#include <stdio.h>

#include <stdlib.h>

typedef struct lk{

int data;

lk\* next;

}lk;

void show(lk\* l);

//2.15

lk\* link(lk\* const ha,int m,lk\* const hb,int n)

{

int k;

lk\* hc=(m<n)?ha:hb;

lk\* p=hc;

while(p->next!=NULL)

{

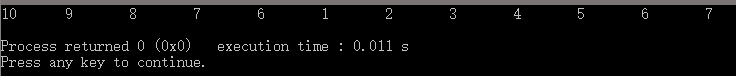
p=p->next;

}

p->next=(hc==ha)?hb->next:ha->next;

return hc;

}



//2.22

void reset(lk\* h)

{

lk\* p=NULL;

lk\* q=h->next;

while(q->next!=NULL)

{

lk\* t=q->next;

q->next=p;

p=q;

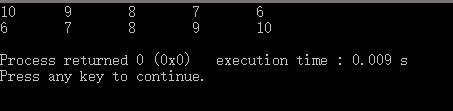
q=t;

}

q->next=p;

h->next=q;

}



//2.25

void intersection(const lk\* ha,const lk\* hb)

{

lk\* p=ha->next;

lk\* q=hb->next;

lk\* hc=(lk\*)malloc(sizeof(lk));

hc->data=0; hc->next=NULL;

lk\* t=hc;

while(p!=NULL)

{

int n=p->data;

while(q!=NULL)

{

int m=q->data;

if(m==n)

{

t->next=(lk\*)malloc(sizeof(lk));

t->next->data=n;

t->next->next=NULL;

t=t->next;

break;

}

else if(m>n) break;

else q=q->next;

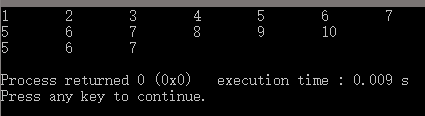
}

p=p->next;

}

show(hc);

}



void show(lk\* l)

{

while(l->next!=NULL)

{

l=l->next;

printf("%d\t",l->data);

}

printf("\n");

}

int main()

{

lk\* h1=(lk\*)malloc(sizeof(lk));

h1->next=NULL;

lk\* p=h1;

for(int i=1;i<8;++i) //链表1包含1，2，3，4，5，6，7

{

lk\* t=(lk\*)malloc(sizeof(lk));

t->data=i;

t->next=NULL;

p->next=t;

p=p->next;

}

lk\* h2=(lk\*)malloc(sizeof(lk));

h2->next=NULL;

p=h2;

for(int i=10;i>4;--i) //链表2包含10,9,8,7,6,5,4

{

lk\* t=(lk\*)malloc(sizeof(lk));

t->data=i;

t->next=NULL;

p->next=t;

p=p->next;

}

return 0;

}