## 习题3-11

#include<stdio.h>

#include<stdlib.h>

#include<conio.h>

//单链表存入字符部分

#define MAXSIZE 1024

typedef char datetype;

typedef struct node

{

datetype data;

struct node \*next;

}linklist;

linklist \*L;

//建立单链表

linklist \*Creat()

{

char ch;

linklist \*head,\*s,\*r;

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

r=head;

while((ch=getche())!='#')

{

s=(linklist\*)malloc(sizeof(linklist));

s->data=ch;

r->next=s;

r=s;

}

r->next=NULL;

return head;

}

//栈运算判断部分

typedef char datatype2;

struct Node{

datatype2 element;

struct Node \*next;

};

struct Node \*Top;

//进栈

struct Node \*PushL(struct Node \*Top,datatype2 e){

struct Node \*p;

p=(struct Node\*)malloc(sizeof(struct Node));

p->element=e;

p->next=Top;

Top=p;

return Top;

}

//出栈

datatype2 \*PopL(struct Node \*\*Top){

datatype2 \*ret;

if(\*Top == NULL){

printf("Stack is underflow\n");

return NULL;

}

else{

ret=(datatype2\*)malloc(sizeof(datatype2));

\*ret=(\*Top)->element;

\*Top=(\*Top)->next;

return ret;

}

}

void main(){

L=Creat();

linklist \*p,\*q;

struct Node \*(\*Top2);

datatype2 \*r=(datatype2\*)malloc(sizeof(datatype2));

Top=(struct Node\*)malloc(sizeof(struct Node));

Top2=&Top;

int i,j,k=1,flag=1;

p=L;

for(i=0;p->next != NULL;i++){

p=p->next;

}

p=L->next;

for(j=0;j<i/2;j++){

Top=PushL(Top,p->data);

p=p->next;

}//一半字符依次进栈

if(i%2!=0) p=p->next;

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

{

r=PopL(Top2);//依次出栈比较

if(\*r==p->data){

p=p->next;

continue;

}

else{

flag=0;

break;

}

}

if(flag) printf("中心对称\n");

else printf("非中心对称\n");

}

## 习题3-16

#include<stdio.h>

#define m 5

char sequ[m];

int rear=-1,quelen=0;

// 队满条件为(quelen==m)

//入队

void EnSequ(char a[],char r){

if(quelen==m) printf("Overflow\n");//判断是否队满

else{

a[(rear+1+m)%m]=r;

quelen++;

rear=(rear+1)%5;

}

}

//出队

char DeSequ(char a[]){

char t;

t=a[(m+rear+1-quelen)%m];

quelen--;

return t;

}

//打印队列

Print(char a[]){

int i,j;

for(i=rear,j=0;j<quelen;i--,j++)

{

printf("%c ",sequ[(i+5)%5]);

}

printf("\n");

}

void main(){

//以下为测试代码

int i,j;

EnSequ(sequ,'a');

EnSequ(sequ,'b');

EnSequ(sequ,'c');

EnSequ(sequ,'d');

EnSequ(sequ,'e');

Print(sequ);

EnSequ(sequ,'f');//将提示溢出

DeSequ(sequ);//'a'出队

Print(sequ);

EnSequ(sequ,'f');//此时'f'可入队

Print(sequ);

}