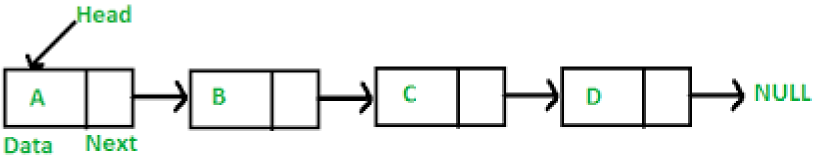
LINKED LIST:

A linked list is a sequence of data structures, which are connected together via links. Linked List is a sequence of links which contains items. Each link contains a connection to another link. Linked list is the second most-used data structure after array.



Applications of Linked Lists:

Few of the main applications of Linked Lists are:

- Linked Lists let us implement queues, stacks, graphs, etc.
- Linked Lists let us insert elements at the beginning and end of the list.
- binary tree, skip, unrolled linked list, hash table, etc.

Advantage of linked list:

The main advantage of a linked list is that we do not need to specify a fixed size for the list. The more elements we add to the chain, the bigger the chain gets.

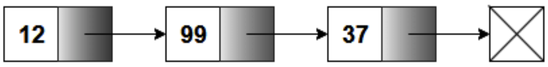
Drawbacks of the linked list:

Some of the important drawbacks of the linked list are given below:

- Random access is not allowed. We need to access elements sequentially starting from the first node. So we cannot perform a binary search with linked lists.
- More memory space for a pointer is required with each element of the list.
- Poor locality, the memory used for the linked list is scattered around in a mess.

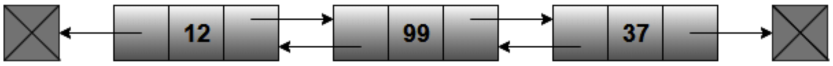
There are multiple types of Linked Lists available:

- **Singly Linked List :**The singly linked list includes nodes which contain a data field and next field. The next field further points to the next node in the line of nodes.



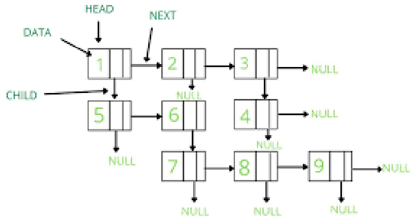
Singly Linked List

- **Doubly Linked List :**The doubly linked list includes a pointer (link) to the next node as well as to the previous node in the list. The two links between the nodes may be called "forward" and "backward, "or "next" and "prev (previous)."

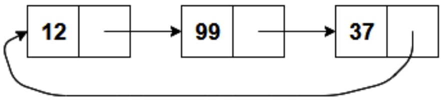


Doubly Linked List

- **Multiply Linked List :**In a multiply linked list, each node consists of two or more link fields. Each field is used to join the same set of records in a different order of the same set, e.g. "by name, by date of birth, by the department, etc.".



- **Circular Linked List :**In the last node of a linked list, the link field often contains a null reference. Instead of including a null pointer at the end of the list, the last node in circular linked lists includes a pointer pointing to the first node.



Circular Linked List

C Programing project:

- Assume you are medical shop owner
- and also you are a C expert programmer
- Write a C program to maintain medicine list in the shop.
- Use linked list to create new medicines/delete medicines/traverse the list.
- At the end of each day before closing the shop, save all the linked list data to a file in the hard disk.
- When the shop is opened, read the file contents and put into a linked list and carry out linked list operations.
- Unique identification, manufacturer’s name, Manufacturing dates, expiry date etc.

Options:
create a new medicine
delete a medicine
View a particular medicine
Update quantity for a particular medicine
Assume medicine name is unique

Linked lists & file operations are the main data structures to be used
Another option is "Another option, close the program"

CODE:

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
struct node{
    int id,price,quantity;
    char medicinename[100],company[100],mfg_date[11],info[5000];
    struct node*p;};
void search(struct node*start);
struct node*insert_end(struct node*);
struct node*del_ele(struct node*);
void update(struct node*);
struct node*del_begin(struct node*start){
    struct node*p;
    if(start==NULL){printf("UNDER FLOW");return start;}
    else{p=start;
    p=p->p;
    free(start);
    return p;}}
int main(){
    struct node*start=NULL,*p;
    int x;
    printf("enter choice\n1.create a new medicine\n2.delete a medicine\n3.view particular
medicine\n4.update quantity for particular medicine\n5.to exit");
    scanf("%d",&x);
    while(x!=5){
    switch(x){
        case 1:start=insert_end(start);
            break;
        case 2:start=del_ele(start);
            break;
        case 3:search(start);
            break;
        case 4:update(start);
            break;
        default:printf("enter correct choice");
            break;}
        printf("enter choice\n1.create a new medicine\n2.delete a medicine\n3.view
particular medicine\n4.update quantity for particular medicine\n5.to exit");
    scanf("%d",&x);}
    printf(".........exit..........");}

