

```
#####C/C++/Rust/Pascal 5#####10#
#####C/C++/Rust/Pascal 1024 M#####2048 M
64bit IO Format: %lld
```

#####

Pigeon aspires to become a string master. Every day he trains extra-hard on string problems.

Today he is solving a problem that goes like this:

Given a string S of length n . You need to perform the following operation exactly k times:

- For the i -th operation ($1 \leq i \leq k$) Choose a substring of S as S_i . $S[l_i, r_i]$ is substring of S . (i.e., the contiguous substring of S starting at position l_i and ending at position r_i , where $1 \leq l_i \leq r_i \leq n$). In addition, the empty string is also a substring of S .
- Concatenate the chosen substrings $S_i (1 \leq i \leq k)$ in order to form a new string $S_1 + S_2 + \dots + S_k$.

Let X be the number of distinct strings that can be formed by concatenating k substrings in this way. Output $X \bmod 998244353$.

#####:

```
The first line contains two integers  $n$  and  $k$  ( $1 \leq n \leq 5 \times 10^5, 1 \leq k \leq 10^9$ ).

The second line contains a string  $S$  of length  $n$ , only consisting of both uppercase and lowercase English letters.

Note that uppercase and lowercase letters are treated as distinct.
```

#####:

```
Output a single integer, the number of distinct strings that can be formed by concatenating  $k$  substrings
(selected in order from the 1st to the  $k$ -th operation) modulo 998244353.
```

#####1

```
##
2 2
```

C++20clang++18

1

ACM

#####

#####

###

#####

#####