

分手是祝愿 (trennen)

【问题描述】

Zeit und Raum trennen dich und mich.

时空将你我分开。

B 君在玩一个游戏，这个游戏由 n 个灯和 n 个开关组成，给定这 n 个灯的初始状态，下标为从 1 到 n 的正整数。

每个灯有两个状态亮和灭，我们用 1 来表示这个灯是亮的，用 0 表示这个灯是灭的，游戏的目标是使所有灯都灭掉。

但是当操作第 i 个开关时，所有编号为 i 的约数（包括 1 和 i ）的灯的状态都会被改变，即从亮变成灭，或者从灭变成亮。

B 君发现这个游戏很难，于是想到了这样的一个策略，每次等概率随机操作一个开关，直到所有灯都灭掉。

这个策略需要的操作次数很多，B 君想到这样的一个优化。如果当前局面，可以通过操作小于等于 k 个开关使所有灯都灭掉，那么他将不再随机，直接选择操作次数最小的操作方法（这个策略显然小于等于 k 步）操作这些开关。

B 君想知道按照这个策略（也就是先随机操作，最后小于等于 k 步，使用操作次数最小的操作方法）的操作次数的期望。

这个期望可能很大，但是 B 君发现这个期望乘以 n 的阶乘一定是整数，所以他只需要知道这个整数对 100003 取模之后的结果。

【输入格式】

从文件 *trennen.in* 中读入数据。

第一行两个整数 n, k 。

接下来一行 n 个整数，每个整数是 0 或者 1，其中第 i 个整数表示第 i 个灯的初始情况。

【输出格式】

输出到文件 *trennen.out* 中。

输出一行，为操作次数的期望乘以 n 的阶乘对 100003 取模之后的结果。

【样例 1 输入】

```
4 0
0 0 1 1
```

【样例 1 输出】

512

【样例 2 输入】

```
5 0
1 0 1 1 1
```

【样例 2 输出】

5120

【样例 3~8】

见选手目录下的 *trennen/trennen3~8.in* 与 *trennen/trennen3~8.ans*。

【子任务】

- 对于 0% 的测试点，和样例一模一样；
- 对于另外 30% 的测试点， $n \leq 10$ ；
- 对于另外 20% 的测试点， $n \leq 100$ ；
- 对于另外 30% 的测试点， $n \leq 1000$ ；
- 对于 100% 的测试点， $1 \leq n \leq 100000, 0 \leq k \leq n$ ；
- 对于以上每部分测试点，均有一半的数据满足 $k = n$ 。