void search(struct node*start){
    char s[100];int c=0;
    printf("enter medicine to find:");
    scanf("%s",&s);
    while(start!=NULL){if(!(strcmp(s,start->medicinename))){
        printf("medicine name:%s\nmedicine quantity:%d\nmedicine price:%d\n",start-
>medicinename,start->quantity,start->price);
                c++;break;}
            start=start->p;}
    if(c==0)printf("\nnot found\n");}

struct node* insert_end(struct node*start){
    char val[100];
    int x;
    struct node*ptr=NULL,*p=start;
    ptr=(struct node*)malloc(sizeof(struct node));
    printf("\nenter medicine name to insert:");
    scanf("%s",&(ptr->medicinename));
    printf("\nenter medicine id:");
    scanf(" %d",&x);
    ptr->id=x;
    printf("\nenter medicine quantity:");
    scanf("%d",&x);
    ptr->quantity=x;
    printf("\nenter medicine info:");
    scanf("%s",&(ptr->info));

    printf("\nenter medicine price:");
    scanf("%d",&x);
    ptr->price=x;
    printf("\nenter medicine mfg date:");
    scanf("%s",&(ptr->mfg_date));

    printf("\nenter medicine company:");
    scanf("%s",&(ptr->company));
    if(start==NULL){
    ptr->p=NULL;return ptr;}
    if(ptr==NULL)printf("overflow");
    else{
        ptr->p=NULL;
        while(start->p!=NULL)start=start->p;
        start->p=ptr;}return p;}
struct node*del_ele(struct node*start){
    char val[100];int c=0;
    printf("enter medicine to delete:");
    scanf("%s",&val);
    if(start==NULL){printf("list is empty\n");return start;}
    else{
    struct node*ptr,*ptr1;
    ptr=ptr1=start;
    while(ptr!=NULL){if(!(strcmp(val,ptr->medicinename))){c++;break;}
            ptr1=ptr;
            ptr=ptr->p;}
    if(c==0)printf("element not found:\n");
    else if(c==1){ptr=del_begin(start);return ptr;}
    else{ptr1->p=ptr->p;
        free(ptr);}return start;}}
void update(struct node*start){
    struct node*ptr;
    ptr=start;
    char a[100];int x=-1;
```

```
        printf("enter medicine to update quantity:");
        scanf("%s",&a);
        while(ptr!=NULL){printf("%s",ptr->medicinename);
                if(!(strcmp(a,ptr->medicinename))){
                        printf("enter updated quantity:");
                        scanf("%d",&x);
                        ptr->quantity=x;}
                ptr=ptr->p;}
        if(ptr==NULL)printf("element not found to update quantity\n");}
```

OUTPUT:

```
enter choice
1.create a new medicine
2.delete a medicine
3.view particular medicine
4.update quantity for particular medicine
5.to exit1

enter medicine name to insert:cough

enter medicine id:32

enter medicine quantity:650

enter medicine info:illness

enter medicine price:20

enter medicine mfg date:12-3-2022

enter medicine company:pharma
enter choice
1.create a new medicine
2.delete a medicine
3.view particular medicine
4.update quantity for particular medicine
5.to exit3
enter medicine to find:cough
medicine name:cough
medicine quantity:650
medicine price:20
enter choice
1.create a new medicine
2.delete a medicine
3.view particular medicine
4.update quantity for particular medicine
5.to exit4
enter medicine to update quantity:750
coughelement not found to update quantity
enter choice
```

```
1.create a new medicine
2.delete a medicine
3.view particular medicine
4.update quantity for particular medicine
5.to exit3
enter medicine to find:cough
medicine name:cough
medicine quantity:750
medicine price:20
enter choice
1.create a new medicine
2.delete a medicine
3.view particular medicine
4.update quantity for particular medicine
5.to exit
enter choice
1.create a new medicine
2.delete a medicine
3.view particular medicine
4.update quantity for particular medicine
5.to exit
2
enter medicine to delete:cough
enter choice
1.create a new medicine
2.delete a medicine
3.view particular medicine
4.update quantity for particular medicine
5.to exit3
enter medicine to find:cough

not found
enter choice
1.create a new medicine
2.delete a medicine
3.view particular medicine
4.update quantity for particular medicine
5.to exit5
.........exit..........(base) student@jessy:~/Desktop$
```