

## String Expansion

### Problem description

We gave an example of string expansion in the question of “Reading the program, writing results” in the preliminary competition for the junior group: If the input string contains a substring similar to “d-h” or “4-8”, we use it as a shorthand and replace the minus sign with successive increasing letters or numbers when outputting, that is, output the above two substrings as “defgh” and “45678”, respectively. In this case, we add some parameter settings to make the string expansion more flexible. Specific provisions are as follows:

(1) In the following cases, the string needs to be expanded: in the input string, the minus sign “-” appears, both sides of the minus sign are lowercase letters or digits, and according to the ASCII code order, the character on the right of the minus sign is strictly larger than the character on the left.

(2) Parameter  $p_1$ : expansion mode. When  $p_1=1$ , fill in lowercase letters for the letter substring; When  $p_1=2$ , fill in uppercase letters for the letter substring. In both cases, the numeric substring is filled in the same way. When  $p_1=3$ , both letter substrings and numeric substrings are filled with asterisks “\*” as many as the letters to be filled.

(3) Parameter  $p_2$ : the number of repeated fill characters.  $p_2=k$  indicates that the same character needs to be filled in  $k$  consecutive times. For example, when  $p_2=3$ , the substring “d-h” should be expanded to “deeefffgggh”. The characters on both sides of the minus sign do not change.

(4) Parameter  $p_3$ : whether to change to reverse order:  $p_3=1$  means to maintain the original order,  $p_3=2$  means to adopt reverse order output, note that the characters at both ends of the minus sign are still not included. For example, if  $p_1=1$ ,  $p_2=2$ , and  $p_3=2$ , the substring “d-h” should be expanded to “dggffeeh”.

(5) If the character on the right of the minus sign is exactly the next of the character on the left, delete only the minus sign in the middle. For example, “d-e” should be output as “de”, and “3-4” should be output as “34”. If the characters on the right of the minus sign are less than or equal to the characters on the left of the minus sign in the ASCII code order, keep the minus sign in the middle. For example, “d-d” should be output as “d-d”, and “3-1” should be output as “3-1”.

### Input

The input file includes two lines.

The first line contains 3 positive integers separated by spaces, representing parameters  $p_1$ ,  $p_2$ , and  $p_3$  in sequence.

The second line is a string consisting only of numbers, lowercase letters, and minus signs (-).

There are no spaces at the beginning or end of a line.

### **Output**

The output file has only one line, which is the expanded string.

### **Sample Input 1**

1 2 1  
abcs-w1234-9s-4zz

### **Sample Output 1**

abesttuuvvw1234556677889s-4zz

### **Sample Input 2**

2 3 2  
a-d-d

### **Sample Output 2**

aCCCBBDd-d

### **Hint**

40% of the data meet: the length of the string is no more than 5 characters

100% of the data meet:  $1 \leq p_1 \leq 3$ ,  $1 \leq p_2 \leq 8$ ,  $1 \leq p_3 \leq 2$ . The length of the string is no more than 100 characters.