15B17CI371 – Data Structures

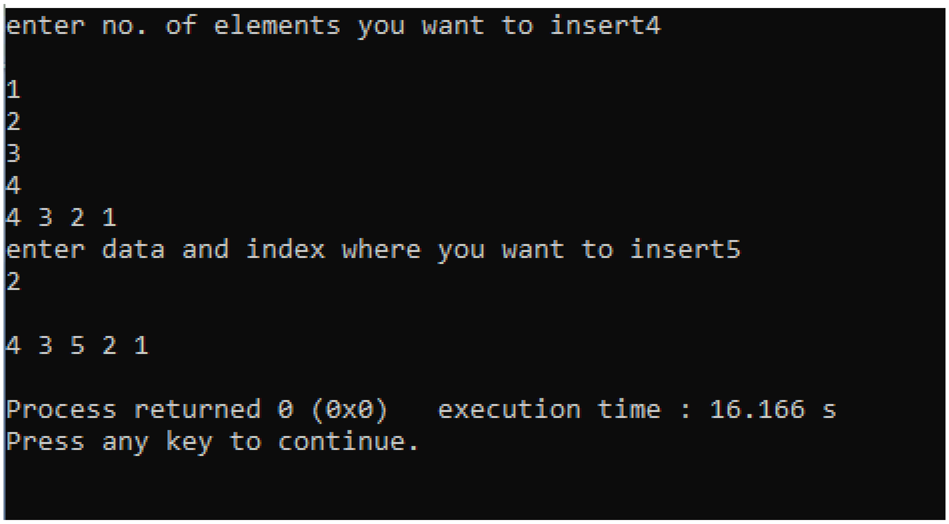
Lab ODD 2024

Week 1-LAB B

Practice Lab

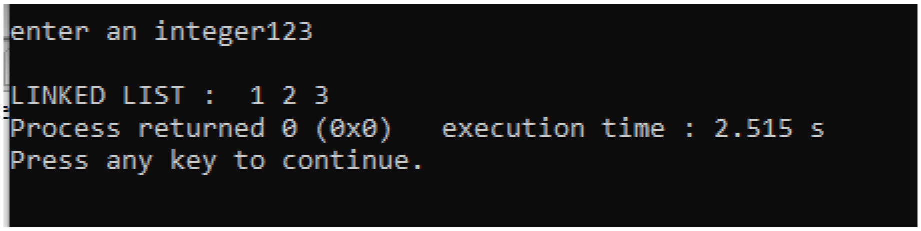
1.

```
#include <iostream>

using namespace std;

struct node

{

   int data;

   struct node *next;

};

void traversal(struct node* head)

{

   struct node *p=head;


   while(p!=NULL)

   {

      cout<<p->data<<" ";

      p=p->next;

   }

}

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

{
    struct node *p=new struct node;
    if(head==NULL)
    {
        p->data=n;
        p->next=NULL;
        head=p;
        return head;
    }
    else{
        p->data=n;
        p->next=head;
        head=p;
        return head;

    }
}
struct node* insertatnode(struct node *head,int data,int pos)
{
    struct node *p=new struct node;
    struct node*ptr=head;
    int i=0;
    while(i!=(pos-1))
    {
        ptr=ptr->next;
        i++;
    }
    p->data=data;
    p->next=ptr->next;
    ptr->next=p;

    return head;

};
int main()
{
    struct node *head=NULL;
    int n;
    cout<<"enter no. of elements you want to insert";
    cin>>n;
    cout<<endl;
    for (int i=0;i<n;i++)
    {
        int data;
        cin>>data;
        head=insert(head,data);
    }
    traversal(head);
    cout<<endl;
    int a,b;
    cout <<"enter data and index where you want to insert";
    cin>>a>>b;

    cout<<endl;

    head=insertatnode(head,a,b);

    traversal(head);

    cout<<endl;

    cout<<"enter an integer";

    int no;

    cin>>no;

    int y=no;

    cout<<endl;

    int count=0;

    while ((no%10)>0)

    {


        count++;

        no=no/10;

    }

    for (int i=0;i<count;i++)

    {

        head=insert(head,(y%10));

        y=y/10;

    }

    traversal(head);


}
```



```
enter no. of elements you want to insert4

1
2
3
4
4 3 2 1
enter data and index where you want to insert5
2

4 3 5 2 1

Process returned 0 (0x0)   execution time : 16.166 s
Press any key to continue.
```

c)



