

250711快速计算（题解）

题目链接：

<https://www.matiji.net/exam/brushquestion/52/4009/C448715ED43BEA9D2D47CED523050945?from=1>

小码哥接到了实验室大量的计算任务。

每次任务开始的时候，小码哥会收到一个变量 x 和一个数字 p ，他需要循环执行给定的 n 条语句 p 轮，即一共执行 $n \times p$ 个语句，然后输出变化后的 x 值。

语句有两种格式，将以以下格式给出：

ADD a : 表示 $x = (x + a) \bmod 10007$

MUL a : 表示 $x = ax \bmod 10007$

输入格式：

输入的第一行为一个整数 n , ($1 \leq n \leq 100000$)

接下来 n 行，每行为一条语句，格式见题意。（其中 $1 \leq a \leq 100$ ）

紧接着一行一个整数 q ($1 \leq q \leq 100000$)，表示任务数量。

最后 q 行，每行两个数字 x ($1 \leq x \leq 100$) 与 p ($1 \leq p \leq 100000$)，含义见题意。

输出格式：

输出 q 行，每行一个整数表示对应计算任务完成后 x 的值。

输入：

2

ADD 1

MUL 2

1

1 2

输出：

10

样例解释：

一开始 $x=1$ ，将执行给定的2条语句两轮。

第一轮： $x = (1+1) \bmod 10007 = 2$, $x = (2 \times 2) \bmod 10007 = 4$

第二轮： $x = (4+1) \bmod 10007 = 5$, $x = (5 \times 2) \bmod 10007 = 10$

故输出10。

解法一：

```
#include <bits/stdc++.h>
using namespace std;
const int MOD = 10007;
```

```

int main() {
    int n;
    cin >> n;
    // 读取操作类型和数值
    char c[n][3]; // 操作类型 (A或M)
    int a[n];      // 操作数值
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < 3; j++) {
            cin >> c[i][j]; // 读取操作
        }
        cin >> a[i]; // 读取操作值
    }
    int x = 1, y = 0;
    // 处理操作: A -> y = (y + a[i]) % 10007, M -> x = (x * a[i]) % 10007, y
    = (y * a[i]) % 10007
    for (int i = 0; i < n; i++) {
        if (c[i][0] == 'A') {
            y = (y + a[i]) % MOD;
        } else if (c[i][0] == 'M') {
            x = (x * a[i]) % MOD;
            y = (y * a[i]) % MOD;
        }
    }
    // 处理查询
    int q;
    cin >> q;
    int xx[n], yy[n];
    xx[0] = x;
    yy[0] = y;

    int num = 1; // 查询前的初始步数
    for (int i = 0; i < q; i++) {
        int f, p;
        cin >> f >> p;
        if (num < p) {
            // 更新 xx 和 yy 的值, 以支持后续查询
            for (int j = num; j < p; j++) {
                xx[j] = (xx[j - 1] * x) % MOD; // 更新 x
                yy[j] = (yy[j - 1] * x + y) % MOD; // 更新 y
            }
            num = p; // 更新步数
        }
        // 查询结果: xx[p-1] * f + yy[p-1]
        cout << (xx[p - 1] * f + yy[p - 1]) % MOD << endl;
    }
    return 0;
}

```