

15B17CI371 – Data Structures

Lab ODD 2024

Week 1-LAB B

Practice Lab

1.

a.

```
#include<iostream>
```

```
using namespace std;
```

```
struct node
```

```
{
```

```
    int data;
```

```
    struct node *next;
```

```
};
```

```
struct node* insert(struct node *head,int data)
```

```
{ struct node *ptr=new struct node;
```

```
    if(head==NULL)
```

```
    {
```

```
        ptr->data=data;
```

```
        ptr->next=NULL;
```

```
        head=ptr;
```

```
        return head;
```

```
    }
```

```

else{
    ptr->data=data;
    ptr->next=head;
    head=ptr;
    return head;
}
}

void print(struct node *ptr)
{
    while(ptr!=NULL)
    {
        cout<<ptr->data<<" ";
        ptr=ptr->next;
    }

}

int main()
{
    struct node *head;
    head=NULL;

    int n;
    cout<<"Enter elements you want to insert in the beginning:";
    cin>>n;

```

```

int s;

while(n!=0)

{ cin>>s;

    head=insert(head,s);

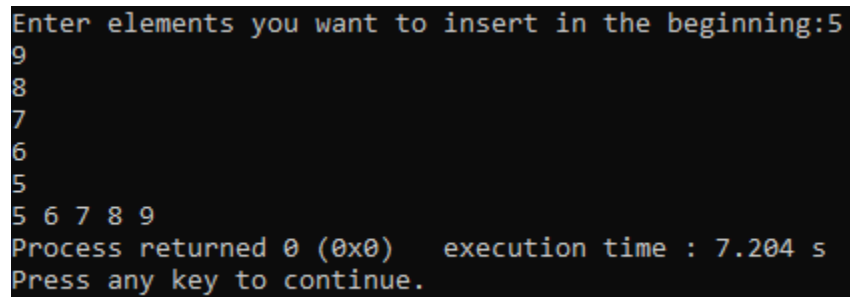
    n--;

}

print(head);

}

```



```

Enter elements you want to insert in the beginning:5
9
8
7
6
5
5 6 7 8 9
Process returned 0 (0x0)   execution time : 7.204 s
Press any key to continue.

```

b.

```

#include<iostream>

using namespace std;

struct node

{

    int data;

    struct node *next;

};

struct node* insert(struct node *head,int data)

{ struct node *ptr=new struct node;

```

```
if(head==NULL)
{
    ptr->data=data;
    ptr->next=NULL;
    head=ptr;
    return head;

}
else{
    ptr->data=data;
    ptr->next=head;
    head=ptr;
    return head;
}
}

void print(struct node *ptr)
{
    while(ptr!=NULL)
    {
        cout<<ptr->data<<" ";
        ptr=ptr->next;
    }

}
```

```

struct node* insertatnode(struct node *head,int k)
{
    struct node *ptr=head;

    struct node *p=new struct node;

    int e;

    cout<<"Enter the value you want to insert";

    cin>>e;

    while(k-2!=0){

        ptr=ptr->next;

        k--;

    }

    p->data=e;

    p->next=ptr->next;

    ptr->next=p;

    return head;

}

int main()
{

    struct node *head;

    head=NULL;

    int n;

    cout<<"Enter elements you want to insert in the beginning:";

    cin>>n;

```

```

int s;

while(n!=0)

{  cin>>s;

    head=insert(head,s);

    n--;

}

print(head);

cout<<endl;

int k;

cout<<"Enter the specified position where you want to insert the element"<<endl;

cin>>k;

head=insertatnode(head,k);

print(head);

}

```

```

Enter elements you want to insert in the beginning:5
8
7
6
5
4
4 5 6 7 8
Enter the specified position where you want to insert the element
2
Enter the value you want to insert7
4 7 5 6 7 8
Process returned 0 (0x0)   execution time : 9.550 s
Press any key to continue.

```

c.

```
#include<iostream>
```

```
using namespace std;

struct node
{
    int data;
    struct node *next;

};

struct node* insertdigit(struct node *head,int num)
{
    int c=0,r;
    while(num>0)
    { r=num%10;
      if(head==NULL)
      { struct node *ptr=new struct node;
        ptr->data=r;
        ptr->next=NULL;
        head=ptr;
      }
      else{
        struct node *ptr=new struct node;
        ptr->data=r;
        ptr->next=head;
        head=ptr;
      }
    }
}
```

```

    }

    num=num/10;

    c++;

    }

return head;

}

void print(struct node *ptr)
{
    while(ptr!=NULL)
    {
        cout<<ptr->data<<" ";

        ptr=ptr->next;
    }

}


int main()
{
    struct node *head;

    head=NULL;

    int n;

    cout<<"Enter a number:";

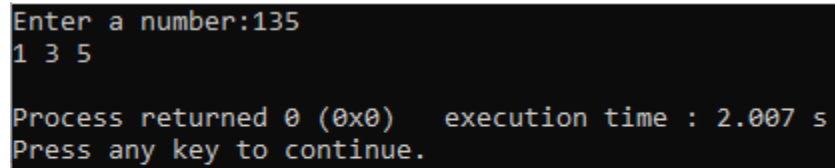
    cin>>n;

    head=insertdigit(head,n);

