

E. The Art of Dealing with ATM

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

ATMs of a well-known bank of a small country are arranged so that they can not give any amount of money requested by the user. Due to the limited size of the bill dispenser (the device that is directly giving money from an ATM) and some peculiarities of the ATM structure, you can get at most k bills from it, and the bills may be of at most two distinct denominations.

For example, if a country uses bills with denominations 10, 50, 100, 500, 1000 and 5000 burles, then at $k = 20$ such ATM can give sums 100 000 burles and 96 000 burles, but it cannot give sums 99 000 and 101 000 burles.

Let's suppose that the country uses bills of n distinct denominations, and the ATM that you are using has an unlimited number of bills of each type. You know that during the day you will need to withdraw a certain amount of cash q times. You know that when the ATM has multiple ways to give money, it chooses the one which requires the minimum number of bills, or displays an error message if it cannot be done. Determine the result of each of the q of requests for cash withdrawal.

Input

The first line contains two integers n, k ($1 \leq n \leq 5000, 1 \leq k \leq 20$).

The next line contains n space-separated integers a_i ($1 \leq a_i \leq 10^7$) — the denominations of the bills that are used in the country. Numbers a_i follow in the strictly increasing order.

The next line contains integer q ($1 \leq q \leq 20$) — the number of requests for cash withdrawal that you will make.

The next q lines contain numbers x_i ($1 \leq x_i \leq 2 \cdot 10^8$) — the sums of money in burles that you are going to withdraw from the ATM.

Output

For each request for cash withdrawal print on a single line the minimum number of bills it can be done, or print - 1, if it is impossible to get the corresponding sum.

Examples

input
6 20 10 50 100 500 1000 5000 8 4200 100000 95000 96000 99000 10100 2015 9950
output
6 20 19 20 -1 3 -1 -1

input
5 2 1 2 3 5 8 8 1 3 5 7 9 11 13 15
output
1 1 1 2 2 2 2 -1