真·人工智障·方程式配平器

全年龄向·人民群众喜闻乐见的娱乐方式

多次试图以降低用户体验的方式减少工作量

已加入1810豪华午餐

Description

仅供娱乐

~~不论如何都不能让迟老师知道这份代码的存在~~

~~虽然这东西什么用都没有但是不论如何都不能~~

Input

输入两行到文件$balancer.in$

如果你觉得文件输入很麻烦的话，可以在代码中找到这句话

```
freopen("balancer.in","r",stdin);
```

然后像下面这样把它注释掉，就可以直接输入了

```
//freopen("balancer.in","r",stdin);
```

第一行输入反应物，第二行输入生成物

物质间用空格作为分隔符

大概为半$markdown$格式，无下角标，大小写敏感

也就是说如果你想要表示

$Ca(OH)_2+CO_2$

请输入

$Ca(OH)2\quad CO2$

支持离子方程式~~(大概吧)~~

同样大小写敏感，但需要输入字符 ^ 作为物质与上标间的分隔符

也就是说如果你想要表示

OH^-+H^+

请输入

$OH\text{^}-\quad H\text{^}+$

Output

~~输出反正是由我来做为什么我要在这里开一个版面~~

以$markdown$格式输出到文件$balancer.md$中

~~所以如果你的电脑上没有markdown编辑器你就别玩子~~

如果你的电脑上没有$markdown$编辑器的话，你可以把输出文件拷贝到一些在线编辑器上，例如说马克飞象

当然你也可以选择修改输出格式与输出文件，~~如果你有兴趣看这个又长又臭的源码的话~~

具体操作大概是，先找到426行的这句

```
freopen("balancer.md","w",stdout);
```

改成下面这样

```
freopen("balancer.txt","w",stdout);
```

然后往上翻到405行的样子，找到下面的这个地方

```
printf("$$");
int print_tot=0;
for (int i=1;i<=tot;++i){
    if ((x[i].x/k)!=1) printf("%d",x[i].x/k),print_tot+=calc(x[i].x/k);
    for (int j=0,ssz=ss[i].size();j<ssz;++j){
        printf("%c",ss[i][j]);
        if (ss[i][j]!='_' && ss[i][j]!='^' && ss[i][j]!='{' && ss[i][j]!='}')
            ++print_tot;
    }
    if (i==tt) printf("\\xlongequal{\\quad\\quad}"),++print_tot;
    else if (i^tot) printf("+"),++print_tot;
    if (print_tot>=70){
        printf("$$\n$$");
        print_tot=0;
    }
}
printf("{}$$\n\n");
```

把上面那个很丑的代码改成这样

```
for (int i=1;i<=tot;++i){
    printf("%d ",x[i].x/k);
    if (i==tt || i==tot) printf("\n");
}
```

然后这份代码就会输出两行数字

第一行对应反应物，第二行对应生成物，和输入顺序一致

Sample Input

$Fe3C \quad HNO3$

$Fe(NO3)3 \quad CO2 \quad NO2 \quad H2O$

Sample Output

$Fe_3C + 22HNO_3 ==== 3Fe(NO_3)_3 + CO_2 + 13NO_2 + 11H_2O$

Titbits

这份代码存在一些（或者说很多）局限性

一开始我准备用纯markdown格式输入，但是这样可能因为太繁琐而对非信息组同学比较不友好，~~而且似乎并不会方便我的处理~~，具体格式可以参见输出文件或者本文档的源码

理论上来讲只要参与反应的化学物质多于参加反应的元素数$+1$这个程序就会直接崩了，~~但因为水平问题我没想到这样的方程式~~

物质不能带结晶水（当然如果你非要写结晶水你也可以把xH_2O写成$H_{2x}O_x$，而且不能写那个·），也不能够处理像下面这种人都搞不定的

$C + O_2 ==== CO + CO_2$

如果世界上真的存在某些同学说的那种原子平了电荷不能平的普通方程式

~~那我也没办法反正娱乐一下就好~~

当然对于多解的情况我做了一些非常粗略的判定

（你可以尝试一下输入上面那个方程式看看会出来什么）

所以应该也不会跑出来什么运行时错误

希望现场能有人发现这份代码中，信息学层面的问题（如你让这份代码跑出了运行时错误）

~~酌情考虑和各组竞赛教练抢人~~

总而言之祝各位玩得开心

Principle

关于这部分内容，如果你并不很想把这个程序玩坏那么就并不是很需要看了

原理很简单，对于一个离子方程式我们优先平电荷，其他情况下我们考虑平原子

这个程序并没有识别元素的能力，出现的所有元素都以字符串的形式存储，因为大小写敏感，所以一种元素会被看成一个第一位为大写字母后面都是小写字母的串（也就是说你可以输入一些.....别的东西，详情见文件$balancer.in$的最后两行），每种化学物质中某种原子的数量用Map存下来，同时我们用一个全局的Map存下所有出现过的元素

对于一个离子方程式我们额外记下每种物质的电荷数

接下来我们开始列方程

离子方程式优先列出电荷数的等量关系，之后按某种顺序依次考虑每一种元素，列出关于这种元素的等量关系，同时检查之前列出的所有方程，如果存在某个方程和现在列出的这个完全等价就把它丢掉

这个时候我们发现并不能解出具体的值，所以我们随便选一个化学物质，钦定它的系数为一，然后写一个分数类做高斯消元即可

不难发现当参与反应的化学物质多于参加反应的元素数$+1$时，我们并列不出这么多的方程，而同时根据数学知识我们也知道这个方程会有多解，然后这个程序就会报错

这个程序唯一可能跑挂的地方应该是在存在多解的时候会出现$x/0$这样的东西，我在求gcd的时候判掉了这个

~~如果你找到了其它问题那可以说是真的非常强了~~

希望有人能够造出这个程序本应该跑出来但实际上会挂掉的数据吧

$Disposrestfully$