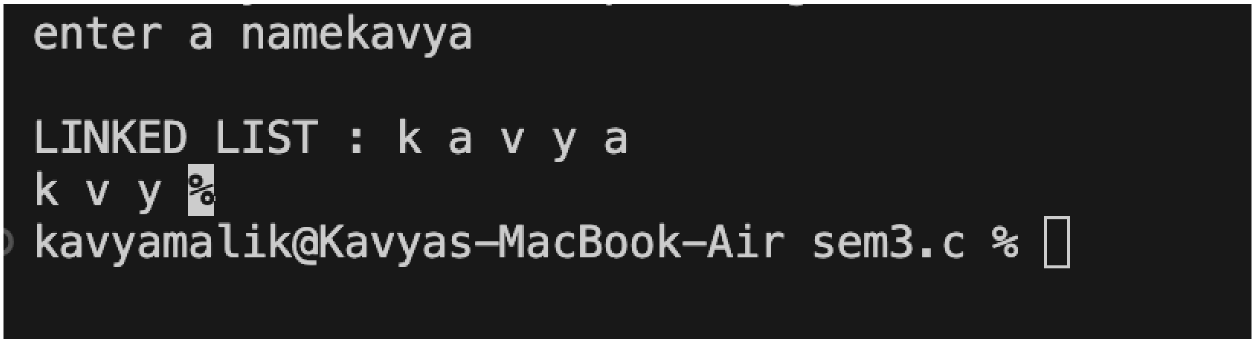
```
enter an integer123

LINKED LIST :  1 2 3
Process returned 0 (0x0)   execution time : 2.515 s
Press any key to continue.
```

2.

```
#include <iostream>
using namespace std;
struct node
{
   char data;
   struct node *next;
};
struct node* insert(struct node *head, int data)
{
   struct node *p=new struct node;
   if (head==NULL)
   {
     p->data=data;
     p->next=NULL;
     head=p;
```

```
       return head;
    }
    else
    {
        p->data=data;
        p->next=head;
        head=p;
        return head;
    }
};
void traversal(struct node* head)
{
    struct node *p=head;
    while(p!=NULL)
    {
        cout<<p->data<<" ";
        p=p->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')
        {
```

