Problem C. Antipalindrome

Input file:	*standard input*
Output file:	*standard output*
Time limit:	3 seconds
Memory limit:	256 mebibytes

Consider an alphabet consisting of lowercase and uppercase English letters. Let's enumerate the letters of the alphabet in such a way that lowercase letters have numbers from 1 to 26 (in alphabetical order) and uppercase letters have numbers from 27 to 52 (also in alphabetical order). For example, symbol b has number 2, symbol Y has number 51.

You are given a string s consists of the first k symbols of alphabet. You are also given a matrix C of size $k \times k$, consisting of non-negative integers. The element C_{ij} denotes the cost to change the symbol number i to the symbol number j at some position of the string s. It's guaranteed that $C_{ii} = 0$.

You should change some symbols of the string s in such a way that s will not contain any palindrome substring of length more than 1. Also, find the cheapest way to do that.

Note the changes of the symbol number i to the symbol number j at different positions are counted separately. Note also that you can change the symbol at each position not more than once.

Input

The first line contains one integer k ($1 \le k \le 52$) — the size of the alphabet.

The second line contains a string s ($1 \le |s| \le 5 \cdot 10^6$) constisting of only lowercase and uppercase English letters with numbers not greater than k.

The i-th of the next k lines contains k integers C_{ij} ($0 \le C_{ij} \le 10^9$) — the cost to change the symbol number i to the symbol number j.

Note that the length of the string s may be quite large, so use fast methods to read it (for example function `scanf` in `C++` language and `BufferedReader` class in `Java` language).

Output

If it is possible to get a string without palindrome substrings of length more than 1 print the single integer c — the minimum cost to obtain such a string. Otherwise print the only integer "`-1`" (without quotes).

Examples

standard input	standard output
3 aaa 0 7 5 1 0 1 1 1 0	12
3 abc 0 1 1 1 0 1 1 1 0	0