Primitive Root

In the field of Cryptography, prime numbers play an important role. We are interested in a scheme called "Diffie-Hellman" key exchange which allows two communicating parties to exchange a secret key. This method requires a prime number **p** and **r** which is a primitive root of p to be publicly known. For a prime number p, r is a primitive root if and only if it's exponents r, r^2, r^3 ... r^{p-1} are distinct (mod p).

Cryptography Experts Group (CEG) is trying to develop such a system. They want to have a list of prime numbers and their primitive roots. You are going to write a program to help them. Given a prime number p and another integer r < p, you need to tell whether r is a primitive root of p.

Input

There will be multiple test cases. Each test case starts with two integers **p** ($p < 2^{31}$) and **n** ($1 \leq n \leq 100$) separated by a space on a single line. p is the prime number we want to use and n is the number of candidates we need to check. Then n lines follow each containing a single integer to check. An empty line follows each test case and the end of test cases is indicated by p=0 and n=0 and it should not be processed. The number of test cases is at most 60.

Output

For each test case print "YES" (quotes for clarity) if r is a primitive root of p and "NO" (again quotes for clarity) otherwise.

Example

Input:

```
5 2
3
4

7 2
3
4

0 0
```

Output:

```
YES
NO
YES
NO
```

Explanation

In the first test case 3^1, 3^2 , 3^3 and 3^4 are respectively 3, 4, 2 and 1 (mod 5). So, 3 is a primitive root of 5.

4^1, 4^2 , 4^3 and 4^4 are respectively 4, 1, 4 and 1 respectively. So, 4 is not a primitive root of 5.