Locatives in C

CSE 326: Data Structures

March 25, 1996

The section of the Lewis and Denenberg textbook on Locatives is somewhat unusual, but necessary to understand some of the algorithms that arise later in the book. In order to help you understand locatives, the program fragment below implements Algorithm 1.6 in C. For this to be helpful, you will have to look carefully at the differences between the textbook's algorithm and the function `LLInsert` below.

```
/************************************************************
 * Implementation of Lewis and Denenberg's Algorithm 1.6 in C,
 * using pointers to simulate locatives.
 ************************************************************/

typedef int key;

struct node
{
    key             Key;
    struct node     *Next;
};
typedef struct node *list;

list NewNode(key K, list L)
/* return a pointer to a new node with fields K and L */
{
    list new;
    new = malloc(sizeof(struct node));
    new->Key = K;
    new->Next = L;
    return new;
}

void LLInsert(key K, list *P)
/* Insert a cell containing key K in list *P if none exists already */
/* Note: P is a pointer to a pointer to a node; you should understand
 *      why one assignment is of the form P = ... and one *P = ... */
{
    while (*P != NULL && (*P)->Key < K)
        P = &((*P)->Next);
    if (*P != NULL && (*P)->Key == K) return;
    *P = NewNode(K, *P);
}
```