Linked Lists

SLLists provide pseudo-generic singly linked lists. DLLists provide doubly linked lists. The lists are designed for the simple maintenance of elements in a linked structure, and do not provide the more extensive operations (or node-sharing) of class **List**. They behave similarly to the **slist** and similar classes described by Stroustrup.

All list nodes are created dynamically. Assignment is performed via copying.

Class **DLList** supports all **SLList** operations, plus additional operations described below.

For purposes of illustration, assume the specification of class **intSLList**. In addition to the operations listed here, SLLists support traversal via Pixes. See *Pseudo-indexes* in **/NextLibrary/Documentation/GNU/libg++/Intro.rtf**.

intSLList a;	Declares a to be an empty list.
intSLList b = a;	Sets b to an element-by-element copy of a.
a.empty()	returns true if a contains no elements
a.length();	returns the number of elements in a.
a.prepend(x);	places x at the front of the list.
a.append(x);	places x at the end of the list.
a.join(b)	places all nodes from b to the end of a, simultaneously destroying b.
x = a.front()	returns a reference to the item stored at the head of the list, or triggers

	an error if the list is empty.
a.rear()	returns a reference to the rear of the list, or triggers an error if the list is empty.
x = a.remove_front()	deletes and returns the item stored at the head of the list.
a.del_front()	deletes the first element, without returning it.
a.clear()	deletes all items from the list.
a.ins_after(Pix i, item);	inserts item after position i. If i is null, insertion is at the front.
a.del_after(Pix i);	deletes the element following i. If i is 0, the first item is deleted.

Doubly linked lists

Class **DLList** supports the following additional operations, as well as backward traversal via Pixes.

x = a.remove_rear();	deletes and returns the item stored at the rear of the list.
a.del_rear();	deletes the last element, without returning it.
a.ins_before(Pix i, x)	inserts x before the i.
a.del(Pix& iint dir = 1)	deletes the item at the current position, then advances forward if dir is positive, else backward.