Problem D. Double Palindrome

Time limit: 2 seconds
Memory limit: 512 megabytes

A *palindrome* is a string that reads the same backward as forward. For example, `rotator`, `lil` and `abba` are palindromes, but `shalash` is not.

A *double palindrome* is a string that is either a palindrome or a concatenation of two (not necessarily distinct) palindromes. For example, `susanna`, `potato` and `abba` are double palindromes, but `zzyzx` and `abaabb` are not.

Dalila has just realized that her name is a double palindrome! Now she wonders how many non-empty strings of length at most n composed of the first k English letters have the same property. Help her find this number and output it modulo 998 244 353.

Input

The only line contains two integers n and k — the maximum length of the string and the size of the alphabet ($1 \le n \le 10^5$; $1 \le k \le 26$).

Output

Output the number of non-empty double palindromes of length at most n composed of the first k English letters, modulo 998 244 353.

Examples

standard input	standard output
3 3	33
6 2	114
42 7	83419789

Note

In the first example the strings to be counted are: `a`, `b`, `c`, `aa`, `ab`, `ac`, `ba`, `bb`, `bc`, `ca`, `cb`, `cc`, `aaa`, `aab`, `aac`, `aba`, `abb`, `aca`, `acc`, `baa`, `bab`, `bba`, `bbb`, `bbc`, `bcb`, `bcc`, `caa`, `cac`, `cbb`, `cbc`, `cca`, `ccb`, `ccc`.