A. 硬币（coins）

这是一道交互题。

时间限制：3 秒，空间限制：2 GiB

有 n 堆硬币，第 i（$0 \le i < n$）堆硬币有 a_i 枚。每一枚硬币的质量是 5 克，但有一堆当中的所有硬币都是假币，假币的质量是每一枚 6 克。除了这一堆外的所有硬币都是真币。

你有一个精确的称重仪器，在它上面放置任意枚硬币之后，它会返回这些硬币的质量之和。

你需要找到假币是哪一堆，且称重的次数尽可能少。

实现要求

你的程序应当包含如下的头文件：

- ```
  #include "coins.h"
  ```

你的程序不应包含 `main` 函数，而应当实现以下函数：

- ```
  int solve(std::vector<int> a);
  ```

  该函数接受大小为 n 的数组 a，含义与题面中相同。该函数应当返回一个 0 到 $n-1$ 中的整数，表示假币堆的编号。

你的程序可以调用以下函数：

- ```
  long long weigh(std::vector<int> p);
  ```

  该函数接受大小为 n 的数组 p，其中 $0 \le p_i \le a_i$，表示将第 i 堆中的 p_i 枚硬币放到称重仪器上。该函数返回所有放到仪器上的硬币的质量之和。称重完成后，所有硬币将会回到它本来所在的堆。

你的程序不应进行任何输入和输出操作。但是，作为特例，你可以向标准错误流（`stderr`）输出信息，但是注意这也会计算进你的运行时间。

样例

假设 $a = [1, 2]$，假币堆的编号为 0。评分程序将会调用 `solve([1, 2])`。

示例解决方案将会调用 `weigh([1, 0])`，得到返回值 6。

示例解决方案找到了假币堆，于是返回其编号 0。

评分方式

在每个测试点中，`solve` 函数将会被调用恰好一次。如果你的程序没有正确运行，或是没有返回正确的答案，则得分为 0。

如果你的程序正确运行，且返回了正确的答案，令 W 是如果使用最优策略，在最坏情况下确定假币堆需要的称重次数的最小值。令 C 为你的程序调用 `weigh` 函数的次数。则你在该测试点的得分为：
  ```

C	得分
$\leq W$	100%
$= W + 1$	50%
$= W + 2$	25%
$\geq W + 3$	0%

你在一个子任务的得分是该子任务中每个测试点得分的最小值。

在正式的评分程序中，如果你调用 `weigh` 函数的次数不超过 $W + 2$，则保证评分程序占用的时间不超过 1 秒，占用的空间不超过 256MiB。

数据约束

- $1 \leq n \leq 10^6$
- $1 \leq a_i \leq 10^9$（$0 \leq i < n$）

子任务的列表如下：

子任务	额外约束	分数
1	$a_i = 1$（$0 \leq i < n$）	8
2	至多一次称重就能确定假币堆，即 $W \leq 1$	8
3	$n \leq 1000$，且至多两次称重就能确定假币堆，即 $W \leq 2$	28
4	所有 a_i 相等	12
5	$n \leq 10^3$	32
6	无	12

测试

下发文件中包含了如下内容：

- `coins.h`：你的程序需要包含的头文件。
- `coins.cpp`：本题解决方案的示例代码。
- `grader.cpp`：示例评分程序。
- `coins1~3.in`：样例输入。

你可以使用以下的命令编译示例评分程序（其中 `coins.cpp` 是你的解决方案）：

```
g++ -std=c++14 -O2 -o grader coins.cpp grader.cpp
```

编译好的评分程序将会从标准输入以如下的格式读入数据：

- 第一行两个整数 n, k，表示硬币的堆数和假币堆的编号。
- 第二行 n 个整数 $a_0, a_1, \ldots, a_{n-1}$，表示每一堆硬币的枚数。

如果你的程序执行了不合法的操作，或是得到了错误的答案，评分程序会返回如下的错误：

- `Wrong answer [1]`：p 的大小不为 n。

- `Wrong answer [2]`：p_i 的范围不在 0 到 a_i 中。
- `Wrong answer [3]`：`solve` 的返回值不在 0 到 $n-1$ 中。
- `Wrong answer [4]`：`solve` 返回了错误的硬币堆。

如果你的程序正确运行并返回了正确答案，则评分程序会告知你调用 `weigh` 函数的次数。

注意！示例评分程序和正式评分使用的评分程序不同：

- **示例程序并不会检查你的称重次数是否是最少的。**
- **正式评分程序是适应性的，即假币堆并不是事先确定的，而是可以在交互过程中随时改变，只要与之前的所有称重结果一致。**

B．局部最大值（local）

时间限制：3 秒，空间限制：2 GiB

对于一个非空数组 A，定义 $f(A)$ 如下：

- 重复以下操作直到 A 只包含一个数：将 A 替换为 A 的所有**局部最大值**保持相对顺序形成的数组。其中 A_i 是局部最大值当且仅当下列条件均满足：
  - $i=1$ 或 $A_i > A_{i-1}$；
  - $i=|A|$ 或 $A_i > A_{i+1}$。
- $f(A)$ 是上述操作的重复次数。

例如，对于数组 $[1,5,2,3,4]$，第一次操作后得到 $[5,4]$，第二次操作后得到 $[5]$，因此 $f([1,5,2,3,4])=2$。

给定 N,K,P，求有多少大小为 N 的排列 P 满足 $f(P)=K$，答案对 P 取模。

输入格式

一行三个整数 N,K,P。

输出格式

输出一行一个整数，表示答案。

样例

样例 1 输入

