写最后一题的解题报告之前，想说两句。

这次中秋的热身赛，主要是这样几个目的：首先让同学预先熟悉熟悉上机环境；其次，给一些虽然没有什么基础，但是感兴趣并进行了一定的预习的同学一个练习的契机；然后，由于助教对倪闷的情况并不是十分了解，于是也想看一看哪些同学是有基础的，以后可以在上机的时候对大家有些帮助。

这次上机前八道题都不需利用数组（虽然个别题使用数组会更加方便），也就是说采用前四章的知识，几乎完全可以解决。此外，一些题诸如B题的复杂算式化简、E题的素数三元组的唯一性、以及C题的放盘子，这几道题是想向大家传达一种思想——上机不是机械化的计算，一些题目在做之前或者多次错误 ，不妨先在草纸上演算演算，看看是否有什么好的思路。就拿第二题来说吧，如果稍稍化简一下就可以看出，这题和第一题完全没有任何区别。

九十题是针对那些有基础的大神来写的，第九题应用了栈和链表的相关知识，而第十题考的是模拟算法（夹杂着一些对字符串的理解）。所以对于大部分还没有深入学习数据结构和算法甚至对C++也仍是初来乍到的人来说，做不出来是很正常的。

不过也不需担忧或者有什么心理负担。首先，上机的题目会考虑绝大多数人的水平要求，难度将远远比不上这次的题目；其次，只要认真对待这门课程，平时用心去练习，多做题多看书多思考，一个学期后，大多数人的成绩都会很理想的，甚至也会有一些新的大神，拔地而起，顶天立地。（向沛伦学长致敬！）

最后，祝大家在国庆回来的上机中，刷题愉快（题目真的很水的哦~~）

好了，闲话就到这里，请看题解：

Chem is Try！

# Description

如果你来到这里，那么说明你距离胜利已经不远了。  
所以，就让暴风雨来的更猛烈些吧——  
请听题——  
给你一个化学方程式，请你判断方程式是否配平。  
没有气体或者沉淀条件等符号。等号用“=”表示，保证不混入可逆反应、离子方程式、有机反应等稀奇古怪的东西\~^O^~/元素均为前四周期元素。

# Input

多组测试数据，每组测试数据为一行，包含一个字符串，为对应化学方程式  
。系数均为整数

# Output

对于每组数据，输出一行：  
如果方程式配平，则输出“YES”  
如果未配平，则输出“NO”

# Sample Input

2H2O=2H2+O2  
CH4+2O2=H2O+CO2  
2Mg+O2=2MgO

# Sample Output

YES  
NO  
YES

# Hint

注意括号呦0w0

# 解题分析：

理论上这题其实没什么像前面几道题那样的技术含量，就是对字符串的处理和各种情况的判断很坑爹。

由于已提出了各种限制条件，所以本题不需考虑化合物等是否合理，只需要判断物质守恒即可。这里注意，本题的核心解决思路就是——物质守恒

如此一来就简单多了。由于只涉及前四周期的元素，因此不妨进行打表，通过一个函数来将元素转换成原子序数存放在数组里。然后，通过对等号左面、右面计算元素的和并分别存储在两个表中，最后对两个表中的记录进行比较即可。

比较坑爹的是存在诸如（NH4）2CO3这样的带有括号的东西。于是渣诚的思路是使用三个计数器，一个统计整个化合物前面的计量数，一个统计括号后的计量数，一个统计元素后的计量数。对于括号，将元素后的计量数乘以括号后的计量数，再乘以化合物前面的计量数，就是这一部分元素的计量数。

# 参考代码：

#include<iostream>

#include<string>

#include<cstdio>

#include<cstring>

using namespace std;

int check(char c[])

{

if(strcmp(c,"H")==0) return 0;

else if(strcmp(c,"He")==0) return 1;

else if(strcmp(c,"Li")==0) return 2;

else if(strcmp(c,"Be")==0) return 3;

else if(strcmp(c,"B")==0) return 4;

else if(strcmp(c,"C")==0) return 5;

else if(strcmp(c,"N")==0) return 6;

else if(strcmp(c,"O")==0) return 7;

else if(strcmp(c,"F")==0) return 8;

else if(strcmp(c,"N")==0) return 9;

else if(strcmp(c,"Na")==0) return 10;

else if(strcmp(c,"Mg")==0) return 11;

