# B. Cakes (cakes)

It is Liliana's birthday, and she has invited all her friends to celebrate! To make the party extra special, she plans to serve multiple cakes, each decorated with various toppings such as strawberries, almonds, or pralines. Liliana has $N$ types of toppings, and she owns $a_i$ pieces of topping $i$.

The tastiness of a cake is determined by the number of times the most frequent topping appears on it. For example:

- A cake with toppings $\{1, 1, 2, 2, 2\}$ has a tastiness of 3, because topping 2 appears three times.
- A cake with toppings $\{0, 0, 1, 1, 2\}$ has a tastiness of 2, because both toppings 0 and 1 appear twice and no topping appears more often.

Liliana wants to bake several cakes of the same tastiness while using **all toppings** without any leftovers. She has not yet decided how many cakes she wants to bake. She is considering $Q$ scenarios, each specifying a particular number of cakes, $K_j$. For each scenario, determine whether it is possible to distribute all her toppings to create exactly $K_j$ cakes, all with the same tastiness. The cakes may have different amounts of topping, but every cake needs to receive at least one topping. Please note that different cakes may contain a different number of topping types.

## Input

The first line of input contains two integers $N$ and $Q$, representing the number of topping types and the number of scenarios. The second line contains $N$ integers, $a_0, a_1, ..., a_{N-1}$, where $a_i$ denotes the number of pieces of topping $i$. The following $Q$ lines each contain a single integer, $K_j$, specifying the number of cakes for scenario $j$.

## Output

Output $Q$ lines. The $j$th line should contain `YES` if it is possible to distribute all toppings into exactly $K_j$ cakes with the same tastiness, and `NO` otherwise.

## Constraints

- $1 \leq N, Q \leq 100\,000$.
- $1 \leq a_i \leq 100\,000$.
- $1 \leq K_j \leq 10^{18}$.

## Scoring

Your program will be tested on several test cases grouped into subtasks. To obtain the score for a subtask, you must correctly solve all the tests it contains.

- **Subtask 0 [ 0 points]**: Examples.
- **Subtask 1 [ 9 points]**: $N = 1$.
- **Subtask 2 [22 points]**: $Q = 1$ and $K_j = 2$.
- **Subtask 3 [24 points]**: $Q \leq 5$, $N \leq 1000$, $a_i \leq 1000$.
- **Subtask 4 [24 points]**: $Q \leq 5$.
- **Subtask 5 [21 points]**: No additional constraints.

## Examples

| stdin | stdout |
|---|---|
| 4 5<br>2 5 1 1<br>1<br>2<br>3<br>4<br>5 | YES<br>NO<br>YES<br>NO<br>YES |
| 1 1<br>4<br>2 | YES |
| 5 3<br>1 1 1 1 1<br>1<br>1000000000000000000<br>5 | YES<br>NO<br>YES |

In the first example, Liliana has four types of toppings: two toppings of type 0 (depicted by green triangles), five toppings of type 1 (depicted by yellow stars), one topping of type 2 (depicted by an orange circle), and one topping of type 3 (depicted by a blue square).

For $K = 1$, Liliana can make one cake with a tastiness of 5, by putting all the toppings on a single cake as follows:

- Cake 1: $\{0, 0, 1, 1, 1, 1, 1, 2, 3\}$ (topping 1 appears five times).

Figure 1: Example distribution for $K = 1$.

For $K = 2$, it is impossible for Liliana to distribute all her toppings to make two cakes with the same tastiness.

For $K = 3$, Liliana can make 3 cakes, each with a tastiness of 2, by distributing the toppings as follows:

- Cake 1: $\{0, 0, 1\}$ (topping 0 appears twice).
- Cake 2: $\{1, 1, 2\}$ (topping 1 appears twice).
- Cake 3: $\{1, 1, 3\}$ (topping 1 appears twice).

Figure 2: Example distribution for $K = 3$.

For $K = 4$, it is impossible for Liliana to distribute all her toppings to make four cakes with the same tastiness.

For $K = 5$, Liliana can make five cakes, each with a tastiness of 1, by distributing the toppings as follows:

- Cake 1: $\{0, 1\}$ (toppings 0 and 1 each appear once).
- Cake 2: $\{0, 1\}$ (toppings 0 and 1 each appear once).
- Cake 3: $\{1\}$ (topping 1 appears once).
- Cake 4: $\{1, 2\}$ (toppings 1 and 2 each appear once).
- Cake 5: $\{1, 3\}$ (toppings 1 and 3 each appear once).

Figure 3: Example distribution for $K = 5$.