Hash Cat

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Source: `https://hashcat.net`

It is the Year 202020. The universe was not as peaceful as the meows had hoped. With the technology to wipe out planets readily available to all inter-galactic species, everyone has gone into hiding throughout the universe. Any civilization foolish enough to show their location is almost instantly eliminated in unspectacular fashion. All that remains of meow-kind resides on a planet in Star system 7355608.

One day, the meows receive a signal from outside their star system. The Scientist Meows manage to intercept the signal. However, they are unable to directly make sense of the signal as the message is a hash. The meows wish to know what the message is about. Having been the species that pioneered hashing algorithms during peacetime, the meows know all the hashing algorithms other species are likely to use in communications. They deduce that the following hashing algorithm was used:

1. Split the message into packets of 4 ASCII characters. Concatenate their bytes to form a 4-byte integer, m.
2. Generate N pairs of integers (a_i, b_i) and multiply them with m to generate N integers $a_i b_i m$.
3. Sum all N integers and modulo the result by an 8 decimal digit prime number P to form E.

$$E = (\sum_{i=1}^{N} a_i b_i m) mod P$$

Assuming that the messages contains only numbers and alphabetical characters, the meows would like to know what are the possible messages that could have generated the hash packets they received.

They received Q packets, each containing an integer e_i denoting the i_{th} hash.

It is guaranteed there is at least one valid message that can hash to any given e_i.

Input

The first line of input contains two integers, $N(1 \leq N \leq 1000), P(10000019 \leq P \leq 99999989)$ — the number of pairs in the key and the big prime.

The second line of input consists of N integers $a_1, a_2, ..., a_N(1 \leq a_i \leq 10^6)$ — the first half of the pairs.

The third line of input consists of N integers $b_1, b_2, ..., b_N(1 \leq b_i \leq 10^6)$ — the second half of the pairs.

The fourth line of input contains a single integer, $Q(1 \leq Q \leq 10^5)$ — the number of packets.

The fifth line of input consists of Q integers, $e_1, e_2, ..., e_N(0 \leq e_i \leq 2^{32} - 1)$ — the hashed payloads.

Output

Output Q lines where the i^{th} line contains the messages that could have generated the i^{th} hash.

If there are multiple, output them in ascending order separated by space.

Example

standard input	standard output
10 71580473 3 2 10 4 2 3 10 2 10 3 5 2 1 4 5 10 7 6 10 9 2 48444741 15678009	Q4Xl bEEP sV24 1TXl BeEP Sv24

Note

Hexadecimal	*Decimal*	*Message*
5134586*c*	1362385004	*Q*4*Xl*
62454550	1648706896	*bEEP*
73563234	1935028788	*sV*24

These 3 values hash to 48444741.

Hexadecimal	*Decimal*	*Message*
3154586*c*	827611244	1*TXl*
42654550	1113933136	*BeEP*
53763234	1400255028	*Sv*24

These 3 values hash to 15678009.