```
5 3 100000007
```

样例 1 输出

```
4
```

样例 1 解释

满足条件的序列是：

- $[4,1,3,2,5]$，
- $[4,2,3,1,5]$，
- $[5,1,3,2,4]$，
- $[5,2,3,1,4]$。

样例 2~5

见下发文件 `local2~5.in/out`。

数据约束

- $2 \le N \le 5\,000$
- $1 \le K \le N-1$
- $10^8 \le P \le 10^9$

子任务的列表如下：

子任务	额外约束	分数
1	$N \le 10$	10
2	$N \le 30$	20
3	$N \le 1\,000$	50
4	无	20

C. 并行程序（program）

时间限制：9 秒，空间限制：2 GiB

有 n 个程序。所有程序共享一个全局整数型变量 x，此外，每个程序都有一个私有的计数器 y。每个程序都由一连串的指令组成，每个指令都属于以下四种类型之一：

- `W`：将全局变量的值 x 载入私有计数器 y。
- `Z`：将私有计数器 y 的值写入全局变量 x。
- `+` c：将 y 的值加一正常数 c。
- `-` c：将 y 的值减一正常数 c。

这些程序将会并行运行。所有计数器 y 和变量 x 的初始值都是 0。这些程序的指令**交错**执行，即所有程序的所有指令都是一个接一个地执行，对于每个时刻，每个程序满足它的指令的一个前缀以一定顺序被执行。

计算所有程序并行执行后变量 x 的最小可能值是多少。

输入格式

第一行一个整数 t，表示该测试点的数据组数。

每组数据第一行一个整数 n，表示程序的数量。每组数据的第 2 到 $2n+1$ 行，每两行表示一个程序。

每个程序的第一行一个整数 l_i，表示该程序的指令数量。每个程序的第二行是 l_i 个空格分隔的指令，每个指令的格式是如下四种之一：

- 一个字符 `W`：表示载入指令；
- 一个字符 `Z`：表示写入指令；
- 空格分隔的一个字符 `+` 和一个数字 c_{ij}：表示给私有计数器加 c_{ij}；
- 空格分隔的一个字符 `-` 和一个数字 c_{ij}：表示给私有计数器减 c_{ij}。

输出格式

每组数据输出一行一个整数，表示答案。

样例

样例 1 输入

```
2
2
12
w + 2 z w + 2 z w + 2 z w + 2 z
12
w + 3 z w + 3 z w + 3 z w + 3 z
3
3
w w - 5
5
+ 9 z + 1 z w
8
+ 10 z - 2 z - 5 w - 1 z
```

样例 1 输出

```
5
7
```

样例 1 解释

对于第一组数据，得到最小的 x 程序指令执行顺序如下表。

			w				+ 2					z	w		+ 2	z	w	+ 2		z	w	+ 2	z	
	w	+ 3		z	w	+ 3		z	w	+ 3	z			w					+ 3					z
y_1	0	0	**0**	0	0	0	**2**	2	2	2	2	2	**2**	2	**4**	4	**4**	**6**	6	6	**6**	**8**	8	8
y_2	**0**	**3**	3	3	**3**	**6**	6	6	**6**	**9**	9	9	9	**2**	2	2	2	2	**5**	5	5	5	5	5
x	0	0	0	**3**	3	3	3	**6**	6	6	**9**	**2**	2	2	2	**4**	4	4	4	**6**	6	6	**8**	**5**

该样例满足子任务的约束。

样例 2-4

见下发文件。

数据约束

- $1 \le t \le 10^6$
- $1 \le n \le 10^6$
- $1 \le l_i \le 10^6$
- $1 \le c_{ij} \le 10^9$
- 一个测试点中的所有数据的 $\sum l_i$ 之和不超过 10^6

子任务的列表如下：

子任务	额外约束	分数

子任务	额外约束	分数
1	$t \le 10$，每组数据中 $\sum l_i \le 10$	3
2	$n = 2$，每个测试点中的所有数据的 $\sum l_i$ 之和不超过 10^4	10
3	$n = 2$	15
4	$n \le 50$，每个测试点中的所有数据的 $\sum l_i$ 之和不超过 10^4	13
5	每个测试点中的所有数据的 $\sum l_i$ 之和不超过 10^4	21
6	$l_i \le 10$	22
7	无	16