中缀表达式转变为后缀表达式

#include<stdio.h>

#include<stdlib.h>

#define MAX 20

#define ADD 10

typedef char ElemType;

typedef struct

{ ElemType \*base;

ElemType \*top;

int stackSize;

}SeqStack;

SeqStack \*s;

//置空栈

Init\_SeqStack(SeqStack \*s)

{ s->base=(ElemType \*)malloc(MAX \* sizeof(ElemType));

if(!s->base) return 0;

s->top=s->base;

s->stackSize=MAX;

}

//入栈

push(SeqStack \*s,ElemType e)

{ if(s->top-s->base>=s->stackSize)

{

s->base=(ElemType \*)realloc(s->base,(s->stackSize+ADD)\* sizeof(ElemType));

if(!s->base) return 0;

s->top=s->base+s->stackSize;

s->stackSize=s->stackSize+ADD;

}

\*(s->top)=e;

s->top++;

}

//出栈

pop(SeqStack \*s,ElemType \*e)

{ if(s->top==s->base)

return 0;

else {\*e=\*--(s->top);}

}

int st(SeqStack s)

{ return (s.top - s.base);

}

int main()

{ SeqStack s;

char c,e;

Init\_SeqStack(&s);

scanf("%c",&c) ;

while(c!='#')

{ while( c>='0' && c<='9' )

{

printf("%c", c);

scanf("%c", &c);

if( c<'0' || c>'9' )

{

printf(" ");

}

}

if( ')' == c )

{

pop( &s,&e);

while( '(' != e )

{

printf("%c ", e);

pop(&s,&e) ;

}

}

else if( '+'==c || '-'==c )

{

if( !st(s) )

{

push(&s, c);

}

else

{

do

{

pop(&s,&e) ;

if( '(' == e )

{

push(&s, e);

}

else

{

printf("%c ", e);

}

}while( st(s)&& '('!=e );

push(&s, c);

}

}

else if( '\*'==c || '/'==c )

{

if( !st(s) )

{

push(&s, c);

}

else

{

do

{

pop(&s, &e);

if( '(' == e || '+'== e || '-'== e )

{

push(&s, e);

}

else

{

printf("%c ", e);

}

}while( st(s) && '('!=e &&'+'!=e && '-'!=e);

push(&s, c);

}

}

else if( '^'==c || '('==c )

{

push(&s, c);

}

else if( '#'== c )

{

break;

}

else

{

printf("\n输入错误！\n");

return -1;

}

scanf("%c", &c);

}

while( st(s) )

{

pop(&s,&e);

printf("%c ", e);

}

return 0;

}