list

Single linked list

→ Each node contains only one link which points to the subsequent node in the list.



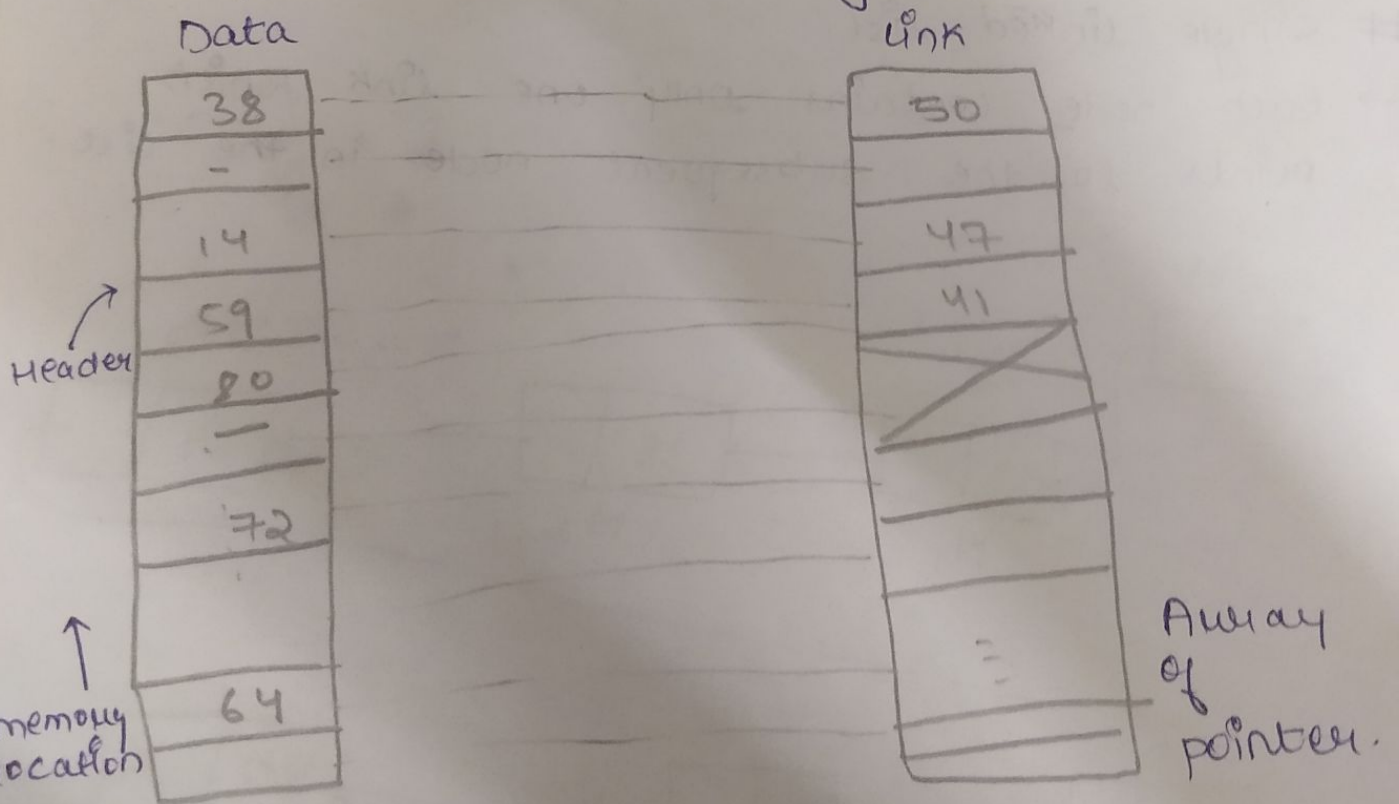
A single linked list with 6 nodes.

Static Representation:

→ Two arrays are maintained: one array for data and the other for links.

→ Two parallel arrays of equal size are allocated which should be sufficient to store the entire linked list.

→ This ~~condn~~ contradicts the idea of the linked list (Non-contagious location of elements). Some programming language. Such a representation is the only representation to manage a linked list.



Dynamic Representation

→ The efficient ways of representing a linked list is using the free pool of storage. In this method, there is a memory bank (collection of free memory space) and a memory manager (a program). During the creation of the linked list, whenever a node is required the request is placed to the memory manager; the memory manager will the search the memory bank for the block requested and, if found, grants the desired block to the caller. Again, there is also another program called garbage collector; it plays whenever a node is no more in use; it basically a list of memory space which is available to the program.

