Circular Linked List

```
#include<iostream>
using namespace std;
typedef struct list
{
   int info;
    struct list *link;
}node;
node *start=NULL,*ptr,*loc;
void ins_beg(int);
void del_beg();
void traverse();
int main()
{
    int x, item;
    char ch;
    do
    {
        cout<<" Enter 1 for insertion, 2 for deletion, 3 for traverse"<<endl;
        cin>>x;
        switch(x)
        {
            case 1: cout<<" Enter the element to add"<<endl;
                    cin>>item;
                    ins_beg(item);
                    break;
            case 2: del_beg();
                    break;
            case 3: traverse();
                    break;
        }
        cout<<" Press Y to re-enter"<<endl;
        cin>>ch;
    }while(ch=='y');
return 0;
}
void ins_beg(int item)
{
    ptr=new node;
    ptr->info=item;
    if(start==NULL)
    {
        start =ptr;
        ptr->link=ptr;
    }
    else
    {
        loc=start;
        while(loc->link!=start)
        {
            loc=loc->link;
        }
        loc->link=ptr;
        ptr->link=start;
        start = ptr;
    }
}
void del_beg()
{
    if (start==NULL)
    {
        cout<<" Linked list is empty, del not possible"<<endl;
    }
    else if(start->link==start)
    {
        ptr=start;
        start=NULL;
        delete ptr;
        cout<<" Node deleted"<<endl;
    }
    else
    {
        loc=start;
        while(loc->link!=start)
        {
            loc=loc->link;
        }
        ptr=start;
        loc->link=ptr->link;
        start=ptr->link;
        delete ptr;
        cout<<" Node deleted"<<endl;
    }
}
void traverse()
{
    if(start==NULL)
    {
        cout<<" List is empty, so it cannot be traversed"<<endl;
    }
    else
    {
        ptr=start;
        while(ptr->link!=start)
        {
            cout<<"Element is "<<ptr->info<<endl;
            ptr=ptr->link;
        }
        cout<<"Element is "<<ptr->info<<endl;
    }
}
```