Task: POD
Divisibility

XXIV OI, Stage I. Source file `pod.*` Available memory: 128 MB. *17.10–14.11.2016*

In the last informatics class, Bytie learned about *positional numeral systems.* In particular, that most of the time people use the decimal system they find natural whereas computers store numbers in binary system. But also, that any integer B larger than 1 can be the base of a positional numeral system. In such system, the digits are $0, 1, 2, \ldots, B-2, B-1$, and the k-digit string $c_{k-1}c_{k-2}\ldots c_2c_1c_0$ denotes the number

$$c_{k-1} \cdot B^{k-1} + c_{k-2} \cdot B^{k-2} + \ldots + c_2 \cdot B^2 + c_1 \cdot B + c_0.$$

For example, in ternary (i.e., base-3) positional system, 201 denotes the number $2 \cdot 3^2 + 0 \cdot 3 + 1$, which is 19 in the decimal system, which can written in short as $201_3 = 19_{10}$.

Bytie chose a particular number B as the base of the numerical system and wrote all B digits of this system on small pieces of paper, some possibly multiple times: for $i = 0, 1, \ldots, B-1$, there are a_i sheets of paper with the digit i. Using these sheets (digits), Bytie would like to compose the largest possible integer divisible by $B-1$. Write a program that will aid him in this task. The number he seeks could be very large, but only some of its digits will suffice for Bytie. We assume that the representation of a positive integer may not have any leading zeros, and the only valid representation of zero is 0.

Caution: For systems with bases larger than 10, we assume that the digits in a number's representation are separated, e.g., with spaces. This allows a uniquely decodable representation with digits written in decimal.

Input

In the first line of the standard input, there are two integers B and q ($B \geq 2$, $q \geq 1$), separated by a single space, which specify the base of the positional numeric system and the number of queries about the digits of the number sought by Bytie respectively.

The second line contains a sequence of B integers $a_0, a_1, \ldots, a_{B-1}$ ($a_i \geq 1$), separated by single spaces, which specify the numbers of sheets with successive digits that Bytie prepared.

Then, the next q lines that follow contain the queries: the i-th such line holds one integer k_i ($0 \leq k_i \leq 10^{18}$).

Output

Exactly q lines should be written to the standard output: the i-th of those is to contain the k_i-th base-B digit of the largest integer divisible by $B-1$ that Bytie may spell out with his sheets. The digits are numbered in the aforementioned way: from right to left (i.e., starting from the least significant digits) beginning with digit 0. If the number sought has less than k_i digits, then -1 should be printed in the i-th line.

Example

For the input data:

```
3 3
1 1 1
0
1
2
```

the correct result is:

```
0
2
-1
```

Explanation for the example: Having one of each of the digits 0, 1, and 2 in the ternary system, Bytie may compose $0_3 = 0_{10}$, $1_3 = 1_{10}$, $2_3 = 2_{10}$, $10_3 = 3_{10}$, $12_3 = 5_{10}$, $20_3 = 6_{10}$, $21_3 = 7_{10}$, $102_3 = 11_{10}$, $120_3 = 15_{10}$, $201_3 = 19_{10}$, and $210_3 = 21_{10}$. Only 0_3, 2_3, and 20_3 are divisible by 2, so the number he seeks is 20_3.

Sample grading tests:

`1ocen`: $B = 10$, $a_i = 1$; $q = 10$, $k_i = i - 1$;

`2ocen`: $B = 2$, $a_0 = 10\,000$, $a_1 = 1$; $q = 10\,001$, $k_i = i - 1$;

`3ocen`: $B = 1\,000\,000$, $a_i = 1$; $q = 1$, $k_1 = 999\,999$.

Grading

The set of tests consists of the following subsets. Within each subset, there may be several test groups.

Subset	**Property**	**Score**
1	$B, a_i, q \leq 100$	30
2	$B, a_i \leq 100$, $q \leq 100\,000$	25
3	$B \leq 1000$, $a_i \leq 1\,000\,000$, $q \leq 1000$	25
4	$B, a_i \leq 1\,000\,000$, $q \leq 100\,000$	20