**线性表之静态链表实现**

#include <stdio.h>

#define MAX\_SIZE 20

#define ElemType char //规定元素类型

typedef struct ListNode {

ElemType data; //数据域

int cur; //游标域

}ListNode;

typedef ListNode StaticList[MAX\_SIZE]; //定义了一个数组类型StaticList,该类型的变量就是一个（由20个结点类型的元素构成的）数组

void InitSList(StaticList &space) {

//开辟备用空间

for (int i = 1; i < MAX\_SIZE - 1; i++) {

space[i].cur = i + 1;

}

space[MAX\_SIZE - 1].cur = 0;

}

//从备用空间里申请空间（得到一个下标值）

int Malloc\_SL(StaticList& space) {

int i = space[1].cur;

if (space[1].cur != 0) {

space[1].cur = space[i].cur;

}

return i;

}

//头插

void Insert(StaticList& space, ElemType x) {

int i = Malloc\_SL(space);

if (i == 0){

printf("申请结点空间失败");

return;

}

space[i].data = x;

//判断是否为第一个数据

if (space[0].cur == -1) {

space[i].cur = -1;

}

else {

space[i].cur = space[0].cur;

}

space[0].cur = i;

}

void ShowSList(StaticList& space) {

int i = space[0].cur;

while (i != -1) {

printf("%c-->", space[i].data);

i = space[i].cur; //p=p->next

}

printf("Nul.\n");

}

void Free\_SL(StaticList& space,int k) {

space[k].cur = space[1].cur;

space[1].cur = k;

}

//头部删除

void Delete(StaticList& space) {

int i = space[0].cur;

space[0].cur = space[i].cur;

Free\_SL(space,i);

/\*space[i].cur = space[1].cur;

space[1].cur = i;\*/

}

void main() {

StaticList SL;

InitSList(SL);

for (int i = 0; i < 5; i++) {

Insert(SL, 'A' + i);

}

ShowSList(SL);

Delete(SL);

ShowSList(SL);

}