# 上机练习2

## 1

程序代码：

#include <stdio.h>

#include <malloc.h>

typedef int elemtype;

typedef struct node

{

elemtype data;

struct node \*next;

}Lnode, \*linklist;

Lnode \*creat();

Lnode \*hb(Lnode \*, Lnode \*);

void print(Lnode \*);

int main()

{

Lnode \*h1, \*h2, \*h3;

h1=creat();

print(h1);

h2=creat();

print(h2);

h3=hb(h1,h2);

printf("合并后......\n");

print(h3);

return 0;

}

Lnode \*creat()

{

Lnode \*h, \*p, \*t;

elemtype num=0;

h=(Lnode\*)malloc(sizeof(Lnode));

h->next=NULL;

t=h;

printf("请输入要输入的节点数据：（为0时截止）\n");

scanf("%d",&num);

while(num!=0)

{

p=(Lnode\*)malloc(sizeof(Lnode));

p->data=num;

p->next=NULL;

t->next=p;

t=p;

scanf("%d",&num);

}

return h;

}

void print(Lnode \*h)

{

Lnode \*p;

p=h->next;

printf("此链表的所有数据为：\n");

while(p)

{

if(p->data>0)

printf("%d ",p->data);

p=p->next;

}

printf("\n");

}

Lnode \*hb(Lnode \*pa, Lnode \*pb)

{

Lnode \*p, \*q, \*pc;

pb=pb;

pa=pa;

pc=(Lnode\*)malloc(sizeof(Lnode));

p=pc;

while(pa&&pb)

{

q=(Lnode\*)malloc(sizeof(Lnode));

if(pb->data<pa->data)

{

q->data=pb->data;

pb=pb->next;

}

else

{

q->data=pa->data;

pa=pa->next;

}

p->next=q;

p=q;

}

while(pa)

{

q=(Lnode\*)malloc(sizeof(Lnode));

q->data=pa->data;

pa=pa->next;

p->next=q;

p=q;

}

while(pb)

{

q=(Lnode\*)malloc(sizeof(Lnode));

q->data=pb->data;

pb=pb->next;

p->next=q;

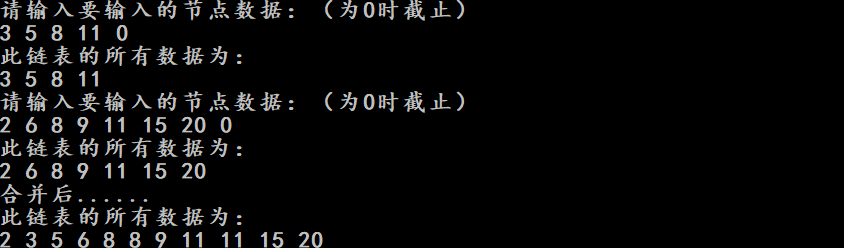
p=q;

}

p->next=NULL;

return (pc);

}运行及展示：



## 4

程序代码：

#include <stdio.h>

#include <stdlib.h>

struct node

{

int no;

struct node \*next;

};

typedef struct node NODE;

void main()

{

int i,j;

NODE\*head,\*p,\*q;

head=(NODE\*)malloc(sizeof(NODE));

head->no=-1;

head->next=head;

for(i=8;i>0;i--)

{

p=(NODE\*)malloc(sizeof(NODE));

p->no=i;

p->next=head->next;

head->next=p;

}

while(p->next!=head)

{

p=p->next;

}

p->next=head->next;

printf("此约瑟夫序列为：\n");

for(i=0;i<8;i++)

{

for(j=0;j<3;j++)

p=p->next;

q=p->next;

p->next=q->next;

printf("%3d",q->no);

free(q);

}

printf("\n");

}

运行及展示：



## 5

程序代码：

#include <stdio.h>

#include <malloc.h>

typedef int elemtype;

typedef struct node

{

elemtype data;

struct node \*next;

}Lnode, \*linklist;

Lnode \*creat();

void print(Lnode \*);

void \*I(Lnode \*, Lnode \*);

void \*U(Lnode \*, Lnode \*);

void \*D(Lnode \*, Lnode \*);

int lenth(Lnode \*);

Lnode \*search(Lnode \*,elemtype);

int main()

{

Lnode \*h1, \*h2;

h1=creat();

print(h1);

h2=creat();

print(h2);

print(I(h1,h2));

print(U(h1,h2));

print(D(h1,h2));

return 0;

}

Lnode \*creat()

{

Lnode \*h, \*p, \*t;

elemtype num=0;

h=(Lnode\*)malloc(sizeof(Lnode));

h->next=NULL;

t=h;

printf("请输入要输入的节点数据：（为0时截止）\n");

scanf("%d",&num);

while(num!=0)

{

p=(Lnode\*)malloc(sizeof(Lnode));

p->data=num;

p->next=NULL;

t->next=p;

t=p;

scanf("%d",&num);

}

return h;

}

void print(Lnode \*h)

{

Lnode \*p;

p=h->next;

printf("长度为%d的链表的所有数据为：\n",lenth(h));

while(p)

{

if(p->data>0)

printf("%d ",p->data);

p=p->next;

}

printf("\n");

}

void \*I(Lnode \*a, Lnode \*b)

{

Lnode \*t, \*c, \*p, \*c0;

p=a->next;

c=(Lnode\*)malloc(sizeof(Lnode));

c->next=NULL;

t=c;

while(p)

{

if(search(b,p->data))

{

c0=(Lnode\*)malloc(sizeof(Lnode));

c0->data=search(b,p->data)->data;

c0->next=NULL;

t->next=c0;

t=c0;

}

p=p->next;

}

return (c);

}

void \*U(Lnode \*a, Lnode \*b)

{

Lnode \*t, \*c, \*p, \*q, \*c0;

p=a->next;

q=b->next;

c=(Lnode\*)malloc(sizeof(Lnode));

c->next=NULL;

t=c;

while(p)

{

if(search(b,p->data))

{

c0=(Lnode\*)malloc(sizeof(Lnode));

c0->data=search(b,p->data)->data;

c0->next=NULL;

t->next=c0;

t=c0;

}

else

{

c0=(Lnode\*)malloc(sizeof(Lnode));

c0->data=p->data;

c0->next=NULL;

t->next=c0;

t=c0;

}

p=p->next;

}

while(q)

{

if(!search(c,q->data))

{

c0=(Lnode\*)malloc(sizeof(Lnode));

c0->data=q->data;

c0->next=NULL;

t->next=c0;

t=c0;

}

q=q->next;

}

return (c);

}

void \*D(Lnode \*a, Lnode \*b)

{

Lnode \*t, \*c, \*p, \*c0;

p=a->next;

c=(Lnode\*)malloc(sizeof(Lnode));

c->next=NULL;

t=c;

while(p)

{

if(!search(I(a,b),p->data))

{

c0=(Lnode\*)malloc(sizeof(Lnode));

c0->data=p->data;

c0->next=NULL;

t->next=c0;

t=c0;

}

p=p->next;

}

return (c);

}

Lnode \*search(Lnode \*h,elemtype x)

{

Lnode \*p;

p=h->next;

while(p&&p->data!=x)

p=p->next;

return (p);

}

int lenth(Lnode \*h)

{

Lnode \*p;

int i=0;

p=h->next;

while(p)

{

i++;

p=p->next;

}

return i;

}

运行结果及展示：

