



ASSIGNMENT 5

NAME: Ujjwal Pant

ROLL NUMBER: 1024030370

Q1

```
#include <iostream>
#include<vector>
using namespace std;
class node{
    public:
    int data;
    node *next;
    public:
    node(int data1,node* next1){
        data=data1;
        next=next1;
    }
    node(int data){
        this->data=data;
        next=nullptr;
    }
};
node* convertvectoll(vector <int> & arr){
    node* head= new node(arr[0]);
    node *mover=head;
    for (int i=1; i<arr.size();i++){
        node* temp= new node(arr[i]);
```

```

        mover->next=temp;
        mover=mover->next; //mover=temp;
    }
    return head;
}
void traversell(node* head){
    node* temp=head;
    while(temp!=nullptr){
        cout<<temp->data<<" ";
        temp=temp->next;
    }
    cout<<endl;
}
int lenghtofll(node * head){
    node * temp=head;
    int cnt=0;
    while(temp){// while temp is valid ie not nullptr
        cnt++;
        temp=temp->next;
    }
    return cnt;
}
int searcher(node *head,int ele){
    node*temp=head;
    int fnd=0;
    while(temp){
        if(temp->data==ele){
            fnd=1;
            break;
        }
        temp=temp->next;
    }
}

```

```

    return fnd;
}
node* removeshead(node *head){
    if (head==NULL) return head;
    node *temp=head;
    head=head->next;
    delete temp;
    return head;
}
node* deletetail(node *head){
    if(head==NULL|| head->next==NULL) return NULL;
    node *temp=head;
    while(temp->next->next!=nullptr){
        temp=temp->next;
    }
    free(temp->next);
    temp->next=nullptr;
    return head;
}
node* deleter(node* head,int k){
    if(head==NULL) return head;
    node* temp=head;
    if(k==1){
        head=head->next;
        free(temp);
        return head;
    }
    int cnt=0;
    node* prev=NULL;
    while(temp){
        cnt++;

```

```

        if(cnt==k){
            prev->next=prev->next->next;
            free(temp);
            break;
        }
        prev=temp;
        temp=temp->next;
    }
    return head;
}

node* deleterval(node* head,int k){
    if(head==NULL) return head;
    node* temp=head;
    if(k==head->data){
        head=head->next;
        free(temp);
        return head;
    }
    int cnt=0;
    node* prev=NULL;
    while(temp){
        cnt++;
        if(temp->data==k){
            prev->next=prev->next->next;
            free(temp);
            break;
        }
        prev=temp;
        temp=temp->next;
    }
    return head;
}

```

```

node* insertion(node* head,int ele,int pos){
    if(head==NULL){
        if(pos==1) return new node(ele);
        else return head;
    }
    if(pos==1){
        node *temp=new node(ele,head);
        return temp;}
    int cnt=0;
    node* prev=NULL;
    node*temp=new node(ele);
    node* iter=head;
    while(iter){
        cnt++;
        if(cnt==pos){
            prev->next=temp;
            temp->next=iter;
            break;
        }
        prev=iter;
        iter=iter->next;
    }

    return head;
}

node* insertlast(node *head, int ele){
    if(head==NULL){return new node(ele,nullptr);}
    node* temp=head;
    node* insert=new node(ele,nullptr);
    while(temp->next!=nullptr){
        temp=temp->next;
    }
}

```

```

    temp->next=insert;
    return head;
}
node * reverser(node *head){

    node *temp=head;
    node *prev=NULL;
    node *fore=NULL;
    while(temp){
        fore=temp->next;
        temp->next=prev;
        prev=temp;
        temp=fore;
    }
    return prev;
}
int main()
{
    vector<int> arr={10,2,3,4};
    node* head=convertvectoll(arr);

    // cout<<head->data<<endl;
    // traversell(head);
    // cout<<lenghtofll(head)<<endl;
    // cout<<searcher(head,7)<<endl;
    // head=removeshead(head);
    // cout<<head->data<<endl;
    // traversell(head);
    // head=deletetail(head);
    // head=deleterval(head,4);
    // traversell(head);
    // head=insertion(head,7,5);

```

```

// traversell(head);
// head=insertlast(head,11);
head=reverser(head);
traversell(head);

return 0;
}

```

Q2

```

#include <iostream>
#include <bits/stdc++.h>
using namespace std;
class node{
public:
int data;
node *next;
public:
node(int data1,node* next1){
    data=data1;
    next=next1;
}
node(int data){
    this->data=data;
    next=nullptr;
}
};
node* counterremover(node* head,int key){

```

```

if(head==NULL){return head;}
int cnt=0;
node* temp=head;
node* prev=NULL;
while(temp){
    if(temp->data==key&& temp==head){
        cnt++;
        temp=temp->next;
        head=temp;
    }
    else if(temp->data==key){
        cnt++;
        prev->next=temp->next;
        temp=temp->next;

    } else{
        prev=temp;
        temp=temp->next;}
}
cout<<"counter of "<<key<<" = "<<cnt<<endl;
return head;
}
node* convertvectoll(vector <int> & arr){
    node* head= new node(arr[0]);
    node *mover=head;
    for (int i=1; i<arr.size();i++){
        node* temp= new node(arr[i]);
        mover->next=temp;
        mover=mover->next; //mover=temp;
    }
    return head;
}

```



```

void traversell(node* head){
    node* temp=head;
    while(temp!=nullptr){
        cout<<temp->data<<"->";
        temp=temp->next;
    }
    cout<<endl;
}
int main()
{
    vector<int> arr={1,2,1,2,1,3,1};
    node* head=convertvecttoll(arr);
    traversell(head);
    head=counterremover(head,1);
    traversell(head);
    return 0;
}

```

Q3

```

#include <iostream>
#include <bits/stdc++.h>
using namespace std;
class node{
public:
    int data;
    node *next;
public:
    node(int data1,node* next1){

```

```

        data=data1;
        next=next1;
    }
    node(int data){
        this->data=data;
        next=nullptr;
    }
};

int middle(node* head){
    if(head == NULL) return -1;

    // first pass: count nodes
    node* temp = head;
    int cnt = 0;
    while(temp){
        cnt++;
        temp = temp->next;
    }

    int centre = cnt / 2;

    // second pass: reach the middle node
    temp = head;
    cnt = 0;
    while(temp){
        if(cnt == centre) return temp->data;
        cnt++;
        temp = temp->next;
    }
    return -1; // fallback, should not hit
}

node* convertvectoll(vector <int> & arr){

```

```

node* head= new node(arr[0]);
node *mover=head;
for (int i=1; i<arr.size();i++){
    node* temp= new node(arr[i]);
    mover->next=temp;
    mover=mover->next; //mover=temp;
}
return head;
}
int main()
{
    vector<int> arr={10,2,7,3,4};
    node* head=convertvectoll(arr);
    int centre=middle(head);
    cout<<centre;
    return 0;
}

```

Q4

```

#include <iostream>
#include <bits/stdc++.h>
using namespace std;
class node{
public:
    int data;
    node *next;
public:
    node(int data1,node* next1){

```

```
        data=data1;
        next=next1;
    }
    node(int data){
        this->data=data;
        next=nullptr;
    }
};
node * reverser(node *head){
    node *temp=head;
    node* prev=NULL;
    node* fore=NULL;
    while(temp){
        fore=temp->next;
        temp->next=prev;
        prev=temp;
        temp=fore;
    }
    return prev;
}

int main()
{

    return 0;
}
```