Left Shifting 3

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

Given a string $S = s_0 s_1 \cdots s_{n-1}$ of length n, you can shift S to the left for at most k times (including zero times). Calculate the maximum number of "nanjing" substrings contained in the string after the operations.

More formally, let $f(S, d)$ be the string obtained by shifting S to the left d times. That is, $f(S, d) = s_{(d+0) \bmod n} s_{(d+1) \bmod n} \cdots s_{(d+n-1) \bmod n}$. Let $g(f(S, d), l, r) = s_{(d+l) \bmod n} s_{(d+l+1) \bmod n} \cdots s_{(d+r) \bmod n}$. Let $h(d)$ be the number of integer pairs (l, r) such that $0 \le l \le r < n$ and $g(f(S, d), l, r) =$ `nanjing`. Find an integer d such that $0 \le d \le k$ to maximize $h(d)$ and output this maximized value.

Input

There are multiple test cases. The first line of the input contains an integer T indicating the number of test cases. For each test case:

The first line contains two integers n and k ($1 \le n \le 2 \times 10^5$, $0 \le k \le 10^9$) indicating the length of the string and the maximum number of left shifts you can perform.

The second line contains a string $s_0 s_1 \cdots s_{n-1}$ of length n. The string consists of lower-cased English letters.

It's guaranteed that the sum of n of all test cases will not exceed 5×10^5.

Output

For each test case, output one line containing one integer, indicating the maximum number of "nanjing" substrings contained in the string.

Example

standard input	standard output
4 21 10 jingicpcnanjingsuanan 21 0 jingicpcnanjingsuanan 21 3 nanjingnanjingnanjing 4 100 icpc	2 1 3 0

Note

For the first sample test case, we can shift the string to the left 6 times and get the string "pcnanjingsuananjingic". There are two "nanjing" substrings.

For the second sample test case, because $k = 0$, we cannot perform any left shifting. There is one "nanjing" substring in the original string.