栈的应用 中缀表达式

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

typedef struct node

{ char data;

int d;

struct node \*next;

}StackNode,\*LinkStack;

main()

{ char equ[30],ch;

int i,j,q,c=0,t=0;

StackNode \*h,\*y;

LinkStack top=NULL,first=NULL;

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

y=(StackNode\*)malloc(sizeof(StackNode));

y->data='(';

y->next=first;

first=y;

y=(StackNode\*)malloc(sizeof(StackNode));

scanf("%s",equ);

for(i=0;equ[i]!='\0';i++)

{ //运算数字入对象栈

if(equ[i]>='0'&&equ[i]<='9')

{ printf("进入数字栈\n");//可删

printf("入栈的数字为%d\n",int(equ[i])-48);//可删

h->d=int(equ[i])-48;

h->next=top;

top=h;

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

}

//运算符号根据优先级情况入算符栈

else if((first->data=='('||(first->data=='+'||first->data=='-'||equ[i]=='\*'||equ[i]=='/')||equ[i]=='(')&&equ[i]!=')')

{ printf("进入符号栈\n");//可删

if(equ[i]=='(')

t=1; //用来判断运算式里是否输入了括号

y->data=equ[i];

y->next=first;

first=y;

c++;

printf("符号栈里符号数为%d\n",c);//可删

printf("入栈的符号为%c\n",first->data);//可删

y=(StackNode\*)malloc(sizeof(StackNode));

}

//若运算表达式里有括号，特殊考虑

else if(equ[i]==')'&&t==1)

{ for(;;)

{ ch=first->data;

printf("出栈的运算符号为%c\n",first->data);//可删

switch(ch)

{ case '+':q=top->next->d+top->d;break;

case '-':q=top->next->d-top->d;break;

case '\*':q=top->next->d\*top->d;break;

case '/':q=top->next->d/top->d;break;

}

printf("出栈的栈顶数字为%d\n",top->d);//可删

printf("出栈的第二个数字为%d\n",top->next->d);//可删

printf("出栈两个数字和一个运算符\n");//可删

printf("栈顶的运算符为%c\n",first->data);//可删

top=top->next->next;

h->d=q;

h->next=top;

top=h;

c--;

printf("出栈运算得%d\n",q);//可删

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

first=first->next;

if(first->data=='(') //当遇到左括号时，结束出栈

{ first=first->next;

c--;

break;

}

y=(StackNode\*)malloc(sizeof(StackNode));

}

}

else if((first->data=='\*'||first->data=='/')&&(equ[i]=='+'||equ[i]=='-'))

{ i--;

printf("出栈的运算符为%c\n",first->data);//可删

ch=first->data;

switch(ch)

{ case '+':q=top->next->d+top->d;break;

case '-':q=top->next->d-top->d;break;

case '\*':q=top->next->d\*top->d;break;

case '/':q=top->next->d/top->d;break;

}

printf("出栈两个数字和一个运算符\n");//可删

top=top->next->next;

h->d=q;

h->next=top;

top=h;

printf("因运算符号优先级优先运算得%d\n",q);//可删

c--;

printf("符号栈里符号数为%d\n",c);//可删

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

first=first->next;

y=(StackNode\*)malloc(sizeof(StackNode));

}

}

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

{ ch=first->data;

printf("出栈的符号为%c\n",first->data);//可删

switch(ch)

{ case '+':q=top->next->d+top->d;break;

case '-':q=top->next->d-top->d;break;

case '\*':q=top->next->d\*top->d;break;

case '/':q=top->next->d/top->d;break;

}

printf("出栈的栈顶数字为%d\n",top->d);//可删

printf("出栈的第二个数字为%d\n",top->next->d);//可删

printf("出栈两个数字和一个运算符\n");//可删

top=top->next->next;

h->d=q;

h->next=top;

top=h;

printf("出栈运算得%d\n",q);//可删

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

first=first->next;

y=(StackNode\*)malloc(sizeof(StackNode));

}

printf("运算结果：%s=%d",equ,q);

}