

```
/* Program to swap two adjacent nodes in a single linked list */
```

```
#include <iostream>
```

```
#include <stdio.h>
```

```
using namespace std;
```

```
struct node
```

```
{
```

```
    int data;
```

```
    struct node *link;
```

```
};
```

```
struct node *root=NULL;
```

```
int loc;
```

```
void append()
```

```
{
```

```
    struct node *temp;
```

```
    temp=(struct node *)malloc(sizeof(struct node));
```

```
    cout<<"Enter node data"<<endl;
```

```
    cin>>temp->data;
```

```
    temp->link=NULL;
```

```
    if(root==NULL)
```

```
        root=temp;
```

```
    else
```

```
    {
```

```
        struct node *p;
```

```
        p=root;
```

```
        while(p->link!=NULL)
```

```
        {
```

```
            p=p->link;
```

```
        }
```

```
        p->link=temp;
```

```
    }
```

```
}
```

```
void display()
```

```
{
```

```
    struct node *temp;
```

```
    temp=root;
```

```
    if(temp==NULL)
```

```
        cout<<"List is empty"<<endl;
```

```
    else
```

```

{
    while(temp!=NULL)
    {
        cout<<temp->data<<"->";
        temp=temp->link;
    }
    cout<<"NULL"<<endl;
}
}

```

```

void swap()
{

```

```

    struct node *p,*r,*q=NULL;
    int i=1;
    p=root;

```

```

    while(i<loc-1)
    {
        p=p->link; //Pointer 1
        i++;
    }

```

```

    q=p->link; //Pointer 2

```

```

    r=q->link; //Pointer 3

```

```

    q->link=r->link; //Connection 1
    r->link=q; // Connection 2
    p->link=r; // Connection 3

```

```

}

```

```

int main()
{
    int n;
    cout<<"Enter number of nodes"<<endl;
    cin>>n;
    for(int i=1;i<=n;i++)
        append();
    cout<<"Displaying nodes before swapping"<<endl;

    display();

```

```
cout<<"Enter Location of node to be swapped"<<endl;
```

```
cin>>loc;
```

```
cout<<"Swapping nodes"<<endl;
```

```
swap();
```

```
cout<<"Displaying nodes after swapping"<<endl;
```

```
display();
```

```
exit(0);
```

```
}
```