4.4 Circular Lists

4.4

Singly-linked Circular Lists

- Can visit a node from any position
- The link field of the last node points to the first node

- Check for the last node
  - if(current->link == ***first***)
- Easier to store the last node of a circular list and access the first node via last->link

Circular Lists : Insert

- Suppose we want to insert a new node at the front of list
- Set link field of new node to ***first*** and set ***first*** to new node
- Go to the last node and set the link field to new node

Circular Lists: Insert at Front

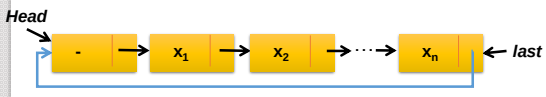
```
Template<class T>
void CircularList<T>::InsertFront(const T& e)
{
   ChainNode<T>* newNode = new ChainNode<T>(e);

   if(last){ // nonempty list
     newNode->link = last->link;
     last->link = newNode;
   }
   else{ // empty list
     last = newNode;
     newNode->link = newNode;
   }
}
```

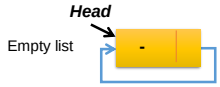
24

Circular Lists

- How to represent an “empty” list?

- Introducing a dummy node “Header”





25