DOUBLE LINKED LIST

```
#include<iostream>
using namespace std;

class node{
    public:
    int data;
    node* prev;
    node* next;
    node(int val){
        data=val;
        prev=NULL;
        next=NULL;
    }
};

void insertAtHead(node* &head,int val){
    node* n=new node(val);
    n->next=head;
    if(head!=NULL){

    head->prev=n;
    }
    head=n;

}


void insertAtTail(node* &head,int val){
    node* n=new node(val);
    node*temp=head;

    if(head==NULL){
        insertAtHead(head,val);
        return;
    }

    while(temp->next!=NULL){
        temp=temp->next;
    }
    temp->next=n;
    n->prev=temp;
}


void display(node* &head){
    node*temp=head;
    while(temp!=NULL){
        cout<<temp->data<<"->";
        temp=temp->next;
    }
    cout<<"Null"<<endl;
}

void deleteAtHead(node* &head){
    node* todelete=head;
    head=head->next;
    head->prev=NULL;
    delete todelete;
}
void deletion(node* &head,int pos){

    if(pos==1){
        deleteAtHead(head);
        return;
    }
    node* temp=head;
    int count=1;

    while(temp!=NULL && count!=pos){
        temp=temp->next;
        count++;
    }
    temp->prev->next=temp->next;
    if(temp->next!=NULL){

    temp->next->prev=temp->prev;
    }
    delete temp;
}
////////// function to find Length of Linked list///////
int length(node* head){
    node*temp=head;
    int l=0;
    while(temp!=NULL){
        l++;
        temp=temp->next;
    }
    return l;
}

int main(){

    // node* n1=new node(1);
    // node* head=n1;
    // insertAtTail(head,2);
    //     insertAtTail(head,3);
    //         insertAtTail(head,4);
    // insertAtTail(head,5);
    // insertAtTail(head,6);
    // insertAtTail(head,7);
    // insertAtTail(head,8);
    // insertAtTail(head,9);
    //     insertAtTail(head,10);
    // display(head);

    // deletion(head,5);
    // display(head);
    // deletion(head,1);
    // display(head);
    // cout<<"Length: "<<length(head)<<endl;
    node* head=NULL;
    int arr[10]={2,3,9,10,36,333,111,21,777,1000};
    for(int i=0;i<10;i++){
        insertAtHead(head,arr[i]);
    }
    display(head);
     return 0;
}
```

Output: 1000->777->21->111->333->36->10->9->3->2->Null