String Divide II

Input file:	standard input
Output file:	standard output
Time limit:	15 seconds
Memory limit:	512 megabytes

Given a string $s = s_1 s_2 \ldots s_n$ of length n, consisting of only lowercase English letters. For convenience, we define string $s_{[l,r]} = s_l s_{l+1} \ldots s_r$, which is the substring of s from index l to r.

Given an integer k, you're curious about those substrings of s which are composed by concatenating k identical strings together. Please find out the longest length of those substrings.

Formally, you should find a set of intervals $\{[l_1, r_1], [l_2, r_2], \ldots, [l_k, r_k]\}$ which satisfies the following two conditions:

- For each i $(1 \leq i \leq k)$, $1 \leq l_i \leq r_i \leq n$;

- For each i $(1 \leq i < k)$, $r_i + 1 = l_{i+1}$, and $s_{[l_i, r_i]} = s_{[l_{i+1}, r_{i+1}]}$.

You need to maximize $r_k - l_1 + 1$.

Input

The first line contains two integers n, k $(2 \leq k \leq n \leq 10^6)$.

The second line contains a string s of length n, consisting of only lowercase English letters.

Output

Output a single integer, denoting the answer. Specially, if there are no such substrings, output 0.

Examples

standard input	standard output
5 3 bacbc	0
7 2 ababbba	4

Note

For the second example, the set we choose is $\{[1, 2], [3, 4]\}$. It can be proved that this is the longest substring satisfying the condition.