```



```
print(head);  
  
cout<<endl;  
  
}
```

A terminal window with a black background and white text. The first line shows the prompt 'Enter a number:' followed by the input '135'. The second line shows the output '1 3 5'. The third line shows the message 'Process returned 0 (0x0) execution time : 2.007 s'. The fourth line shows the prompt 'Press any key to continue.'.

```
Enter a number:135  
1 3 5  
  
Process returned 0 (0x0) execution time : 2.007 s  
Press any key to continue.
```

2.

```
#include<iostream>  
using namespace std;  
struct node  
{  
    string data;  
    struct node * next;  
};  
struct node* insert(struct node *head, string s)  
{ int i=(s.length()-1;  
    while(i>=0){  
  
        if (head==NULL)  
        {  
            struct node *ptr=new struct node;  
            ptr->data=s[i];  
            ptr->next=NULL;  
            head=ptr;  
        }  
        else  
        {
```

```

    struct node *ptr=new struct node;
    ptr->data=s[i];
    ptr->next=head;
    head=ptr;
}
i--;
}
return head;
};
void print(struct node *ptr)
{
    while(ptr!=NULL)
    {
        cout<<ptr->data<<" ";
        ptr=ptr->next;
    }
}
struct node *deletenode(struct node* head,struct node *node)
{
    if(head==node)
    {
        struct node *k=head;
        head=head->next;
        delete k;
        return head;
    }
    struct node*p=head;
    struct node*q=head->next;
    while(q!=node)
    {
        p=p->next;
        q=q->next;
    }
    p->next=q->next;
    delete q;
    return head;
}
struct node* removevowels(struct node *head)
{
    struct node *ptr=head;
    while(ptr!=NULL)
    {
        if(ptr->data=="a" || ptr->data=="e" || ptr->data=="i" || ptr->data=="o" || ptr->data=="u")
        {
            struct node *k=ptr->next;
            head=deletenode(head,ptr);
        }
    }
}

```

```

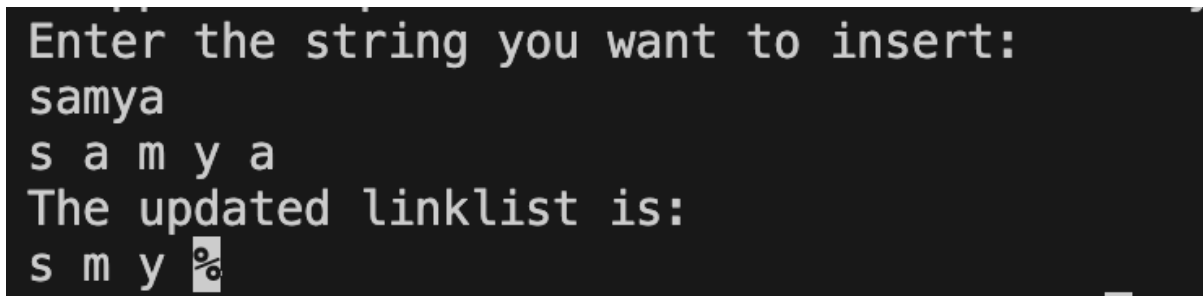
        ptr=k;
    }
    else{
        ptr=ptr->next;
    }

}

return head;
}

int main()
{
    struct node * head;
    head=NULL;
    string n;
    cout<<"Enter the string you want to insert:";
    cin>>n;
    head=insert(head,n);
    print(head);
    cout<<endl;
    head=removevowels(head);
    cout<<"The updated linklist is:"<<endl;
    print(head);
}

```



```

Enter the string you want to insert:
samy a
s a m y a
The updated linklist is:
s m y %

