Cheapest Palindrome

https://vjudge.net/problem/poj-3280

Keeping track of all the cows can be a tricky task so Farmer John has installed a system to automate it. He has installed on each cow an electronic ID tag that the system will read as the cows pass by a scanner. Each ID tag's contents are currently a single string with length M ($1 \le M \le 2{,}000$) characters drawn from an alphabet of N ($1 \le N \le 26$) different symbols (namely, the lower-case roman alphabet).

Cows, being the mischievous creatures they are, sometimes try to spoof the system by walking backwards. While a cow whose ID is "abcba" would read the same no matter which direction the she walks, a cow with the ID "abcb" can potentially register as two different IDs ("abcb" and "bcba").

FJ would like to change the cows's ID tags so they read the same no matter which direction the cow walks by. For example, "abcb" can be changed by adding "a" at the end to form "abcba" so that the ID is palindromic (reads the same forwards and backwards). Some other ways to change the ID to be palindromic are include adding the three letters "bcb" to the begining to yield the ID "bcbabcb" or removing the letter "a" to yield the ID "bcb". One can add or remove characters at any location in the string yielding a string longer or shorter than the original string.

Unfortunately as the ID tags are electronic, each character insertion or deletion has a cost ($0 \le cost \le 10{,}000$) which varies depending on exactly which character value to be added or deleted. Given the content of a cow's ID tag and the cost of inserting or deleting each of the alphabet's characters, find the minimum cost to change the ID tag so it satisfies FJ's requirements. An empty ID tag is considered to satisfy the requirements of reading the same forward and backward. Only letters with associated costs can be added to a string.

Input

Line 1: Two space-separated integers: N and M
Line 2: This line contains exactly M characters which constitute the initial ID string
Lines 3..N+2: Each line contains three space-separated entities: a character of the input alphabet and two integers which are respectively the cost of adding and deleting that character.

Output

Line 1: A single line with a single integer that is the minimum cost to change the given name tag.

Sample

Input	Output
3 4 abcb a 1000 1100 b 350 700 c 200 800	900

Hint

If we insert an "a" on the end to get "abcba", the cost would be 1000. If we delete the "a" on the beginning to get "bcb", the cost would be 1100. If we insert "bcb" at the begining of the string, the cost would be 350 + 200 + 350 = 900, which is the minimum.