

定位

一个一次函数表达式求解和代入，加上一个询问，以及一个April Fools的特判——题目确实非常简单，数据很弱。

实现

这道题直接用常规模拟即可，因为数据比较友好，常规做法是不会TLE的。

题解

我们不妨单独用一个函数来判断一个数是否只且包括4和1。即：

```
bool check(long long s)
{
    bool f1=0, f2=0; //两个bool分别记录是否有4或1
    while(s) //不停地抹掉末位
    {
        if(s%10-4&& s%10-1) return 0; //只要不是4，也不是1就退出
        f1+=s%10==4;
        f2+=s%10==1; //累计
        s/=10; //抹位
    }
    return f1*f2; //返回
}
```

接下来是打印，其实现非常简单：

```
void print(long long s)
{
    if(check(s)) cout<<"April Fools!\n";
    else cout<<s<<endl;
}
```

接着是一次函数表达式。对于 $a[1] = 1, a[i] = j$ ，我们可以用待定系数法求解。假设：

$$y = kx + b$$

把 $x = 1, y = 1; x = i, y = j$ 代入，得：

$$\begin{cases} 1 = k + b & \textcircled{1} \\ j = ki + b & \textcircled{2} \end{cases}$$

② - ①得：

$$j - 1 = ki - k$$

$$k(i - 1) = j - 1$$

$$k = \frac{j - 1}{i - 1}$$

因此：

$$b = 1 - k$$

然后是模拟部分——如果是询问就输出（按照题意），否则改变数值。
即：

```
while(m--){
    cin>>c;
    if(c=='Q'){
        int l;
        cin>>l;
        if(a[l]-t)print(a[l]);
        else print(l*k+b);
    }
    else{
        int l,L;
        cin>>l>>L;
        a[l]=L;
    }
}
```

这样的话，程序其实就已经结束了，因为输出包含在模拟中。当然，数组的初始化（memset 为 0x80）也是非常重要的，这里就不再赘述。

总结

总之，这道题是非常水的，主要是因为数据比较友善，不会导致超时。后续可能会有加强版。