```

3.

```

#include<iostream>
#include<cstring>

```

```

using namespace std;
struct node{
    char data;
    struct node* next;
};
void print(struct node*head){
    struct node*ptr=head;
    cout<<endl;
    while(ptr!=NULL){
        cout<<ptr->data<<" ";
        ptr=ptr->next;
    }
}
struct node* insertatend(struct node*head, char data)
{
    struct node*p=new struct node;
    struct node*ptr=head;
    p->data=data;
    if(ptr==NULL)
    {
        p->next=NULL;
        head=p;
        return head;
    }
    while(ptr->next!=NULL){
        ptr=ptr->next;
    }
    ptr->next=p;
    p->next=NULL;
    return head;
}
bool checksublist(struct node*h1, struct node*h2,int *index)
{

```

```

    struct node*p=h1;
    struct node*q=h2;
    for(int b=0;b<3;b++){
        int count=0;
        struct node*r=q;
        for(int c=0;c<8;c++){
            struct node*s=p;
            pin:
            if(s->data==q->data){
                count++;
                if(count==3)

```

```

{
    *index=c+1;

    return true;
}
q=q->next;
s=s->next;

goto pin;
}

else{

    count=0;
    p=p->next;
    q=r;
    continue;
}

}
p=h1;
q=r->next;
}
return false;
}

struct node* deleteatindex(struct node*head, int index){
    struct node*p=head;
    struct node*q=head->next;
    while((index-2)!=0){
        p=p->next;
        q=q->next;
        index--;
    }
    p->next=q->next;
    delete q;
    return head;
}

struct node* deletesublist(struct node*h1, int a){
    struct node*p=h1;
    if (a==1)
    {
        struct node*p=h1;
        struct node*q=h1->next;
        struct node*r=q->next;;
        struct node*s=r->next;
        h1=s;
    }
}

```

```

        return h1;
        delete p,q,r;
    }
    h1=deleteatindex(h1,a);
    h1=deleteatindex(h1,a+1);
    h1=deleteatindex(h1,a+2);
    return h1;
}
int main(){
    struct node*h1=NULL;
    struct node*h2= NULL;
    char a;
    cout<<"Enter 10 characters\n";
    for(int i=0;i<10;i++){
        cin>>a;
        h1=insertatend(h1,a);
    }
    print(h1);
    cout<<"\nEnter 5 characters\n";
    for(int i=0;i<5;i++){
        cin>>a;
        h2= insertatend(h2,a);
    }
    print(h2);
    cout<<endl;
    int index ;
    if(checksublist(h1,h2, &index )){

        cout<<endl<<"position of the first common letters in the LL 1 : "<<index<<endl;
    }
    else{
        cout<<"No 3 consecutive characters of 2nd LL appears in the 1st LL\n ";
        return 0;
    }
    h1=deletesublist(h1, index);
    cout<<endl<<"Updated LL :\n";
    print(h1);
}

```

```

Enter 10 characters
q
w
e
r
t
y
u
i
o
p

q w e r t y u i o p
Enter 5 characters
q
w
e
r
t

q w e r t

position of the first common letters in the LL 1 : 1

Updated LL :

r t y u i o p %

```

4.

```

#include<iostream>
using namespace std;
struct node
{
int data;
struct node *next;
struct node *prev;
};
void print(struct node *head)
{
struct node * ptr=head;
while(ptr!=NULL)
{
cout<<ptr->data;
ptr=ptr->next;
}
}

```

```

}
struct node * insertatbeginning(struct node* head, int data)
{
if(head==NULL)
{
    struct node *ptr=new struct node;
    ptr->data=data;
    ptr->next=NULL;
    ptr->prev=NULL;
    head=ptr;
    return ptr;
}
else
{
    struct node *ptr=new struct node;
    ptr->data=data;
    ptr->next=head;
    ptr->prev=NULL;
    head=ptr;
    return ptr;
}
}
struct node * insertatlocation(struct node* head, int data,int k)
{
    struct node*ptr=head;
    struct node*p=new struct node;

    while((k-2)!=0)
    {
        ptr=ptr->next;
        k--;
    }
    p->data=data;
    p->next=ptr->next;
    ptr->next=p;
    p->prev=ptr;
    p->next->prev=p;

    return head ;}
int main()
{
    struct node *head;
    head=NULL;
    int n;
    cout<<"enter no of elements";
    cin>>n;

```



```

for(int i=0;i<n;i++)
{
    int s;
    cin>>s;
    head=insertatbeginning(head,s);
}
cout<<endl;
print(head);
cout<<endl;
int no,pos;
cout<<"enter element and pos to insert";
cin>>no>>pos;
head=insertatlocation(head,no,pos);
print(head);
}

```

```

enter no of elements4
1
2
3
4

4321
enter element and pos to insert 7 4
43271%

