

String Builder

Input file: *standard input*
Output file: *standard output*
Time limit: 2 seconds
Memory limit: 256 MB

It's the night before the big derby in Romania's Second Football Division, the pinnacle of World Football.

The Professor had a dream about the winning strategy his team must use, and now he asks you to help him recreate it. He knows that the winning strategy is represented by a known string s , formed by lowercase latin characters.

He is building it on his whiteboard, step by step, starting from the empty string.

At each step, having the string t on his whiteboard, he can do any of the following 3 operations:

1. Append any character after t , with cost 1
2. Append any **substring** of t to t , with cost k_1
3. Append **the reverse of any substring** of t to t , with cost k_2

The Professor wants to recreate s by applying these operations while spending as little as possible, so he can have enough money left for organizing the traditional after-game barbecue.

Help the Professor find the solution, and he will reward you with a glass of *Saint Emilion Grand Cru*.

Input

The first line contains k_1 and k_2 , where $1 \leq k_1, k_2 \leq 10^9$ - the costs for operations 2 and 3.

The second line contains the target string s , where $1 \leq |s| \leq 10^5$, formed only by lowercase latin characters.

Output

A single integer, representing the final minimal cost of building string s using the operations above.

Example

standard input	standard output
2 3 ababbabac	8

Explanation

The Professor can apply the following operations in order:

- Starting from "", and appending *a* with cost 1, he gets "*a*"
- Starting from "*a*", and appending *b* with cost 1, he gets "*ab*"
- Starting from "*ab*", and appending *ab* with cost 2, he gets "*abab*"
- Starting from "*abab*", and appending *baba* with cost 3, he gets "*ababbaba*"
- Starting from "*ababbaba*", and appending *c* with cost 1, he gets "*ababbabac*"

This sequence of cost 8 is minimal.