else if(strcmp(c,"Li")==0) return 12;

else if(strcmp(c,"Si")==0) return 13;

else if(strcmp(c,"P")==0) return 14;

else if(strcmp(c,"S")==0) return 15;

else if(strcmp(c,"Cl")==0) return 16;

else if(strcmp(c,"Ar")==0) return 17;

else if(strcmp(c,"K")==0) return 18;

else if(strcmp(c,"Ca")==0) return 19;

else if(strcmp(c,"Sc")==0) return 20;

else if(strcmp(c,"Ti")==0) return 21;

else if(strcmp(c,"V")==0) return 22;

else if(strcmp(c,"Cr")==0) return 23;

else if(strcmp(c,"Mn")==0) return 24;

else if(strcmp(c,"Fe")==0) return 25;

else if(strcmp(c,"Co")==0) return 26;

else if(strcmp(c,"Ni")==0) return 27;

else if(strcmp(c,"Cu")==0) return 28;

else if(strcmp(c,"Zn")==0) return 29;

else if(strcmp(c,"Ga")==0) return 30;

else if(strcmp(c,"Ge")==0) return 31;

else if(strcmp(c,"As")==0) return 32;

else if(strcmp(c,"Se")==0) return 33;

else if(strcmp(c,"Br")==0) return 34;

else if(strcmp(c,"Kr")==0) return 35;

else return 36;

}

void cal(int name[],char st[],int top)

{

int counter1=1,counter2[100],counter3=1;

char stack[20];

int top0=-1,front=0,numberOfBracket=-1;

char temp[3];

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

{

counter1=0;

while(st[front]>='0'&&st[front]<='9')

counter1=counter1\*10+(st[front++]-'0');

}

for(int i=0;i<100;i++)

counter2[i]=1;

for(int i=1;i<=top;i++)

if(st[i-1]==')')

{

counter2[++numberOfBracket]=0;

while(st[i]>='0'&&st[i]<='9'&&i<=top)

counter2[numberOfBracket]=counter2[numberOfBracket]\*10+(st[i++]-'0');

}

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

if(st[i]<'0'||st[i]>'9')

stack[++top0]=st[i];

else if(i!=0&&st[i-1]!=')')

stack[++top0]=st[i];

while(top0!=-1)

{

if(stack[top0]==')')

{

top--;counter3=1;

while(stack[top0]!='(')

{

if(stack[top0]>='0'&&stack[top0]<='9') counter3=stack[top0--]-'0';

else if(stack[top0]>='a'&&stack[top0]<='z')

{for(int i=0;i<3;i++) temp[i]='\0';temp[1]=stack[top0--];temp[0]=stack[top0--];name[check(temp)]+=counter3\*counter1\*counter2[numberOfBracket];counter3=1;}

else

{for(int i=0;i<3;i++) temp[i]='\0';temp[0]=stack[top0--];name[check(temp)]+=counter3\*counter1\*counter2[numberOfBracket];counter3=1;}

}

numberOfBracket--;

}

else

{

if(stack[top0]>='0'&&stack[top0]<='9') counter3=stack[top0--]-'0';

else if(stack[top0]>='a'&&stack[top0]<='z')

{for(int i=0;i<3;i++) temp[i]='\0';temp[1]=stack[top0--];temp[0]=stack[top0--];name[check(temp)]+=counter3\*counter1;counter3=1;}

else

{for(int i=0;i<3;i++) temp[i]='\0';temp[0]=stack[top0--];name[check(temp)]+=counter3\*counter1;counter3=1;}

}

}

}

int main()

{

string a;

while(getline(cin,a))

{

int namel[37],namer[37];

for(int i=0;i<36;i++)

namel[i]=namer[i]=0;

int l=0,counter1=1;

while(a[l]!='=')

{

char stack[20];

int top=-1;

while(a[l]!='+'&&a[l]!='=')

stack[++top]=a[l++];

cal(namel,stack,top);

if(a[l]=='+') l++;

}l++;

while(l<a.length())

{

char stack[20];

int top=-1;

while(a[l]!='+'&&l<a.length())

stack[++top]=a[l++];

cal(namer,stack,top);

if(a[l]=='+') l++;

}

bool equal=true;

for(int i=0;i<36;i++)

if(namel[i]!=namer[i])

{equal=false;break;}

equal?cout<<"YES\n":cout<<"NO\n";

}

return 0;

}