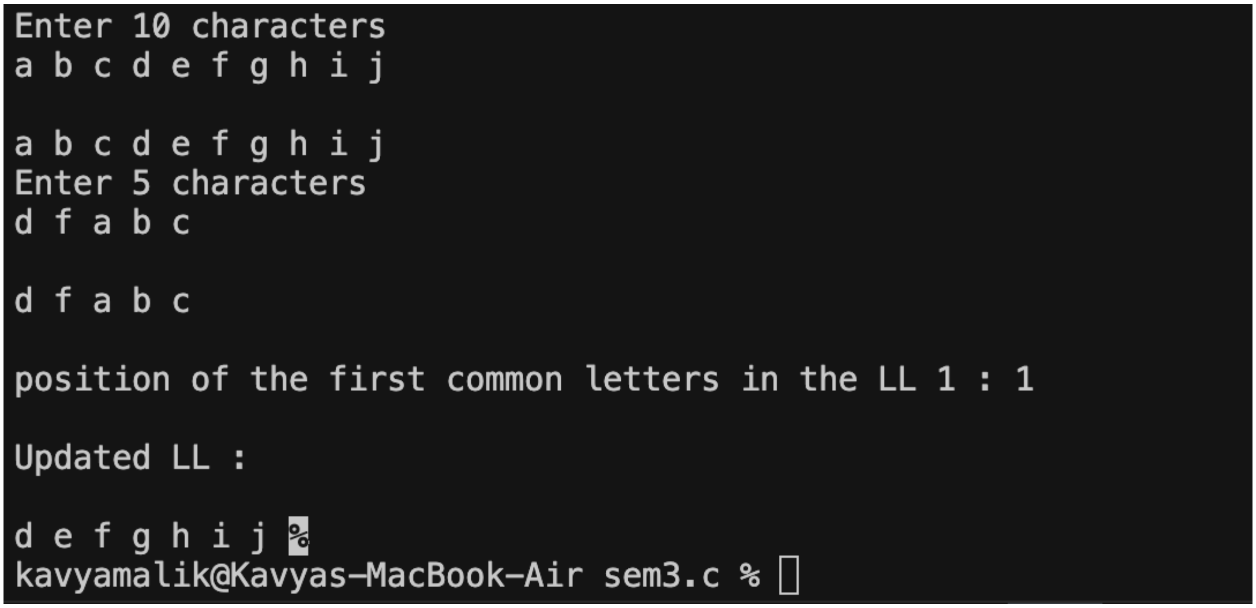
```
      struct node *k=ptr->next;
      head=deletenode(head,ptr);
      ptr=k;
      }
      else{
         ptr=ptr->next;
      }

   }
   return head;
}
int main()
{
  struct node *head=NULL;
  cout<<"enter a name";
  string s;
  cin>>s;
  cout<<endl;
  int n=s.length();
  for (int i=n-1;i>=0;i--)
  {
     head=insert(head,s[i]);
  }
  cout<<"LINKED LIST : ";
  traversal(head);
  cout<<endl;
  head=removevowels(head);
  traversal(head);
}
```



```
enter a namekavya

LINKED LIST : k a v y a
k v y %
kavyamalik@Kavyas-MacBook-Air sem3.c %
```

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);
}
```

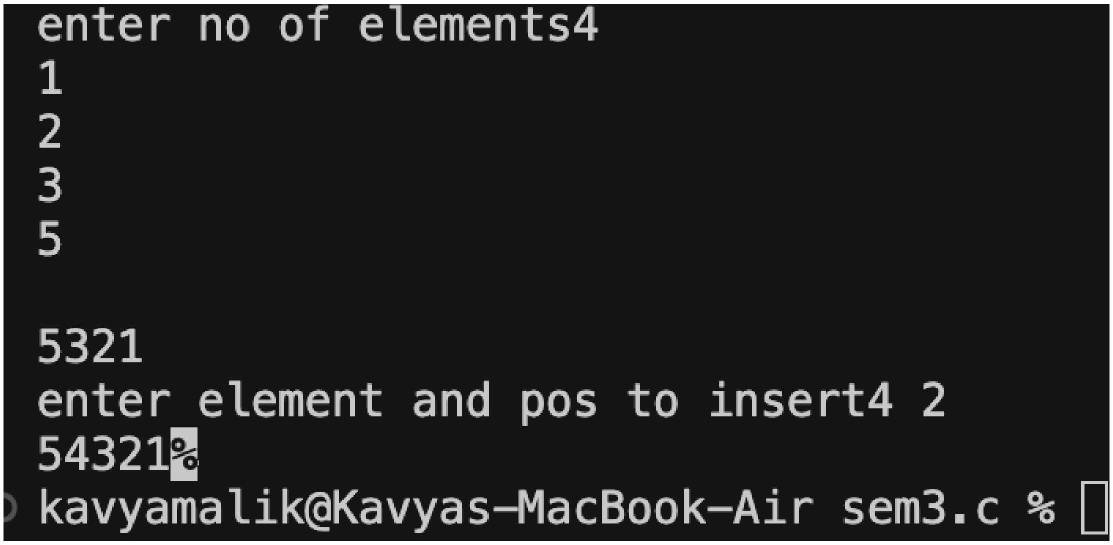


