# 顺序表实现框架程序清单

/\* Linear Table On Sequence Structure \*/

#include <stdio.h>

#include <malloc.h>

#include <stdlib.h>

/\*---------page 10 on textbook ---------\*/

#define TRUE 1

#define FALSE 0

#define OK 1

#define ERROR 0

#define INFEASTABLE -1

#define OVERFLOW -2

typedef int status;

typedef int ElemType; //数据元素类型定义

/\*-------page 22 on textbook -------\*/

#define LIST\_INIT\_SIZE 100

#define LISTINCREMENT 10

typedef struct{ //顺序表（顺序结构）的定义

ElemType \* elem;

int length;

int listsize;

}SqList;

/\*-----page 19 on textbook ---------\*/

status IntiaList(SqList & L);

//status DestroyList(SqList & L);

//status ClearList(SqList &L);

//status ListEmpty(SqList L);

//int ListLength(SqList L);

//status GetElem(SqList L,int i,ElemType & e);

//status LocateElem(SqList L,ElemType e); //简化过

//status PriorElem(SqList L,ElemType cur,ElemType & pre\_e);

//status NextElem(SqList L,ElemType cur,ElemType & next\_e);

//status ListInsert(SqList & L,int i,ElemType e);

status ListDelete(SqList & L,int i,ElemType & e);

status ListTrabverse(SqList L); //简化过

/\*--------------------------------------------\*/

void main(void){

SqList L;

int op=1;

while(op){

system("cls");

printf("\n\n");

printf(" Menu for Linear Table On Sequence Structure \n");

printf("-------------------------------------------------\n");

printf(" 1. IntiaList 7. LocateElem\n");

printf(" 2. DestroyList 8. PriorElem\n");

printf(" 3. ClearList 9. NextElem \n");

printf(" 4. ListEmpty 10. ListInsert\n");

printf(" 5. ListLength 11. ListDelete\n");

printf(" 6. GetElem 12. ListTrabverse\n");

printf(" 0. Exit\n");

printf("-------------------------------------------------\n");

printf(" 请选择你的操作[0~12]:");

scanf("%d",&op);

switch(op){

case 1:

//printf("\n----IntiaList功能待实现！\n");

if(IntiaList(L)==OK) printf("线性表创建成功！\n");

else printf("线性表创建失败！\n");

getchar();getchar();

break;

case 2:

printf("\n----DestroyList功能待实现！\n");

getchar();getchar();

break;

case 3:

printf("\n----ClearList功能待实现！\n");

getchar();getchar();

break;

case 4:

printf("\n----ListEmpty功能待实现！\n");

getchar();getchar();

break;

case 5:

printf("\n----ListLength功能待实现！\n");

getchar();getchar();

break;

case 6:

printf("\n----GetElem功能待实现！\n");

getchar();getchar();

break;

case 7:

printf("\n----LocateElem功能待实现！\n");

getchar();getchar();

break;

case 8:

printf("\n----PriorElem功能待实现！\n");

getchar();getchar();

break;

case 9:

printf("\n----NextElem功能待实现！\n");

getchar();getchar();

break;

case 10:

printf("\n----ListInsert功能待实现！\n");

getchar();getchar();

break;

case 11:

printf("\n----ListDelete功能待实现！\n");

getchar();getchar();

break;

case 12:

//printf("\n----ListTrabverse功能待实现！\n");

if(!ListTrabverse(L)) printf("线性表是空表！\n");

getchar();getchar();

break;

case 0:

break;

}//end of switch

}//end of while

printf("欢迎下次再使用本系统！\n");

}//end of main()

/\*--------page 23 on textbook --------------------\*/

status IntiaList(SqList & L){

L.elem = (ElemType \*)malloc( LIST\_INIT\_SIZE \* sizeof (ElemType));

if(!L.elem) exit(OVERFLOW);

L.length=0;

L.listsize=LIST\_INIT\_SIZE;

return OK;

}

status ListTrabverse(SqList L){

int i;

printf("\n-----------all elements -----------------------\n");

for(i=0;i<L.length;i++) printf("%d ",L.elem[i]);

printf("\n------------------ end ------------------------\n");

return L.length;

}