**《程序设计课程实践》设计文档**

# 作业题目：1.3

**学号：\_\_\_\_\_\_\_\_\_\_\_\_\_\_19151633\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**姓名：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_应宇杰\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

完成时间 2021年 3 月 15 日

第\_\_三\_\_\_次作业 (写上第几次作业)

题目\_\_\_\_\_\_1.3\_\_（写上题目号）

提交代码：

//程序功能：：将正数插入第一个单链表，负数插入第二个单链表 ,递增顺序,然后倒叙再输出

#include<stdio.h>

#include<stdlib.h>

typedef struct node{

int data;

struct node \*next;

}list;

//逆置链表

void insert\_reverse(list \*head){

list \*p=head->next;

if(p){

list \*q=p->next;

head->next=NULL;

while(p){

p->next=head->next;

head->next=p;

if(q){

p=q;

q=p->next;

}else{

break;

}

}

}

}

//插入数字

void insert\_number(list \*head,int number){

list\* nodes,\*p;

nodes=p=(list\*)malloc(sizeof(list));

nodes->data=number;

nodes->next=NULL;

p=head;

while(p->next!=NULL&&p->next->data<number){

p=p->next;

}

if(p->next==NULL){

p->next=nodes;

p->next->next=NULL;

}else{

nodes->next=p->next;

p->next=nodes;

}

}

void insert\_print(list \*head1,list \*head2){

//输出正数

printf("输出正数: \n") ;

while(head1->next!=NULL){

printf("%d ",head1->next->data);

head1=head1->next;

}

printf("\n");

//输出负数

printf("输出负数: \n") ;

while(head2->next!=NULL){

printf("%d ",head2->next->data);

head2=head2->next;

}

printf("\n");

}

void insert\_merge(list \*head1,list \*head2){

while(head2->next!=NULL){

head2=head2->next;

}

head2->next=head1->next;

head1->next=NULL;

}

void insert(int n){

list \*head1;

list \*head2;

head1 = (list\*)malloc(sizeof(list));

head2 = (list\*)malloc(sizeof(list));

head1->next=NULL;

head2->next=NULL;

int number;

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

scanf("%d",&number);

if(number>0){

insert\_number(head1,number);

}else{

insert\_number(head2,number);

}

}

//整合两个链表

//insert\_merge(head1,head2);

insert\_print(head1,head2);

//逆置

printf("逆置: \n");

insert\_reverse(head1);

insert\_reverse(head2);

insert\_print(head1,head2);

//销毁

free(head1);

free(head2);

}

int main(){

int n,number;//代表输入的数字数量

printf("要输入多少个数字 ：");

scanf("%d",&n);

insert(n);

return 0;

}

运行结果：

（可以截图）