```
Enter 10 characters
a b c d e f g h i j

a b c d e f g h i j
Enter 5 characters
d f a b c

d f a b c

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

Updated LL :

d e f g h i j %
kavyamalik@Kavyas-MacBook-Air sem3.c % 
```

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);
}
```

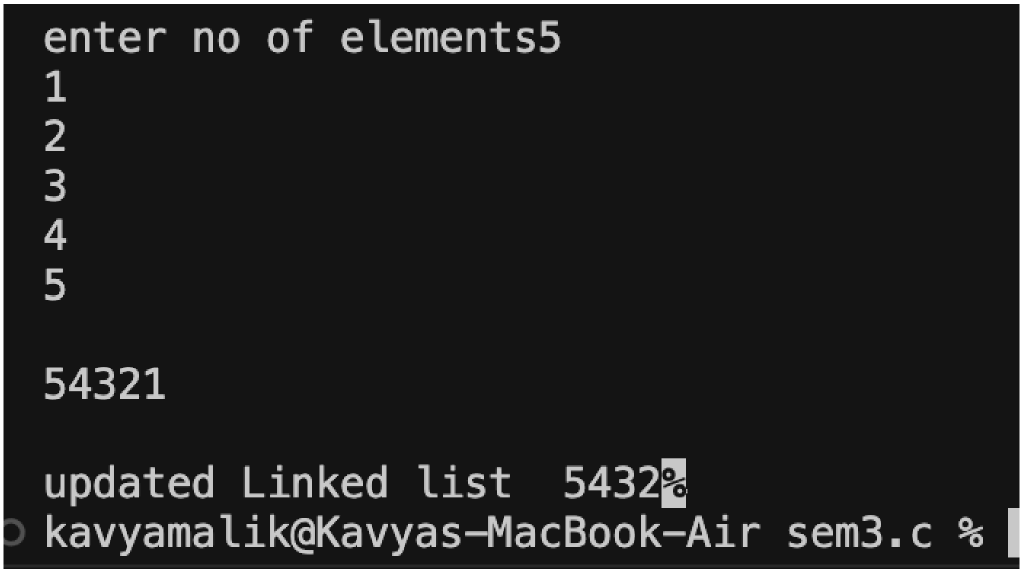


```
enter no of elements4
1
2
3
5

5321
enter element and pos to insert4 2
54321%
kavyamalik@Kavyas-MacBook-Air sem3.c %
```

