括号序列计数

（brackets）

题目描述：

Alice 和 Bob 知道，一个由空格、左括号、右括号组成的序列被称为括号序列。有一类特殊的括号序列被称为“合法括号序列”。已经知道：

（1）“()”是合法的括号序列，空串是合法的括号序列。

（2）如果 S1 是合法的括号序列，S2 是合法的括号序列，则 S1 与 S2 拼接起来的序列 S1＋S2 也是合法的括号序列。

（3）如果 S 是合法的括号序列，在其左右分别插入一个左括号和一个右括号所得到的字符串“（”＋S1＋“）”也是合法的括号序列。

（4）如果 S 是合法的括号序列，在 S 的任何位置（包括头尾位置）插入一个空格，得到的序列也是合法的括号序列。

现在，Alice 希望知道：对于某个已知的有限状态自动机中的状态 s 与 t，存在多少以 s 为起点，t 为终点的长度为 k 的合法括号序列。

所谓有限状态自动机，又可以被认为是一个有向图 G，由 n 个结点组成，每一个结点表示一个状态，且存在三类以此为起点出去的有向边，对于每一个状态（或结点）来说其出去的同一类有向边将指向同样的状态（或结点）。三类有向边分别代表三种符号：左括号“（”，右括号“）”和空格。

这里，我们将状态（或结点）从 0 开始编号。对于第 i 个状态，用 dfa[i][0]，dfa[i][1]，dfa[i][2]分别表示从 i 出发，代表了左括号、右括号和空格的那一类边指向的状态（或结点），再用 dfa2[i][0]，dfa2[i][1]，dfa2[i][2]表示每一类边的个数。

对于一条从 s 出发到 t 结束的路径，满足长度为 k 且路径经过的边对应的符号组成了一组合法的括号匹配，则称作“满足[G,s,t,k]的合法括号序列”。

现在，Alice 为 Bob 提供了自动机 G，并提出 Q 组询问。对于每一组询问，Alice 会给出 s、t 和 k，她希望 Bob 可以告诉她满足[G,s,t,k]的合法括号序列有多少组。她只需要知道答案除以 47 后的余数。

输入文件：

第一行一个整数 n，表示状态数（或结点数）。

之后 n 行，对于其中的第 i 行，有 6 个 32 位整数，依次为 dfa[i-1][0]，dfa2[i-1][0]，dfa[i-1][1]，dfa2[i-1][1]，dfa[i-1][2]，dfa2[i-1][2]。

之后一行有一个整数 Q，表示 Alice 的询问次数。

之后 Q 行，每一行有 3 个 32 位的整数，依次为 s，t，k。

输出文件：

输出文件有 Q 行。

其中第 i 行对应了第 i 个询问，只有一个整数，表示满足[G,s,t,k]的合法括号序列的个数，答案只需要除以 47 后的余数。

样例输入：

1
0 1 0 2 0 3
6
0 0 3
0 0 4
0 0 5
0 0 6
0 0 7
0 0 8

样例输出：

45
9
10
2
19
25

样例说明：

对于第一组询问，长度为 3 的合法括号序列依次有：

（1）三个空格“　　”，则在自动机中的合法方案数为：3×3×3＝27。

（2）对于“（ ）”、“（）　”和“　（）”，则在自动机中的合法方案数为：1×3×2＝6。

所以总的可行方案数为：27＋3×6＝27＋18＝45。

数据规模：

存在 10％的数据：k <= 1000。时间限制 1 秒。

另存在 10％的数据：

n=1，dfa[0][0] = dfa[0][1] = dfa[0][2] = dfa2[0][2]=0，dfa2[0][0] = dfa2[0][1] = 1。

时间限制 5 秒。

另存在 20％的数据：

n=1，dfa[0][0] = dfa[0][1] = dfa[0][2] = 0。

时间限制 5 秒。

另存在 30％的数据：k <= 30000。时间限制 15 秒。

另存在 30％的数据：k <= 100000。时间限制 20 秒。

对于 100％的数据，n <= 2，k <= 100000，Q <= 10。