```

5.

```

#include<iostream>
using namespace std;
struct node
{
    int data;
    struct node *next;
    struct node *prev;
}

```

```

};
void print(struct node *head)
{
    struct node * ptr=head;
    while(ptr!=NULL)
    {
        cout<<ptr->data;
        ptr=ptr->next;
    }
}
struct node * insertatbeginning(struct node* head, int data)
{
    if(head==NULL)
    {
        struct node *ptr=new struct node;
        ptr->data=data;
        ptr->next=NULL;
        ptr->prev=NULL;
        head=ptr;
        return ptr;
    }
    else
    {
        struct node *ptr=new struct node;
        ptr->data=data;
        ptr->next=head;
        head->prev=ptr;
        ptr->prev=NULL;
        head=ptr;
        return ptr;
    }
}
struct node * insertatlocation(struct node* head, int data,int k)
{
    struct node*ptr=head;
    struct node*p=new struct node;

    while((k-2)!=0)
    {
        ptr=ptr->next;
        k--;
    }
    p->data=data;
    p->next=ptr->next;
    ptr->next=p;
    p->prev=ptr;
}

```

```

    p->next->prev=p;

    return head ;}
struct node* deletelast(struct node* head)
{
    struct node* ptr=head;
    while(ptr->next!=NULL)
    {
        ptr=ptr->next;
    }
    ptr->prev->next=NULL;

    delete ptr;
    return head;
}
int main()
{
    struct node *head;
    head=NULL;
    int n;
    cout<<"enter no of elements";
    cin>>n;
    for(int i=0;i<n;i++)
    {
        int s;
        cin>>s;
        head=insertatbeginning(head,s);
    }
    cout<<endl;
    print(head);
    cout<<endl;
    cout<<endl<<"updated Linked list";
    head=deletelast(head);
    print(head);
}

```

enter no of elements5

9

8

7

6

5

56789

updated Linked list 56789%

6.

```
#include<iostream>
using namespace std;
struct node
{
    int data;
    struct node *next;
    struct node *prev;
};
void print(struct node *head)
{
    struct node * ptr=head;
    while(ptr!=NULL)
    {
        cout<<ptr->data;
        ptr=ptr->next;
    }
}
struct node * insertatbeginning(struct node* head, int data)
{
    if(head==NULL)
    {
        struct node *ptr=new struct node;
        ptr->data=data;
        ptr->next=NULL;
        ptr->prev=NULL;
```

```

    head=ptr;
    return ptr;
}
else
{
    struct node *ptr=new struct node;
    ptr->data=data;
    ptr->next=head;
    head->prev=ptr;
    ptr->prev=NULL;
    head=ptr;
    return ptr;
}
}
struct node* swap(struct node*head,int n)
{
    int i=0;
    struct node *ptr=head;
    struct node *ptr1=head->next;
    while(ptr1->next!=NULL)
    {
        ptr1=ptr1->next;
    }
    int j=n/2;
    int s=1;
    while(j!=0)
    {
        int temp=0;
        temp=ptr->data;
        ptr->data=ptr1->data;
        ptr1->data=temp;
        ptr=ptr->next;
        ptr1=ptr1->prev;
        j--;
        cout<<"After "<<s<<" call: ";
        print(head);
        cout<<endl;
        s++;
    }
    return head;
}
int main()
{
    struct node *head;
    head=NULL;
    int n;
    cout<<"enter no of elements";

```

```

cin>>n;
for(int i=0;i<n;i++)
{
    int s;
    cin>>s;
    head=insertatbeginning(head,s);
}
cout<<endl;
print(head);
cout<<endl;
head=swap(head,n);
print(head);
}

```

```

enter no of elements5
1
2
3
4
5

54321
After 1 call: 14325
After 2 call: 12345
12345%

```

7.

```

#include<iostream>

#include<cstring>

using namespace std;

struct node{

    int data;

    int degree;

```

```

struct node* next;

};

void print(struct node*head){
    struct node*ptr=head;
    cout<<endl;
    while(ptr->next!=NULL){
        cout<<ptr->data<<"x^"<<ptr->degree<<" ";
        ptr=ptr->next;
    }
    cout<<ptr->data<<"x^"<<ptr->degree;
    cout<<endl;
}

struct node* insertatend(struct node*head, int data, int degree)
{
    struct node*p=new struct node;
    struct node*ptr=head;

    p->data=data;
    p->degree=degree;
    if(ptr==NULL)
    {
        p->next=NULL;
        head=p;
        return head;
    }
    while(ptr->next!=NULL){

