**Generate All Possible Expressions That Evaluate To The Given Target Value**

Given a string *s* that consists of digits (“0”..”9”) and *target*, a non-negative integer, find all expressions that can be built from string *s* that evaluate to the *target*.

When building expressions, you have to insert one of the following operators between each pair of consecutive characters in *s*: “join” or “\*” or “+”. For example, by inserting different operators between the two characters of string “12” we can get either 12 (1 joined with 2) or 2 (1\*2) or 3 (1+2).

Other operators such as “-” or “÷” are NOT supported.

Expressions that evaluate to the *target* but only utilize a part of *s* do not count: entire *s* has to be consumed.

Precedence of the operators is conventional: “join” has the highest precedence, “\*” – medium and “+” has the lowest precedence. For example, 1+2\*34=(1+(2\*(34)))=1+68=69.

You have to return ALL expressions that can be built from string *s* and evaluate to the *target*.

**Example One**

Input:

*s*="222", *target*=24.

Output:

["22+2", "2+22"] and ["2+22", "22+2"] are both correct outputs.

* 22+2=24: we inserted the “join” operator between the first two characters and the “+” operator between the last two characters of *s*.
* 2+22=24: we inserted the “+” operator between the first two characters and the “join” operator between the last two characters of *s*.

**Example Two**

Input: *s*="1234", *target*=11.

Output: ["1+2\*3+4"]

**Example Three**

Input:

*s*="99", *target*=1.

Output:

[]

**Notes**

Input Format: Function has two arguments: *s* and *target*.

Output: Function must return an array of strings where each string is an expression built from *s* evaluating exactly to the *target*.

ALL such possible expressions have to be returned.

Expressions can appear in the array in any order. For example, both [“22+2”, “2+22”] and [“2+22”, “22+2”] will be accepted for *s*="222" and *target*=24. Returned strings must not contain spaces or any characters other than “0”..”9”, “\*”, “+”. All returned strings must start and end with a digit.

Constraints:

* 1 <= length of *s* <= 13
* 0 <= *target* < 10^13

**Custom Input**

Custom Input Format:

First line of input contains string *s*.

Second line of input contains integer *target*.

If *s*=“222” and *target*=24 then custom input would be:

222

24

Custom Output Format:

Valid output consists of zero or more lines each containing an expression. For the sample input described above, one correct output is:

2+22

22+2

another correct output is:

22+2

2+22