Enumerating Substrings

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

There's an alphabet of size k. For a string S in this alphabet (the text), and string P (the pattern), let $F(S, P)$ = the maximum number of non-overlapping substrings you can take in S, that are equal to P.

Let's call a string Q *beautiful*, if each letter in it occurs no more than 2 times.

Over all possible strings of size n, and all possible beautiful patterns P of size m, calculate the sum of $F(S, P)$. Because this sum can be huge, output the result modulo $10^9 + 7$.

Input

The first and only line of the input contains 3 integers, n, m, k ($1 \le n \le 10^6$, $1 \le m \le 2000$, $m \le n$ and $1 \le k \le 10^9$) — respectively, the length of string S, the length of the pattern P and the alphabet size.

Output

Print a single line, containing one integer — the sum of $F(S, P)$ over all strings S and beautiful strings P modulo $10^9 + 7$.

Examples

standard input	standard output
4 2 3	228
999999 1999 12345678	52352722