/\*

我们通常把未被使用的数组元素称为备用链表

数组第一个元素，即下标为 0 的元素的 cur就存放备用链表的第一个结点的下标

数组的第二个元素的 cur 则存放第一个有数值的元素的下标，相当于单链表中的头结点作用

\*/

#include<stdio.h>

#include<Windows.h>

#include<stdlib.h>

#define MAXSIZE 100

typedef int ElemType;

typedef struct

{

ElemType data;

int cur;

} ComPonent,SLinkList[MAXSIZE];

void Difference(SLinkList &space, int &S);

void Free\_SL(SLinkList &space, int k);

bool InitSpace\_SL(SLinkList &space);

int Malloc\_SL(SLinkList &space);

bool Traver\_SL(SLinkList &space);

void main()

{

SLinkList space; int S;

Difference(space, S);

Traver\_SL(space);

system("pause");

}

bool InitSpace\_SL(SLinkList &space)

{

for (int i = 0; i < MAXSIZE - 1; ++i)

{

space[i].cur = i + 1;

}

space[MAXSIZE - 1].cur = 0;

return true;

}

int Malloc\_SL(SLinkList &space)

{

int i = 0;

i = space[i].cur;

if (space[0].cur)

{

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

}

return i;

}

void Free\_SL(SLinkList &space,int k)

{

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

space[0].cur = k;

}

bool Traver\_SL(SLinkList &space)

{

int i = space[1].cur;

while (i!=0)

{

printf("%d",space[i].data);

i=space[i].cur;

}

return true;

}

void Difference(SLinkList &space, int &S)

{

int m, n,j,i,b;

b=i=m = n = 0;

InitSpace\_SL(space);

S = Malloc\_SL(space);

int r = S;

scanf\_s("%d%d", &m,&n);

for ( j = 1; j <= m; ++j)

{

int i = Malloc\_SL(space);

scanf\_s("%d",&space[i].data);

space[r].cur = i;

r = i;

}

space[r].cur = 0;

for (j = 1; j <= n; ++j)

{

scanf\_s("%d",&b);

int p = S;

int k = space[S].cur;

while (k!=space[r].cur&&space[k].data!=b)

{

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

}

if (k==space[r].cur)

{

i = Malloc\_SL(space);

space[i].data = b;

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

space[r].cur = i;

}

/\* else

{

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

Free\_SL(space, k);

if (r == k) r = p;

}\*/

}

}