#### 位运算法★★

　　注意到时钟只有四种状态，12点，3点，6点，9点。令它们为0，1，2，3（这样的定义好处很大）。

　　则它们对应的二进制数为：000，001，010，011。即我们用三个位来记录一个时钟的状态。

要拨动一个时钟的时候，就给该位加上一，再和数“57521883”按位与（&）以去除高位的1。因为数“57521883”的二进制表示为(011 011 011 011 011 011 011 011 011)。

又如旋转时钟指针的第一种方法可以以数“18911232”来表示，因为该数的二进制表示为（001 001 000 001 001 000 000 000 000）。

由此设move[i]表示题述中的第i+1种方法，则const long move[9] = {18911232，19136512， 2363904，16810048， 2134536，262657，36936，73，4617}

　　令f[q]为原状态，比如用旋转时钟指针中的第k种方法，那么可以写成 f[q + 1] = (f[q] + move[k －1]) & 57521883;

当9个时钟都回归12点的时候，巧的是状态f=0。这样，判断每个状态f是否为0，就知道是否求出可行解。