#include <iostream>

+#include <cstdlib>

+#include <conio.h>

+#define NIL NULL

+#define next(P) (P)next

+#define first(P) ((L).first)

+using namespace std;

+

+void deleteFirst (list \*L, address \*P)

+{

+ \*P = first(\*L);

+ first(\*L) = next(\*P);

+ next(\*P) = NIL;

+}

+

+void deleteLast (list \*L, address \*Q, address \*P)

+{

+ \*Q = first(\*L);

+ while (next(next(\*Q) != NIL))

+ \*Q = next(\*Q);

+ \*P = next(\*Q);

+ next(\*Q) = NIL;

+#include <iostream>

+#include <cstdlib>

+#include <conio.h>

+#define NIL NULL

+#define next(P) (P)next

+#define first(P) ((L).first)

+

+using namespace std;

+

+void deleteAfter (list \*L, address \*P, address \*Prec){

+ \*P = next(\*Prec);

+ next(\*Prec) = next(\*P);

+ next(\*P) = NIL;

+

+}

+

+address searchElement(list L, infotype x){

+ address Q;

+ Q = first(L);

+ while (info(Q) != x) {

+ Q = next(Q);

+ }

+

+ return Q;