Numeral Abbreviation Notation

Most of the writing systems in the world now use a positional system to denote numerals.

But it wasn't always the case. Even as recently as 200 years ago, there were numeral systems which didn't contain '0' (zero) at all. They had to resort to a different non-positional notation to efficiently represent numerals. For the purpose of this problem, let’s consider such a numeral system which represented its powers of 10 as follows:

t = 10
h = 100
T = 1000

For example, the notations:
5T2h3t4, T2h4, 5T2h3t, Tht1 correspond to 5234, 1204, 5230, 1111 respectively.

The notations were a sort of abbreviations:
T2h3t4 = Thousand two hundred and thirty four
Tht = Thousand hundred and ten

If any of the powers (t,h,T) are unmarked it is assumed that 1 is a prefix.
Thus, 3100 is 3Th while 1200 is T2h

Task

Given, an input of two numbers in such notation, your program should be able to add these numbers and output the result in the same notation.

Also, you have to validate if the notational numerals are well-formed. The order of powers of 10 must be in the descending order. It also shouldn't allow 1 as a prefix of t, h or T

2T5ht3, 5h2t are valid.
5t2T3c4, 5h1t are invalid.

If any one of the given input is invalid, you have to print "Invalid notation".

Input

A single line containing two notational numerals (N and M) seperated by a space.
1 <= N,M <= 9998
The input will be such that the sum (N+M) will be within range [1,9999]

Output

If the input is valid: a single line containing one notational numeral (N + M).
If the input is invalid: a single line containing the text "Invalid notation".

Example

Input:

3T4h2t2 Tht

Output:
4T5h3t2

Input:
4T3t2h5 3ht1

Output:
Invalid notation