10692   Huge Mod

The operator for exponentiation is different from the addition, subtraction, multiplication or division operators in the sense that the default associativity for exponentiation goes right to left instead of left to right. So unless we mess it up by placing parenthesis, 2^{3^2} should mean

$$2^{3^{4^5}} \mod 10 = \ ?$$

$2^{(3^2)} = 2^9 = 512$ not $(2^3)^2 = 8^2 = 64$. This leads to the obvious fact that if we take the levels of exponents higher (i.e., 2^3^4^5^3), the numbers can become quite big. But let's not make life miserable. We being the good guys would force the ultimate value to be no more than 10000.

Given $a_1, a_2, a_3, \ldots, a_N$ and $m(= 10000)$ you only need to compute $a_1\hat{\ }a_2\hat{\ }a_3\hat{\ }\ldots\hat{\ }a_N \mod m$.

Input

There can be multiple (not more than 100) test cases. Each test case will be presented in a single line. The first line of each test case would contain the value for M ($2 \leq M \leq 10000$). The next number of that line would be N ($1 \leq N \leq 10$). Then N numbers — the values for $a_1, a_2, a_3, \ldots, a_N$ would follow. You can safely assume that $1 \leq a_i \leq 1000$. The end of input is marked by a line containing a single hash ('#') mark.

Output

For each of the test cases, print the test case number followed by the value of $a_1\hat{\ }a_2\hat{\ }a_3\hat{\ }\ldots\hat{\ }a_N \mod m$ on one line. The sample output shows the exact format for printing the test case number.

Sample Input

```
10 4 2 3 4 5
100 2 5 2
53 3 2 3 2
#
```

Sample Output

```
Case #1: 2
Case #2: 25
Case #3: 35
```