

函数调用 (call)

【题目描述】

函数是各种编程语言中一项重要的概念，借助函数，我们总可以将复杂的任务分解成一个个相对简单的子任务，直到细化为十分简单的基础操作，从而使代码的组织更加严密、更加有条理。然而，过多的函数调用也会导致额外的开销，影响程序的运行效率。

某数据库应用程序提供了若干函数用以维护数据。已知这些函数的功能可分为三类：

1. 将数据中的指定元素加上一个值；
2. 将数据中的每一个元素乘以一个相同值；
3. **依次**执行若干次函数调用，保证不会出现递归（即不会直接或间接地调用本身）。

在使用该数据库应用时，用户可一次性输入要调用的函数序列（一个函数可能被调用多次），在**依次**执行完序列中的函数后，系统中的数据被加以更新。某一天，小 A 在应用该数据库程序处理数据时遇到了困难：由于频繁而低效的函数调用，系统在执行操作时进入了无响应的状态，他只好强制结束了数据库程序。为了计算出正确数据，小 A 查阅了软件的文档，了解到每个函数的具体功能信息，现在他想请你根据这些信息帮他计算出更新后的数据应该是多少。

【输入格式】

输入文件名为 `call.in`。

第 1 行一个正整数 n ，表示数据的个数。

第 2 行 n 个整数，第 i 个整数表示下标为 i 的数据的初始值为 a_i 。

第 3 行一个正整数 m ，表示数据库应用程序提供的函数个数。函数从 1 ~ m 编号。

接下来 m 行中，第 j ($1 \leq j \leq m$) 行的第一个整数为 T_j ，表示 j 号函数的类型：

1. 若 $T_j = 1$ ，接下来两个整数 P_j, V_j 分别表示要执行加法的元素的下标及其增加的值；
2. 若 $T_j = 2$ ，接下来一个整数 V_j 表示所有元素所乘的值；
3. 若 $T_j = 3$ ，接下来一个正整数 C_j 表示 j 号函数要调用的函数个数，

随后 C_j 个整数 $g_1^{(j)}, g_2^{(j)}, \dots, g_{C_j}^{(j)}$ 依次表示其所调用的函数的编号。

第 $m + 4$ 行一个正整数 Q ，表示输入的函数操作序列长度。

第 $m + 5$ 行 Q 个整数 f_i ，第 i 个整数表示第 i 个执行的函数的编号。

【输出格式】

输出文件名为 `call.out`。

一行 n 个用空格隔开的整数，按照下标 $1 \sim n$ 的顺序，分别输出在执行完输入的函数序列后，数据库中每一个元素的值。答案对 998244353 取模。

【样例 1 输入】

```
3
1 2 3
3
1 1 1
2 2
3 2 1 2
2
2 3
```

【样例 1 输出】

```
6 8 12
```

【样例 1 解释】

1 号函数功能为将 a_1 的值加一。2 号函数功能为所有元素乘 2。3 号函数将先调用 1 号函数，再调用 2 号函数。

最终的函数序列先执行 2 号函数，所有元素的值变为 2,4,6。

再执行 3 号函数时，先调用 1 号函数，所有元素的值变为 3,4,6。再调用 2 号函数，所有元素的值变为 6,8,12。

【样例 2 输入】

```
10
1 2 3 4 5 6 7 8 9 10
8
3 2 2 3
3 2 4 5
3 2 5 8
2 2
3 2 6 7
1 2 5
1 7 6
2 3
3
1 2 3
```

【样例 2 输出】

36 282 108 144 180 216 504 288 324 360

【样例 3】

见选手目录下的 call/call3.in 与 call/call3.ans。

【数据范围与提示】

测试点编号	$n, m, Q \leq$	$\sum C_j$	其他特殊限制
1 ~ 2	1000	$= m - 1$	函数调用关系构成一棵树
3 ~ 4		≤ 100	无
5 ~ 6	20000	≤ 40000	不含第 2 类函数或不含第 1 类函数
7		$= 0$	无
8 ~ 9		$= m - 1$	函数调用关系构成一棵树
10 ~ 11		$\leq 2 \times 10^5$	无
12 ~ 13	10^5	$\leq 2 \times 10^5$	不含第 2 类函数或不含第 1 类函数
14		$= 0$	无
15 ~ 16		$= m - 1$	函数调用关系构成一棵树
17 ~ 18		$\leq 5 \times 10^5$	无
19 ~ 20	$\leq 10^6$		

对于所有数据： $0 \leq a_i \leq 10^4$ ， $T_j \in \{1,2,3\}$ ， $1 \leq P_j \leq n$ ， $0 \leq V_j \leq 10^4$ ，

$1 \leq g_k^{(j)} \leq m$ ， $1 \leq f_i \leq m$ 。