→ It returns the unused node to the memory bank.

Such a memory managment is known as dynamic memory management.

The dynamic representation of linked list uses the dynamic memory managment policy.

Note :→ In python it take place auto-matically, as a user we don't have to take care about it.

(2) In c, c++ we have take care to return the unused node garbage collector

Operations on a single linked list.

(1) Traversing the list
(2) Inserting a node into the list
(3) Deleting a node from the list
(4) Copying the list to make a duplicate of it.
(5) merging the linked list with another one to make a large list.
(6) searching for an element in the list.

Why Linked List Over Array:

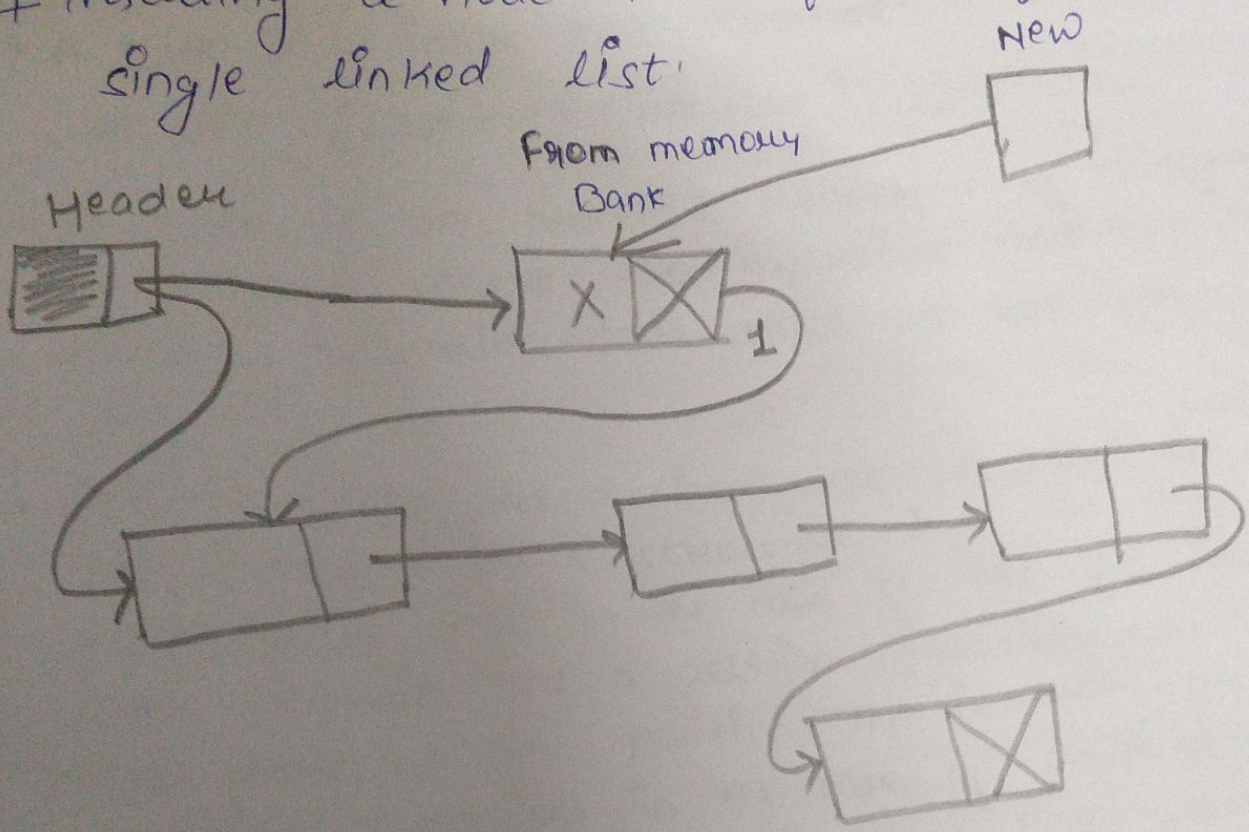
→ In case of a array, data are stored in contiguous memory location, so insertion & deletion operation are quite time - consuming.

→ In case of array amount of memory required for a array must be known before hand. Once it is allocated we cannot expand the size.

→ A linked list uses a dynamic memory managment scheme; memory allocation is decided during the run time as and when required. Also if a memory is not required any more, it can be returned to the free storage space.

→ Linked list consume more space than space req. for actual data as we have to maintain the links among the nodes.

Inserting a Node in the front of a single linked list.



At the End.