Linked List CPP

```cpp
#include <iostream>

using namespace std;

class Node
{
public:
    int data;
    Node *next;
};

class LinkedList
{
private:
    Node *first;
public:
    LinkedList(){first=NULL;}
    LinkedList(int A[],int n);
    ~LinkedList();

    void Display();
    void Insert(int index,int x);
    int Delete(int index);
    int Length();

};

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

    while(p)
    {
        cout<<p->data<<" ";
        p=p->next;
    }
    cout<<endl;
}
int LinkedList::Length()
{
    Node *p=first;
    int len=0;

    while(p)
    {
        len++;
        p=p->next;
    }
```

```
    return len;
}

void LinkedList::Insert(int index,int x)
{
    Node *t,*p=first;

    if(index <0 || index > Length())
        return;
    t=new Node;
    t->data=x;
    t->next=NULL;

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

int LinkedList::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;
```

```
    }
    else
    {
        p=first;
        for(int i=0;i<index-1;i++)
        {
            q=p;
            p=p->next;
        }
        q->next=p->next;
        x=p->data;
        delete p;
    }
    return x;
}


int main()
{
    int A[]={1,2,3,4,5};
    LinkedList l(A,5);

    l.Insert(3,10);

    l.Display();

    return 0;
}
```