//#include "hflist.h"

#include "stdio.h"

#define MAXLENGTH 100

typedef int ELEMT;

typedef unsigned POS;

typedef struct

{

ELEMT elements[MAXLENGTH+1];

POS last;

} FLIST;

// прототипи функцій

ELEMT retrieve(FLIST \*pl,POS k,int \*error);

void edit(FLIST \*pl,POS k,ELEMT p,int \*error);

void insert(FLIST \*pl,POS k,ELEMT p,int \*error);

ELEMT del(FLIST \*pl, POS k, int \*error);

POS locate (FLIST \*pl, ELEMT p);

int len(FLIST \*pl);

int main()

{

FLIST L;

L.last=0;

int i;

i=0;

int nerror;

// Додаєм елементи списку

insert(&L,1,100,&nerror);

insert(&L,2,200,&nerror);

insert(&L,3,300,&nerror);

insert(&L,4,400,&nerror);

// Видаєм список на консоль (екран)

printf(" After insert:\n");

for (i=1;i<=len(&L); i++)

printf("L[%d] =%d\n",i,retrieve(&L,i,&nerror));

// Перевірка коригування елементів списку

for (i=1;i<=len(&L); i++)

edit(&L,i,i,&nerror);

// Показуємо результат коригування

printf(" After ubdate:\n");

for (i=1;i<=len(&L); i++)

printf(" L[%d] =%d\n",i,retrieve(&L,i,&nerror));

// Видалення елементу № 2

del(&L, 2, &nerror);

// Показуємо результат після видалення

printf(" After delete:\n");

for (i=1;i<=len(&L); i++)

printf(" L[%d] =%d\n",i,retrieve(&L,i,&nerror));

return 0;

}

int len(FLIST \*pl)

{

return pl->last;

}

// Функція отримує зі списку елемент з номером k та повертає як результат

ELEMT retrieve (FLIST \* pl, POS k, int \*error)

{

ELEMT elem =0;

if (k<1 || k>pl->last) \*error=1;

else

{

elem = pl->elements[k];

\*error = 0;

}

return elem;

}

// Функція виконує редагування (заміну) елементу списку

void edit(FLIST \*pl,POS k, ELEMT p, int \*error)

{

if (k<1||k>pl->last) \*error=1;

else

{

pl->elements[k] = p;

\*error = 0;

}

}

// Функція вставляє елемент в позицію k списку

void insert(FLIST \*pl, POS k, ELEMT p, int \*error)

{

if (k<1||k>pl->last+1||pl->last==MAXLENGTH) \*error=1;

else

{

for (POS i=pl->last; i>=k; i--)

{

pl->elements[i+1]=pl->elements[i];

}

pl->elements[k]=p;

pl->last++;

\*error=0;

}

}

// Функція видаляє k-й елемент списку та повертає його як ре-зультат

ELEMT del(FLIST \*pl, POS k, int \*error)

{

ELEMT elem=0;

if (k<1||k>pl->last) \*error=1;

else

{

elem=pl->elements[k];

for (POS i=k; i<pl->last; i++)

{

pl->elements[i]=pl->elements[i+1];

}

pl->last--;

\*error=0;

}

return elem;

}

// Функція повертає місце знаходження в списку для запропо-нованого

// значення, або місце для нового елементу якщо запропонова-ний в списку відсутній

POS locate(FLIST \*pl,ELEMT p)

{

for (POS q = 1; q<= pl->last; q++)

if (pl->elements[q]==p) return q;

return pl->last;

}