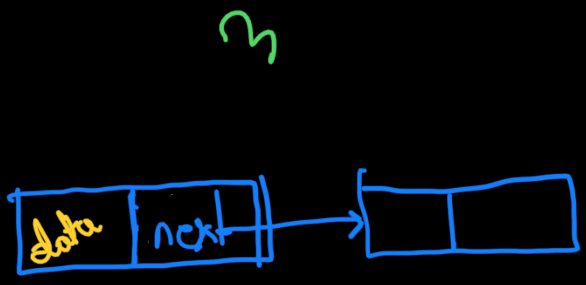
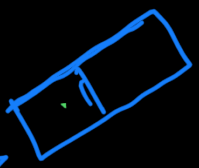
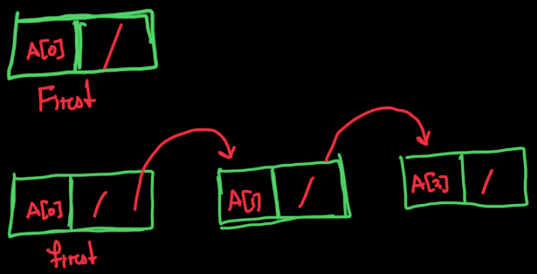
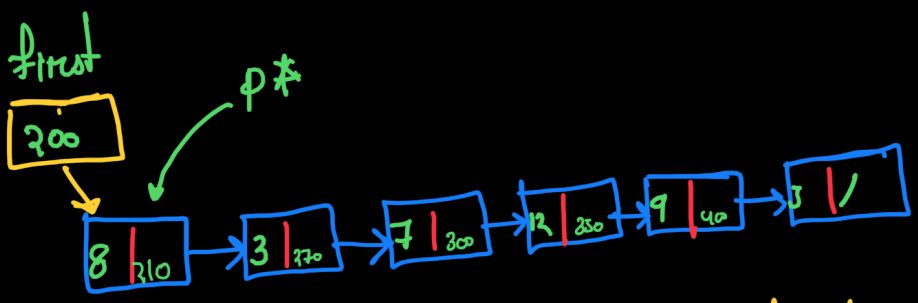
```
//
//  main.cpp
//  all linked list class
//
//  Created by KAIBALYA SAHOO on 20/03/2021.
//

#include <iostream>
using namespace std;
class Node
{
public:
    int data;
    Node *next;

};

class LinkdList
{
private:
    Node *first;
public:
    LinkdList()
    {
        first=NULL;
    }
    LinkdList(int A[],int n);
    ~LinkdList();
    void display();
    void insert(int index,int n);
    int Delete(int index);
    int Length();

};

LinkdList::LinkdList(int A[],int n)
{
    Node *last,*t;
    int i=0;
    first=new Node;
    first->data=A[0];
    first->next=NULL;
    last=first;
    for( i=1;i<n;i++)
    {
        t=new Node;
```







```
        t->data=A[i];
        t->next=NULL;
        last->next=t;
        last=t;
    }

}

LinkdList::~LinkdList()
{
    Node *p=first;
    while(first)
    {
        first=first->next;
        delete p;
        p=first;
    }

}

void LinkdList::display()
{
    Node *p=first;
    while(p)
    {

        cout<<p->data<<"";
        p=p->next;
    }
    cout<<endl;
}

int LinkdList::Length()
{

    Node *p=first;
    int len=0;
    while(p)
    {
        len++;
        p=p->next;
    }
    return len;
}
void LinkdList:: insert(int index, int x)
{
    Node *t,*p=first;
    if(index<0 || index>Length())
        return;
    t=new Node;
    t->data=x;
```



temporary pointer

moving node to next node

showing the data

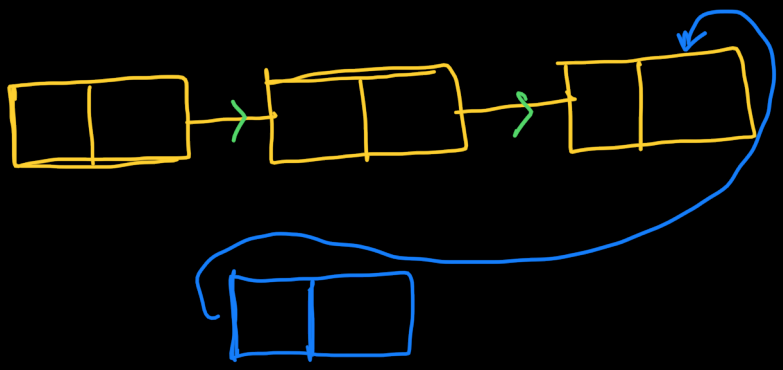
* we don't use Loop Because we don't know that how much we have to go

→ you have to traverse all through element.

→ you can also perform (Sum, Searching maximum,) O(n)

→ To check the validity of the Index. If it is Inbetween 0 to size

```
    t->next=NULL;
    if(index==0)
    {
        first=t;
    }
    else
    {
        for(int i=0;i<index-1;i++)
        {
            p=p->next;

        }
        t->next=p->next;
        p->next=t;
    }


}

int LinkdList::Delete(int index)
{
    Node *p ,*q=NULL;
    int x=-1;
    if(index<1 || index>Length())
        return -1;
    if(index==1)
    {
        p=first;
        first=first->next;
        x=p->data;
        delete p;




    }
    return x;
}









int main()
{
    int A[]={1,2,3,4,5};
    LinkdList l(A,5);
    l.display();
    cout<<l.Delete(1)<<endl;
    l.display();
```



}

* Recursive function of Linked list to traverse through the elements.

```
int Count(struct Node *p)
{
  int c=0
  while(p!=0)
  {
    p=p->next;
    c++;
  }
return c;
}
```

```
int Count(struct Node *p)
{
  if(p==0)
    return 0;
  else
    return Count(p->next)
           +1;
}
```