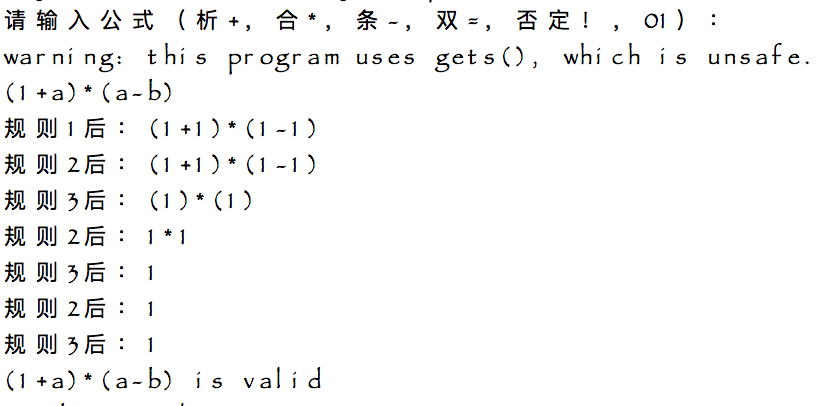
w2判断公式的合法性

1. 运行结果：



1. 代码改进与注释（红字为此段伪代码解释，红字加下划线为修改过的代码解释，//黑字为单句解释）

void rule1(char a[],int i)

{

//如果是单个字符，合法表达式则为1

if((a[i]>='a')&&a[i]<='z')

a[i]='1';

//若为常量，则合法表达式为1

else if(a[i]=='0')

a[i]='1';

}

这里是把a或者A或者0都变成1

void rule2(char a[],int i)

{

int n=strlen(a);

//如果当前位置起形如“！1”则换成“1”

if((i+1<n)&&(a[i]=='!')&&(a[i+1]=='1'))

{

a[i]='1';

i++;

while(a[i+1]!='\0')

{

a[i]=a[i+1];//后面的字符往前移1格

i++;

}

a[i]='\0';

}

//如果当前位置起形如“（1）”则换成“1”

else if((i+2<n)&&(a[i]=='(')&&(a[i+1]=='1')&&a[i+2]==')')

{

a[i]='1';

i++;

while(a[i+2]!='\0')

{

a[i]=a[i+2];

i++;

}

a[i]='\0';

}

}

像（1）或者！1都变成1，因为这是合法公式，用1来代替就好

这里修改了，把返回的result给去掉了，直接变成void函数

void rule3(char a[],int i)

{

//1+1,1\*1,1-1,1=1均换成1

int n=strlen(a);

if((i+2<n)&&(a[i]=='1')&&((a[i+1]=='\*')||(a[i+1]=='=')||(a[i+1]=='-')||(a[i+1]=='+'))&&(a[i+2]=='1'))

{

a[i]='1';

//后面的字符往前移2格

i++;

while(a[i+2]!='\0')

{

a[i]=a[i+2];

i++;

}

a[i]='\0';

}

}

把一个完整的运算公式都变成1，因为这是合法公式，用1来代替就好

这里修改了，把返回的result给去掉了，直接变成void函数，并且，为了减少代码长度，把+\*-=都放在一个函数里面写。

int main()

{

char pstate[120],pstate0[120];

int i=0,nold=1,nnew=0;

printf("请输入公式（析+，合\*，条-，双=，否定！，01）：\n");

gets(pstate0);

fflush(stdin);

这里修改，直接不要nold和nnew两个变量，用strlen（pstate0）代替

for(i=0;i<strlen(pstate0);i++)

pstate[i]=pstate0[i];

pstate[i]='\0';

for(i=0;i<strlen(pstate);i++)

rule1(pstate,i);

printf("规则1后：%s\n",pstate);

while(nnew<nold)

{

nold=strlen(pstate);

//使用规则2对字串进行处理，会改变长度

for(i=0;i<strlen(pstate);i++)

rule2(pstate,i);

printf("规则2后：%s\n",pstate);

for(i=0;i<strlen(pstate);i++)

rule3(pstate,i);

printf("规则3后：%s\n",pstate);

nnew=strlen(pstate);

//if(nnew==1&&pstate=='1')

//break;

}

这里修改：由于之前函数不是int形式而是void形式，所以把while换成for（i<strelen(pstate)）

if((pstate[0]=='1')&&(strlen(pstate)==1))

printf("%s is valid\n",pstate0);

else

printf("%s is invalid\n",pstate0);

return 0;

}