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;
}
}
```

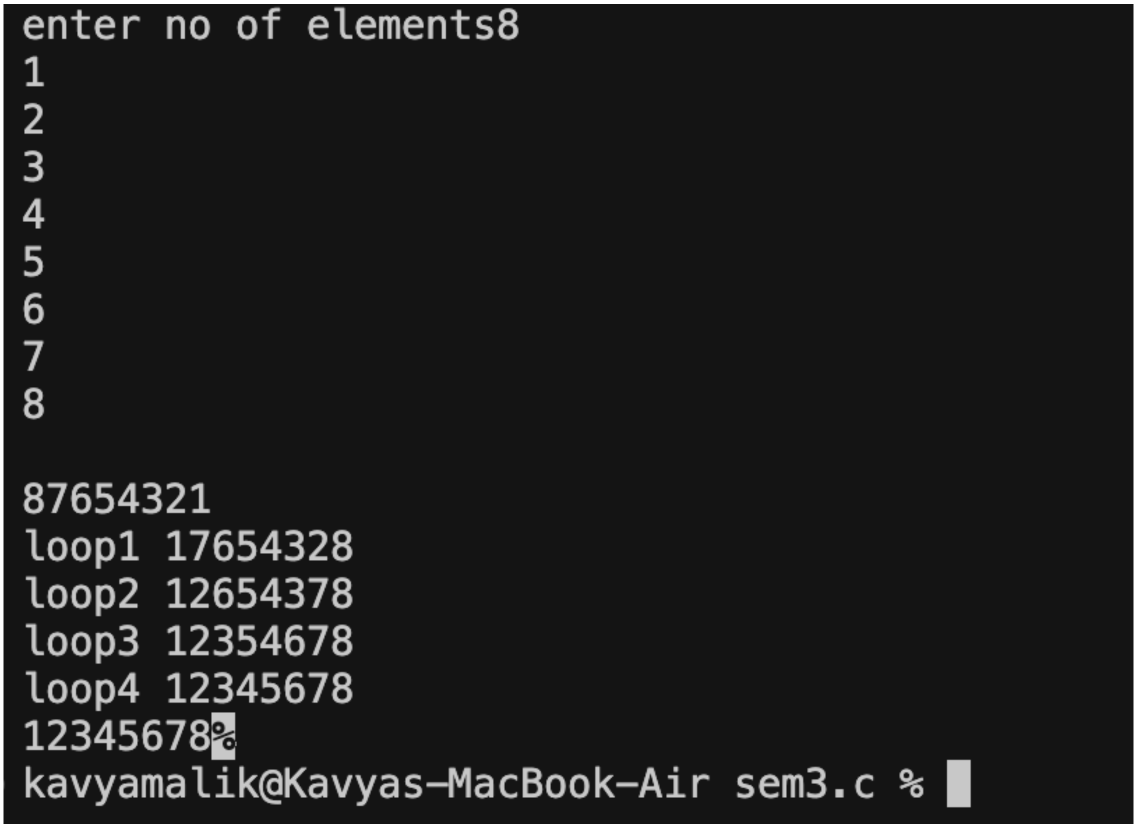
```
struct node * insertatbeginning(struct node* head, int data)
{
if(head==NULL)
{
    struct node *ptr=new struct node;
    ptr->data=data;
    ptr->next=NULL;
    ptr->prev=NULL;
    head=ptr;
    return ptr;
}
else
{
    struct node *ptr=new struct node;
    ptr->data=data;
    ptr->next=head;
    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
1
2
3
4
5

54321

updated Linked list  5432%
kavyamalik@Kavyas-MacBook-Air sem3.c %
```

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)
{
   struct node* p1=head;
   struct node *p2=head;
   while (p2->next!=NULL)
   {
      p2=p2->next;
   }
   for(int i=0;i<n/2;i++)
{
   int s=p1->data;
   p1->data=p2->data;
   p2->data=s;
   p1=p1->next;
   p2=p2->prev;
   cout<<"loop"<<i+1<<" ";
   print(head);
   cout<<endl;

}
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 elements8
1
2
3
4
5
6
7
8

87654321
loop1 17654328
loop2 12654378
loop3 12354678
loop4 12345678
12345678%
kavyamalik@Kavyas-MacBook-Air sem3.c %
```

