

Problem 1. treasure

Input file: **treasure.in**
Output file: **treasure.out**
Time limit: 1 second
Memory limit: 256 MB

最近 Mr. Hu 得到了一个宝箱, 但宝箱被上了锁, 需要解决一个问题才能将其打开, 问题是这样的:

你有 n 个不同的苹果, 你想从里面选出 m 个来, 问方案数, 结果可能很大, 输出模 M 后的结果输出, 其中 M 是 k 个不同的素数 p_1, p_2, \dots, p_k 的乘积.

Mr. Hu 一眼就解决了这个问题, 但他知道你们最近才学了这方面的知识, 于是就把这个问题交给了你们.

Input

第 1 行, 3 个整数 $n \ m \ k$, 意义如上。

第 2 行 k 个不同的素数: $p_1 \ p_2 \ \dots \ p_k$.

Output

输出一个整数表示答案.

Sample

treasure.in	treasure.out
9 5 2 3 5	6

Note

- 对于 10% 的数据, $1 \leq n \leq 10^3, k = 1$;
- 对于 30% 的数据, $1 \leq n \leq 10^5, k = 1, n < p_1$;
- 对于 60% 的数据, $1 \leq n \leq 10^{18}, k = 1$;
- 对于 100% 的数据, $1 \leq n \leq 10^{18}, 1 \leq k \leq 10$
- 对于所有数据, $0 \leq m \leq n \leq 10^{18}, 2 \leq M \leq 10^{18}, 2 \leq p_i \leq 10^5$ 且 p_i 互不相同.

Problem 2. kand

Input file: `kand.in`
Output file: `kand.out`
Time limit: 1 second
Memory limit: 256 MB

Mr. Hu 创造了一堆数, $a_1, a_2, a_3, \dots, a_n$, 并从中选出 k 个来, 把它们按位与起来, 这样就得到了一个数 r , Mr. Hu 非常喜欢 s , 所以希望你们能计算出有多少种选法, 使得 $r = s$ 。

Input

第一行 3 个数: n k s 。第二行 n 个整数: a_1 a_2 a_3 ... a_n 表示给定序列。

Output

输出有多少种选法, 答案对 $10^9 + 7$ 取模。

Sample

<code>kand.in</code>	<code>kand.out</code>
3 2 2 1 2 3	1

选 2 个数出来总共有 3 中方案, 分别是: (1, 2), (1, 3), (2, 3), 其按位与分别是 0, 1, 2, 所以只有一个符合要求, 输出 1。

Note

- 对于 10% 的数据, $1 \leq n \leq 10$, $0 \leq a_i < 2^{10}$;
- 对于 30% 的数据, $1 \leq n \leq 100$, $0 \leq a_i < 2^{10}$;
- 对于 70% 的数据, $1 \leq n \leq 10^5$, $0 \leq a_i < 2^{15}$;
- 对于 100% 的数据, $1 \leq n \leq 10^5$, $0 \leq a_i < 2^{20}$, $1 \leq k \leq n$ 。

Problem 3. solar

Input file: solar.in
Output file: solar.out
Time limit: 1 second
Memory limit: 256 MB

Mr. Hu 非常喜欢遨游太空，所以经常看见星球碰撞，这天，他观察到了周围星球的位置和运动速度，想让你推演一下接下来的演化。这片空间是三维的，其中坐标 (x, y, z) 的范围为 $[0, nx) \times [0, ny) \times [0, nz)$ ，其中有 n 个星球，每个星球都有个质量 m_i 。

现在告诉你这 n 个星球的坐标，其中第 i 个星球坐标为 (x_i, y_i, z_i) (保证 $0 \leq x_i < nx, 0 \leq y_i < ny, 0 \leq z_i < nz$)，运动的速度为 (vx_i, vy_i, vz_i) 。

如果一个星球当前坐标为 (x, y, z) ，速度为 (vx, vy, vz) ，那么下一秒它会到达

$$(x + vx \bmod nx, y + vy \bmod ny, z + vz \bmod nz)$$

如果两个或多个星球在某个整数秒的时刻运动到了同一个位置，则发生了一次碰撞，这些星球会融合成一个质量更大的星球，坐标不变，速度是所有星球速度和的平均值（如果某一维不是整数，则将该维分量向 0 取整），质量变为所有星球的质量总和。

现在询问你在最后一次碰撞发生后（融合完成后），每个星球质量，坐标及速度。

Input

第 1 行 4 个整数，表示 $n \ nx \ ny \ nz$ ，即最开始的星球数及各维坐标的范围。

接下来 n 行，每行 7 个数： $m_i \ x_i \ y_i \ z_i \ vx_i \ vy_i \ vz_i$ ，给出一个星球的信息。

Output

第 1 行 1 个整数 r ，表示最后一次碰撞发生后，还剩下的星球数。

接下来 r 行，每行输出 $m_i \ x_i \ y_i \ z_i \ vx_i \ vy_i \ vz_i$ ，表示一个星球的信息。输出先按质量从大到小排序，之后再按位置坐标字典序从小到大排序。

Sample

solar.in	solar.out
2 8 8 8 12 4 1 4 5 3 -2 10 1 2 1 8 -6 1	1 22 1 4 2 6 -1 0

样例解释：两个星球会在第 1 秒相撞，平均速度为 $(\frac{13}{2}, -\frac{3}{2}, -\frac{1}{2})$ ，向 0 取整后为 $(6, -1, 0)$ 。

solar.in	solar.out
2 10 20 30 10 1 0 0 2 0 0 15 2 0 0 4 0 0	2 15 2 0 0 4 0 0 10 1 0 0 2 0 0

样例解释：两个星球永远不会撞，直接输出第 0 秒的情况。

Note

- 对于 10% 的数据， $n = 2, 1 \leq nx, ny, nz \leq 10$ 。
- 对于 30% 的数据， $1 \leq n \leq 5, 1 \leq nx, ny, nz \leq 10$ 。
- 对于另外 40% 的数据， $1 \leq n \leq 100, 1 \leq nx, ny, nz \leq 10^3$ ，保证 nx, ny, nz 都是质数。
- 对于 100% 的数据， $1 \leq n \leq 100, 1 \leq nx, ny, nz \leq 10^3, 0 \leq x_i < nx, 0 \leq y_i < ny, 0 \leq z_i < nz, -nx < vx_i < nx, -ny < vy_i < ny, -nz < vz_i < nz, 1 \leq m_i \leq 100$ 。