

# #A、模拟计算器

## 问题描述

模拟程序型计算器，依次输入指令，可能包含的指令有

数字：NUM X，X 为一个只包含大写字母和数字的字符串，表示一个当前进制的数

运算指令：ADD,SUB,MUL,DIV,MOD，分别表示加减乘，除法取商，除法取余

进制转换指令：CHANGE K，将当前进制转换为 K 进制

输出指令：EQUAL，以当前进制输出结果

重置指令：CLEAR，清除当前数字

指令按照以下规则给出：

数字，运算指令不会连续给出，进制转换指令，输出指令，重置指令有可能连续给出

运算指令后出现的第一个数字，表示参与运算的数字。且在该运算指令和该数字中间不会出现运算指令和输出指令

重置指令后出现的第一个数字，表示基础值。且在重置指令和第一个数字中间不会出现运算指令和输出指令

进制转换指令可能出现在任何地方

运算过程中中间变量均为非负整数，且小于  $2^{63}263$ 。

### 输入格式

第 1 行：11 个  $n_i$ ，表示指令数量

第 2 ~  $n+12 \sim n+1$  行：每行给出一条指令。指令序列一定以 CLEAR 作为开始，并且满足指令规则

### 输出格式

依次给出每一次 EQUAL 得到的结果

### 样例输入

```
. 7
. CLEAR
. NUM 1024
. CHANGE 2
. ADD
. NUM 100000
. CHANGE 8
. EQUAL
```

### 样例输出

```
. 2040
```

这题涉及到进制转换和加减乘除取余这些运算，但遇到向 16 进制，有字母的就不好计算了。

那怎么办呢？

我们可以先把他全转成十进制，然后转成最后输入的进制，就非常的 Easy；

定义两个函数，一个输入转十进制，一个输出转最后输入的进制 k；

Like this:

```
string getRadixStr(ll decNum,int radix){
    string res;
    if(decNum==0) return "0";
    for(;decNum!=0;decNum/=radix)
        res.insert(res.begin(),decNum%radix>9?char(decNum%radix-10+'A'):char(decNum%radix+'0'));
    return res;
}

ll toDec(string str,int radix){
    ll res=0;
    for(int i=0;i<str.size();i++)
        res=res*radix+(isdigit(str[i])?str[i]-'0':str[i]-'A'+10);
    return res;
}
```

再来一个函数，这是用来运算的。算好的结果存到 val 里；

如果为 clear，就把 val 清零；

如果是 CHANGE，就输入 k；

如果是"EQUAL，就输出；

注意：

1. 如果一开始没输入 k，那就是将输入的数转成 0 进制（0 进制是什么鬼？！），所以要给 k 赋上初值 10；
2. 一定要把 val 附上 1，不然一直为 0；
3. 要设成全局变量；
4. 要开 long long，不然会错；
5. 考虑到要转进制，所以数字也要开 string