Huge Pascal triangle

Given ***P*** a prime number, and ***N*** an integer, Dukkar found a really fast way to compute how many numbers are divisible by ***P*** on the N^{th} row of the Pascal triangle. Now the task will be much harder : it's for all the ***N*** first rows.
Moreover ***N*** will be a giant number, given in base ***P*** for convenience.

Input

The first line of input contains an integer ***T***, the number of test cases. Follow 2×***T*** lines.
For each test case, on the first line your are given two integers ***P*** and ***k***.
On the second line you are given ***k*** integers : the digits of ***N*** in base ***P***.

$$N = a_0 \times P^0 + ... + a_i \times P^i + ... + a_{k-1} \times P^{k-1}$$

Output

For each test case, you have to print the number of numbers in all the first ***N*** rows of the Pascal triangle that are divisible by ***P***. As the answer could not fit in a 64bit container, give your answer modulo 1000000007.

Example

Input:
3
5 2
0 1
5 2
1 1
7 3
2 0 2

Output:
0
4
2689

Explanations

For the first case, N = $0\times5^0 + 1\times5^1 = 5$. No numbers are divisible by 5 in the first 5 rows.
For the second case, N = $1\times5^0 + 1\times5^1 = 6$. Only 4 numbers are divisible by 5 in the first 6 rows.

1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 **5 10 10 5** 1

For the third case, N = $2\times7^0 + 0\times7^1 + 2\times7^2 = 100$.

Constraints

$0 < T < 300$
$0 < P < 10^9$, a prime number
$0 < k < 1000$
$0 <= a_i < P$, $a_{k-1}>0$

For your information, my 300B-python3 code get AC in 3.03s with 11MB of memory print.
My C-code get AC in 0.08s with 1.6MB of memory print.
Have fun ;-)

Edit(25/I/2015) With Cube cluster my C-time is 0.01s and my PY3.4-time is 0.26s.