```

```

    ptr=ptr->next;
}
ptr->next=p;
p->next=NULL;
return head;
}
struct node* addpol(struct node*h1, struct node*h2)
{
    int degree,data;
    struct node*sum=new struct node;
    sum=NULL;
    struct node*p=h1;
    struct node*q=h2;
    while(p!=NULL)
    {
        degree=p->degree;
        data=p->data + q->data;
        sum=insertatend(sum,data,degree) ;
        p=p->next;
        q=q->next;
    }
    return sum;
}

```



```

int main(){
    struct node*h1=NULL;
    struct node*h2= NULL;
    int a,k,s;
    cout<<"Enter degree of polynomial:\n";
    cin>>s;
    for(int i=s;i>=0;i--){
        cout<<"enter coefficient of "<<i<<" degree term in Pol 1 :\n";
        cin>>a;
        h1=insertatend(h1,a,i);
    }
    for(int i=s;i>=0;i--){
        cout<<"enter coefficient of "<<i<<" degree term in Pol 2 :\n";
        cin>>a;
        h2=insertatend(h2,a,i);
    }
    print(h1);
    print (h2);
    cout<<"Sum of Polynomials:\n";
    struct node*sum=addpol(h1,h2);
    print(sum);
}

```

```

Enter degree of polynomial:
2
enter coefficient of 2 degree term in Pol 1 :
7
enter coefficient of 1 degree term in Pol 1 :
7
enter coefficient of 0 degree term in Pol 1 :
7
enter coefficient of 2 degree term in Pol 2 :
7
enter coefficient of 1 degree term in Pol 2 :
7
enter coefficient of 0 degree term in Pol 2 :
7

7x^2+7x^1+7x^0

7x^2+7x^1+7x^0
Sum of Polynomials:

14x^2+14x^1+14x^0

```

8.

```

#include<iostream>

#include<cstring>

using namespace std;

struct node{

    int data;

    int degree;

    struct node* next;

};

void print(struct node*head){

    struct node*ptr=head;

    cout<<endl;

```

```

while(ptr->next!=NULL){
// cout<<"Degree: "<<ptr->degree<<" Coefficient: "<<ptr->data<<endl;
// ptr=ptr->next;
cout<<ptr->data<<"x^"<<ptr->degree<<"+";
ptr=ptr->next;
}
cout<<ptr->data<<"x^"<<ptr->degree;
cout<<endl;
}
struct node* insertatend(struct node*head, int data, int degree)
{
    struct node*p=new struct node;
    struct node*ptr=head;
    p->data=data;
    p->degree=degree;
    if(ptr==NULL)
    {
        p->next=NULL;
        head=p;
        return head;
    }
    while(ptr->next!=NULL){
        ptr=ptr->next;
    }
    ptr->next=p;
}

```

```

    p->next=NULL;
    return head;
}
struct node* prodpol(struct node*h1, struct node*h2)
{
    int degree,data;
    struct node*product=new struct node;
    product=NULL;
    struct node*p=h1;
    struct node*q=h2;
    while(p!=NULL)
    {
        while(q!=NULL)
        {
            degree=p->degree+q->degree;
            data=(p->data)*(q->data);
            product=insertatend(product,data,degree) ;
            q=q->next;
        }
        q=h2;
        p=p->next;
    }
    return product;
}

```

```

int main(){
    struct node*h1=NULL;
    struct node*h2= NULL;
    int a,k,s;
    cout<<"Enter degree of polynomial:\n";
    cin>>s;
    for(int i=s;i>=0;i--){
        cout<<"enter coefficient of "<<i<<" degree term in Pol 1 :\n";
        cin>>a;
        h1=insertatend(h1,a,i);
    }
    for(int i=s;i>=0;i--){
        cout<<"enter coefficient of "<<i<<" degree term in Pol 2 :\n";
        cin>>a;
        h2=insertatend(h2,a,i);
    }
    print(h1);
    print (h2);
    cout<<"Product of Polynomials:\n";
    struct node*product=prodpol(h1,h2);
    print(product);
}

```

```
Enter degree of polynomial:
2
enter coefficient of 2 degree term in Pol 1 :
7
enter coefficient of 1 degree term in Pol 1 :
7
enter coefficient of 0 degree term in Pol 1 :
7
enter coefficient of 2 degree term in Pol 2 :
1
enter coefficient of 1 degree term in Pol 2 :
1
enter coefficient of 0 degree term in Pol 2 :
1

7x^2+7x^1+7x^0

1x^2+1x^1+1x^0
Product of Polynomials:

7x^4+7x^3+7x^2+7x^3+7x^2+7x^1+7x^2+7x^1+7x^0
```