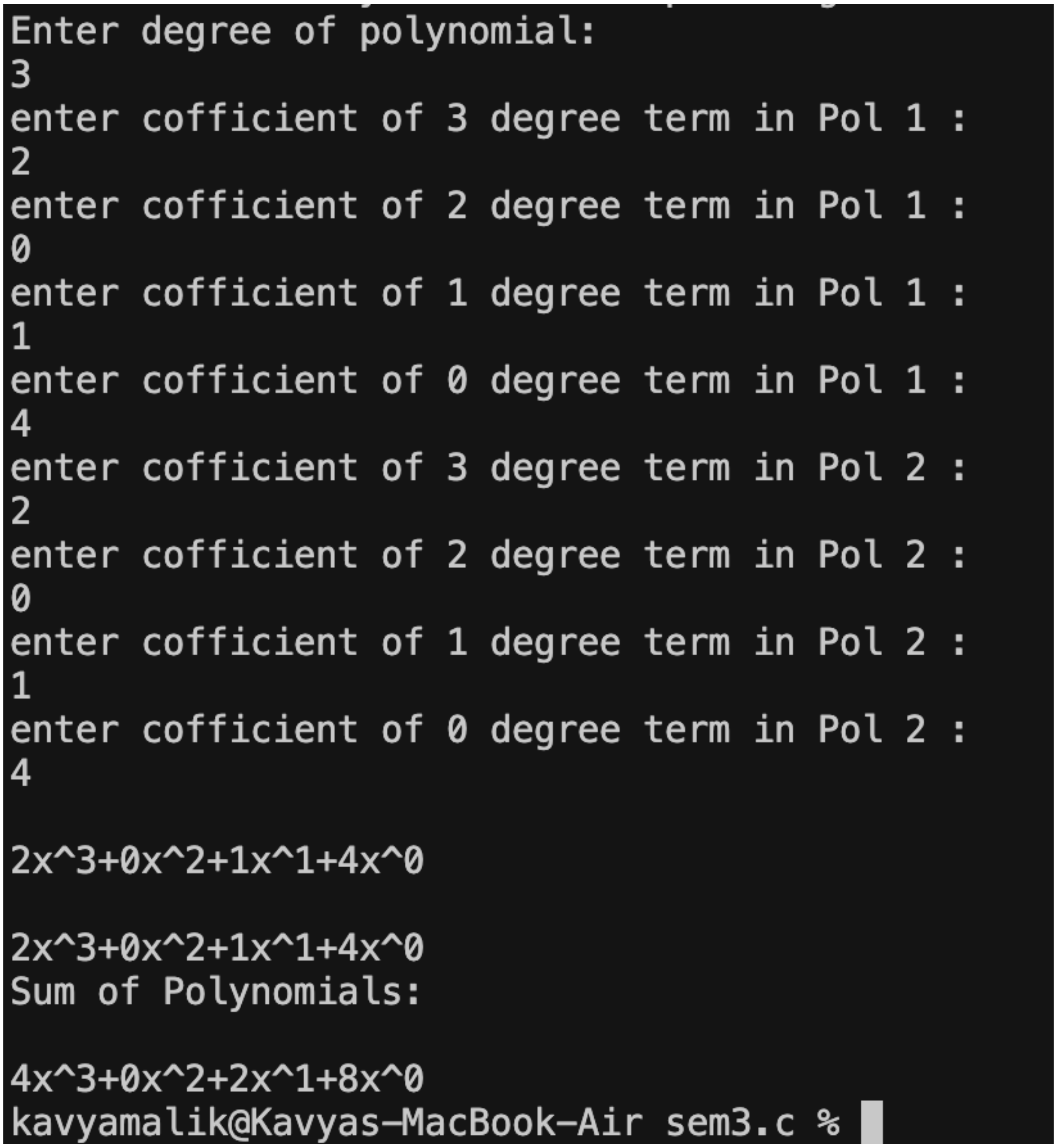
7.

```
#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";
```

