Binary search on arrays 2

In the lecture, we learned binary search, and we want you to implement the binary search. It is a very efficient algorithm, and it will not only used on arrays..

Anyway, in this problem, you are given an **SORTED** integer array $a[1..n]$ and m integers $x_1, x_2, \cdots, x_m$ that we need to search from this array. The elements on the array are distinct. For each x_j, please find whether x_j exists in the array a or not, and if exists, please find the position of x_j in a (i.e. find k such that $a[k] = x_j$ hold.)

To reject solutions that read all $x_1, x_2, \cdots, x_m$ and compute answers at once, we decided to give inputs in a different way. Please refer to the input format and notes for further information.

Input

Your input consists of an arbitrary number of records, but no more than 5.

Each record starts with a line containing two integers n and m ($1 \le n, m \le 100{,}000$), the number of integers. The next line contains n distinct integers $a[1], a[2], \cdots, a[n]$ ($0 \le a[1] < a[2] < \cdots < a[n] < 10^9$), which are the elements of the array. The next line contains $y_1, y_2, \cdots, y_m$ ($0 \le y_j < 10^9$), which are used to generate the sequence $x_1, x_2, \cdots, x_m$. You should generate like this:

- $x_1 = y_1$.
- For all $j \ge 2$, $x_j = (x_{j-1} + y_j) \bmod 10^9$ if x_{j-1} exists in array a, and $x_j = (x_{j-1} - y_j) \bmod 10^9$ otherwise.

Be aware that if you apply the modulo operation of C++ on negative numbers, the result becomes negative. For example, (5 – 12) % 1000000000 == -7. So try to avoid that situation. (We've discussed this before)

The end of input is indicated by a line containing only the value -1.

Output

For each input record, print a line that contains m integers, each separated by a space. The j-th ($1 \le j \le m$) integer should be the position of x_j in array a if x_j exists in array a, and 0 otherwise.

Example

Standard input	Standard output
3 6 1 2 3 1 1 1 1 1 1 3 5 1 3 5 2 1 2 999999999 999999997 -1	1 2 3 0 3 0 0 1 2 0 3

Notes

For the first sample:

- $x_1 = y_1 = 1$. $a[1] = 1$.
- $x_2 = x_1 + y_2 = (1 + 1) \bmod 10^9 = 2$. $a[2] = 2$.
- $x_3 = x_2 + y_3 = (2 + 1) \bmod 10^9 = 3$. $a[3] = 3$.
- $x_4 = x_3 + y_4 = (3 + 1) \bmod 10^9 = 4$. 4 doesn't exist in array a.
- $x_5 = x_4 - y_5 = (4 - 1) \bmod 10^9 = 3$. $a[3] = 3$.
- $x_6 = x_5 + y_6 = (3 + 1) \bmod 10^9 = 4$. 4 doesn't exist in array a.

For the second example:

- $x_1 = y_1 = 2$. 2 doesn't exist in array a.
- $x_2 = x_1 + y_2 = (2 - 1) \bmod 10^9 = 1$. $a[1] = 1$.
- $x_3 = x_2 + y_3 = (1 + 2) \bmod 10^9 = 3$. $a[2] = 3$.
- $x_4 = x_3 + y_4 = (3 + 999999999) \bmod 10^9 = 2$. 2 doesn't exist in array a.
- $x_5 = x_4 - y_5 = (2 - 999999997) \bmod 10^9 = 5$. $a[3] = 5$.

Time Limit

2 seconds.