

Too Long

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

Given a set of N strings, and an array A of length 26 represents a value for each character (A_1 corresponds to the character 'a', A_2 corresponds to the character 'b', A_{26} corresponds to the character 'z').

Let's define a valid substring as a substring that has a frequency of each character bigger than or equal to the corresponding value of this character in array A . For example, if $A_1 = 3$ and $A_2 = 4$, then this substring must contain at least 3 characters of 'a' and 4 characters of 'b'.

You are given some queries, each query is one of two types:

1. $1\ i\ c$: Append the character c to the end of the string at index i .
2. 2: Let's calculate array g where g_i is the length of the shortest valid substring in string s_i ($1 \leq i \leq N$). You should print the minimum and maximum g_i or print -1 if there are no valid substrings.

A substring of a string is a contiguous subsequence of that string. So, string "bcb" is a substring of string "aabcbbaa", but string "bca" is not.

Input

The first line of the input contains an integer t ($1 \leq t \leq 10$) represents the number of test cases.

The first line of each test case contains an integer N ($1 \leq N \leq 10^6$) represents the number of strings.

Followed by N lines, the i_{th} line contains a string which consists of lowercase Latin letters.

Followed by two lines each one containing 13 integers representing the array A ($0 \leq A_i \leq 10^9$).

Followed by a line containing an integer Q ($1 \leq Q \leq 10^6$) represents the number of queries.

Followed by Q lines, each of them will be either:

$1\ i\ C$: ($1 \leq i \leq N$), C is a lowercase letter, represents the query from the first type.

Or

2: represents query from the second type.

It's guaranteed that:

Each test case contains **at least** one query of the second type, the sum of N and Q over all test cases don't exceed 10^6 . The sum of lengths of strings after all queries overall test cases doesn't exceed $2 \cdot 10^6$.

Output

For each query of the second type, print minimum and maximum values as described in the statement, or -1 if there are no such substrings.

Example

standard input	standard output
2	6 10
3	5 7
abcabbb	-1
abccbabb	
aacccccbbba	
2 3 0 0 0 0 0 0 0 0 0 0	
0 0 0 0 0 0 0 0 0 0 0 0	
3	
2	
1 3 a	
2	
2	
abc	
def	
2 3 0 0 0 0 0 0 0 0 0 0	
0 0 0 0 0 0 0 0 0 0 0 16	
1	
2	

Note

For the **first** test case:

The valid substrings must contain at least 2 letters of 'a', 3 letters of 'b', and 0 letters for all other characters. Before the first query, the valid substrings will be:

For the first string "abcabbb", the valid substring will be from [1 : 6] inclusive (one-based), so $g_1 = 6$.

For the second string "abccbabb", the valid substring will be from [1 : 7] inclusive (one-based), so $g_2 = 7$.

For the third string "aacccccbbba", the valid substring will be from [1, 10] inclusive (one-based), so $g_3 = 10$.

So the minimum value among the values [6, 7, 10] will be 6, and the maximum value will be 10.

After the second query, the character 'a' will be appended to the end of the third string.

Before the third query, the valid substrings will be:

For the first string "abcabbb", the valid substring will be from [1 : 6] inclusive (one-based), so $g_1 = 6$.

For the second string "abccbabb", the valid substring will be from [1 : 7] inclusive (one-based), so $g_2 = 7$.

For the third string "aacccccbbbaa", the valid substring will be from [9 : 13] inclusive (one-based), so $g_3 = 5$.

So the minimum value among the values [6, 7, 5] will be 5, and the maximum value will be 7.

You might need to use Fast input/output for large input.