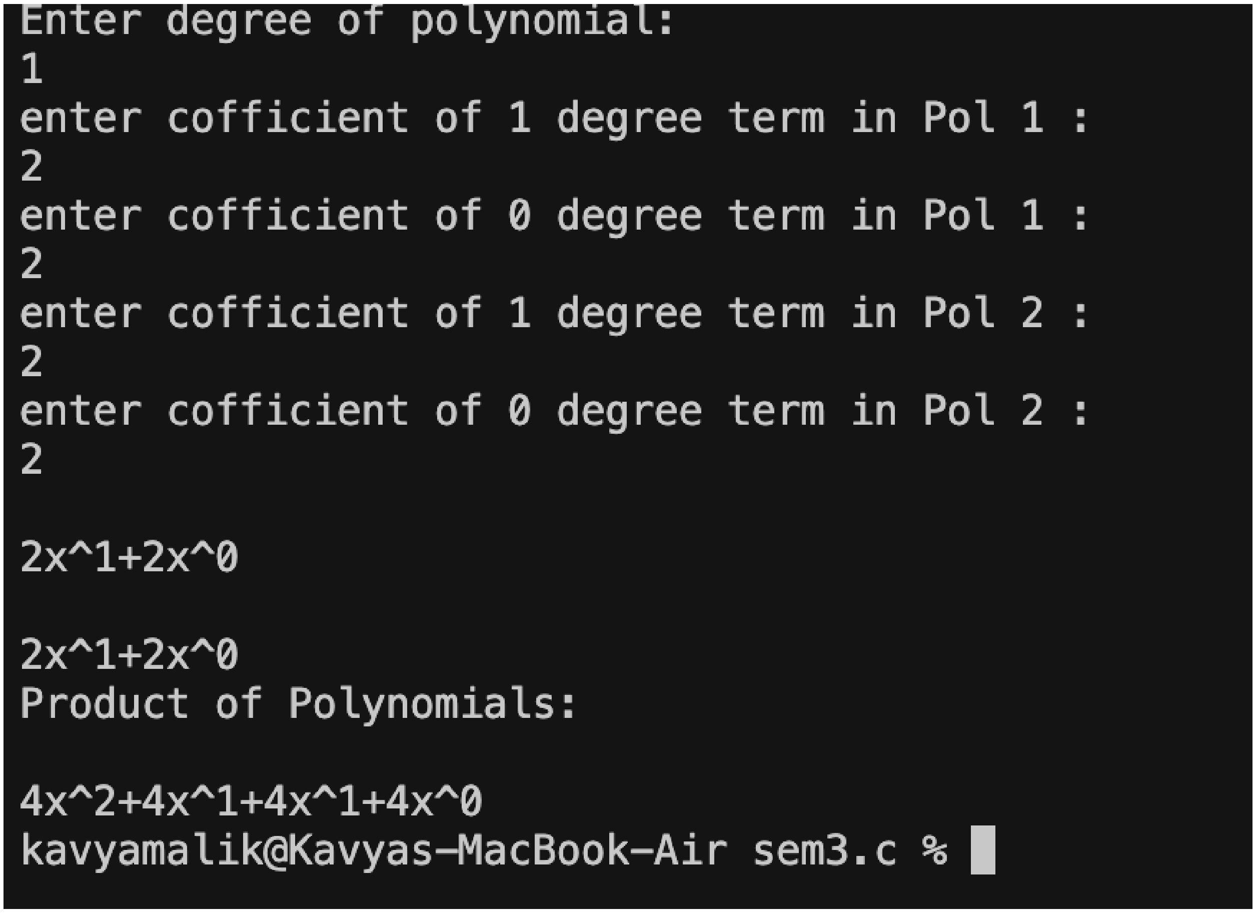
```
cin>>s;

for(int i=s;i>=0;i--){

    cout<<"enter cofficient of "<<i<<" degree term in Pol 1 :\n";

    cin>>a;

    h1=insertatend(h1,a,i);

}

for(int i=s;i>=0;i--){

    cout<<"enter cofficient 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:
3
enter cofficient of 3 degree term in Pol 1 :
2
enter cofficient of 2 degree term in Pol 1 :
0
enter cofficient of 1 degree term in Pol 1 :
1
enter cofficient of 0 degree term in Pol 1 :
4
enter cofficient of 3 degree term in Pol 2 :
2
enter cofficient of 2 degree term in Pol 2 :
0
enter cofficient of 1 degree term in Pol 2 :
1
enter cofficient of 0 degree term in Pol 2 :
4

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

2x^3+0x^2+1x^1+4x^0
Sum of Polynomials:

4x^3+0x^2+2x^1+8x^0
kavyamalik@Kavyas-MacBook-Air sem3.c % 
```

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 cofficient of "<<i<<" degree term in Pol 1 :\n";
    cin>>a;
    h1=insertatend(h1,a,i);
}
for(int i=s;i>=0;i--){
    cout<<"enter cofficient 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:
1
enter cofficient of 1 degree term in Pol 1 :
2
enter cofficient of 0 degree term in Pol 1 :
2
enter cofficient of 1 degree term in Pol 2 :
2
enter cofficient of 0 degree term in Pol 2 :
2

2x^1+2x^0

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

4x^2+4x^1+4x^1+4x^0
kavyamalik@Kavyas-MacBook-Air sem3.c %
```