

Linklist1 assignment1

Abhishek.21402@knit.ac.in

Q.1 In a singly linked list, deletion of data requires modification of how many pointers?

Ans 2(two);

Q.2 Predict the output for linked list = 1->2->3->4->5:

```
#include<bits/stdc++.h>
using namespace std;
class node{
public :
int data;

node *next;
node(int n){
data = n;
next = NULL;
}
};
class linkedlist{
public:
node *head,*tail;
int size=0;
linkedlist(){
head = NULL;
tail = NULL;
}
void insert(int val){
node* temp=new node(val);
if(size==0)tail=head=temp;
else{
temp->next=head;
head=temp;
}
size++;
}
void traverse() {
while(head and head->next) {
```

```

    cout << head->data <<" ";
    head= head->next->next;
}
}
};

int main(){
    linkedlist ll;
    ll.insert(1);
    ll.insert(2);
    ll.insert(3);
    ll.insert(4);
    ll.insert(5);

    ll.traverse();
}

```

Q.3Q3. Implement a Linked List class.

```

#include<iostream>
using namespace std;
class Node
{
public:
    int val;
    Node* next;
    Node(int val){
        this->val=val;
        this->next=NULL;
    }
};

class Linklist{
public:
    Node* head;
    Node* tail;
    int size;
    Linklist(){
        head=tail=NULL;
        size=0;
    }
    void Inserttail(int val){
        Node* temp=new Node(val);
        if(size==0)head=tail=temp;
    else{
        tail->next=temp;
        tail=temp;
    }
}

```

```

}
size++;

}
void Inserthead(int val){
    Node* temp=new Node(val);
    if(size==0)head=tail=temp;
else{
    temp->next=head;
    head=temp;
}
size++;
}
void Insertidx(int idx,int val){
    if(idx<0||idx>size){
        cout<<"invalidindex"<<endl;
        return;
    }
    Node* k=new Node(val);

    if(size==0)head=tail=k;
    else if(idx==0)Inserthead(val);
    else if (idx==size-1)Inserttail(val);
else{
    Node*t=new Node(val);
    Node*temp=head;
    for(int i=0;i<=idx-1;i++){
        temp=temp->next;
    }
    t->next=temp->next;
    temp->next=t;
}
size++;
}
void searchidx(int idx){
    if(idx<0||idx>size){
        cout<<"invalidindex"<<endl;
        return;
    }

    Node*temp=head;
    for(int i=0;i<=idx;i++){
        temp=temp->next;
    }
    cout<<temp->val<<endl;
}

```

```

    }

    void deletetail(){
        Node*temp=head;
        while(temp->next!=NULL){
            temp=temp->next;
        }
        temp->next=NULL;
        tail=temp;
        size--;
    }

    void display(){
        Node*temp=head;
        while(temp!=NULL){
            cout<<temp->val<<" ";
            temp=temp->next;
        }
        cout<<endl;
    }
};

```

```

int main() {
    Linklist ll;
    ll.Inserttail(10);
    ll.Inserttail(20);
    ll.Inserttail(40);
    ll.display();
    ll.Inserthead(22);
    ll.display();
    cout<<ll.size<<endl;
    ll.Insertidx(2,100);
    ll.display();
    ll.searchidx(2);
    //ll.deletetail();
    // ll.deletetail();
    ll.deletetail();
    ll.display();
}

```

}