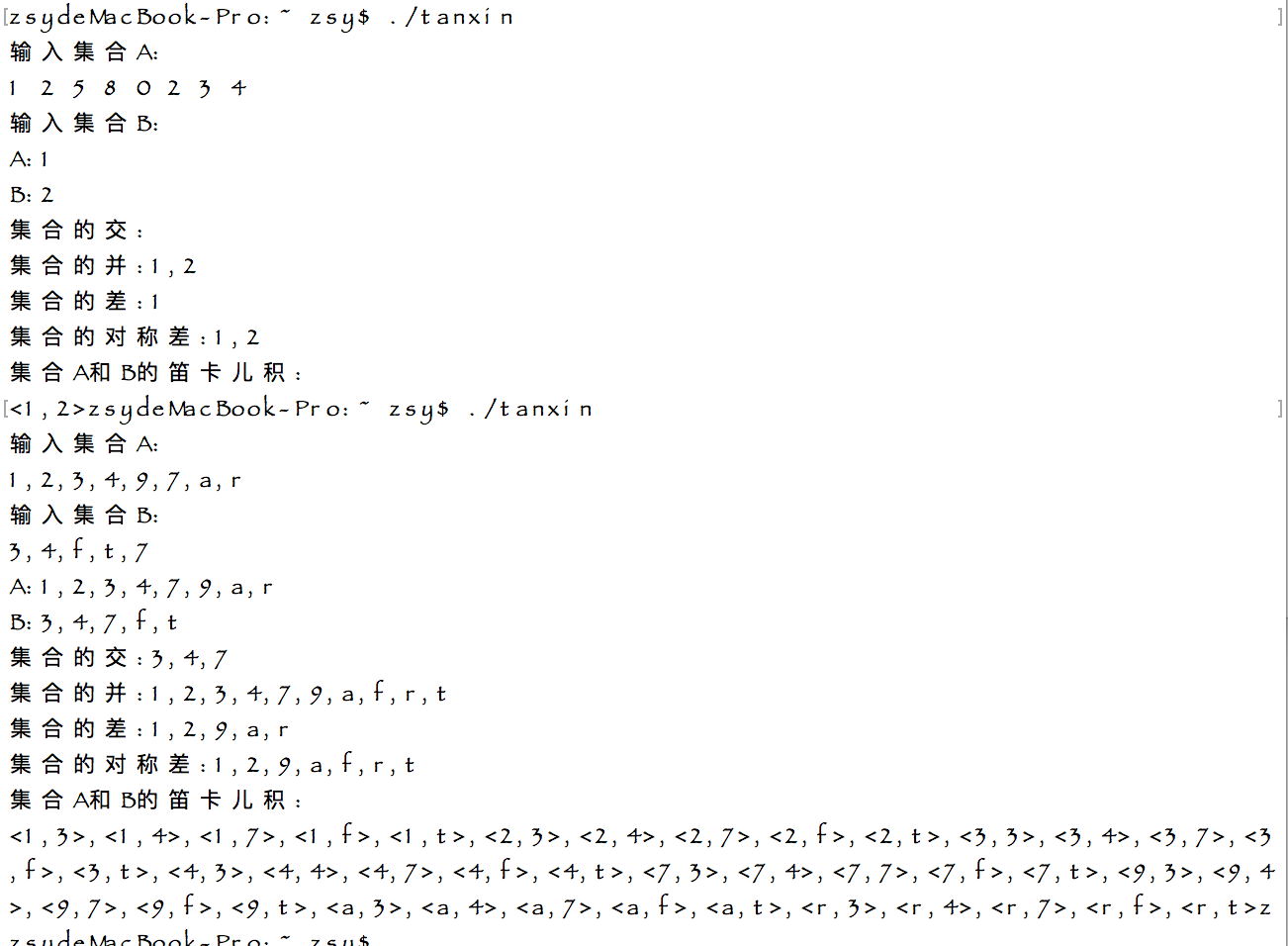
w2判断公式的合法性

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）

#include<iostream>

using namespace std;

long long InputSet(string str)

{

int i,k;

long long c=0,y=1;

for(i=0;i<str.length();i++)

{

if((str[i]>='0')&&(str[i]<='9'))k=str[i]-'0';

else if((str[i]>='a')&&(str[i]<='z'))k=str[i]-'a'+10;

else if((str[i]>='A')&&(str[i]<='Z'))k=str[i]-'A'+10+26;

else k=-1;

if(k>=0)c=(c|y<<k);

**k等于数字或字母的时候，c存的是每一位是否有值！**

**类似100010101111100000……这样的序列，用string存下来**

**Y是1，移动k位（k是acsii码）**

**C为什么要| y<<k？**

**因为在for循环里面，每多一个集合里面的元素，就把这个元素的位置标记上1。**

}

return c;

}

char SetI2C(int i)

{

char c;

if(i<10)c='0'+i;

else if(i<36)c='a'+i-10;

else c='A'-10-26+i;

return c;

}

string OutputSet(long long a)

{

string str="";

long long y=1;

for(int i=0;i<62;i++)

{

if(a&(y<<i))str=str+SetI2C(i)+',';

}

return (str=="")?"":str.substr(0,str.length()-1);

}

**这个相当于翻译，或者说解码，把之前保存的插旗标记数翻译成a-z，A-Z和数字**

string OutputSetDCL(long long a,long long b)

{

string str="";

char ca,cb;

long long c,y=1;

int i,j;

for(i=0;i<62;i++)

{

if(!(a&(y<<i)))continue;

ca=SetI2C(i);

for(j=0;j<62;j++)

{

if(b&(y<<j))str=str+'<'+ca+','+SetI2C(j)+">,";

**求积**

}

}

return (str=="")?"":str.substr(0,str.length()-1);

}

int main()

{

long long a=0,b=0;

string sa,sb;

cout<<"输入集合A:"<<endl;

cin>>sa;

a=InputSet(sa);

cout<<"输入集合B:"<<endl;

cin>>sb;

b=InputSet(sb);

cout<<"A:"<<OutputSet(a)<<endl;

cout<<"B:"<<OutputSet(b)<<endl;

cout<<"集合的交:"<<OutputSet(a&b)<<endl;

**直接把之前的两个标记数组相与，就知道有哪些交集了，用outputset来翻译成集合的形式**

cout<<"集合的并:"<<OutputSet(a|b)<<endl;

**直接把之前的两个标记数组相或，用outputset来翻译成集合的形式**

cout<<"集合的差:"<<OutputSet(a&~(a&b))<<endl;

cout<<"集合的对称差:"<<OutputSet((a|b)&~(a&b))<<endl;

cout<<"集合A和B的笛卡儿积:"<<endl;

cout<<OutputSetDCL(a,b);

return 0;

}