Question - 1
Find the First Repeated Word in a Sentence

A sentence is minimally defined as a word or group of words. We consider a word to be a sequence of letters (i.e.: *a-zA-Z*) delimited by a space or non-letter character. A *repeated word* is a case-sensitive word that appears more than once in a sentence (e.g.: *'had' ≠ 'Had'*). Because substrings of a word are *not* delimited, they are *not* considered to be words.

Complete the *firstRepeatedWord* function in your editor. It has *1* parameter:

1. A string, *s*, describing a sentence.

It must return a string containing the first *repeated word* in s.

Input Format
The locked stub code in your editor reads a single string, *s*, from stdin and passes it to your function.

Constraints

- $0 < |s| < 1024$
- The following characters are delimiters between words: space, tab, comma (`,`), colon (`:`), semicolon (`;`), dash (`-`), and period (`.`).
- It is guaranteed that each sentence *s* contains one or more repeated words.
- Each word is separated by one or more delimiters.

Output Format
Your function must return the first repeated word in sentence *s*. This will be printed to stdout by the locked stub code in your editor.

Sample Input 1

```
He had had quite enough of this nonsense.
```

Sample Output 1

```
had
```

Explanation
Sample Case 1: 'had' is the first (and only) word to appear twice in the sentence.

Question - 2
Anagram

Sid loves to read short stories. Being a Computer Science student, he decides to do some frequency analysis on his favorite reading material. For each data point, chooses a string of length ***a*** from one book, and a string of length ***b*** from a second book. The strings' lengths differ by no more than 1.

> $|a-b| \leq 1$, where $|x|$ represents the absolute value function.

are from being anagrams of one another. Your challenge is to help him find the minimum number of characters of the first string he needs to change to make it an anagram of the second string. He can neither add nor delete characters from the first string. Only replacement of the characters with new ones is allowed.

Input Format
The first line will contain an integer T representing the number of test cases. Each test case will contain a string having length ($a+b$) which will be concatenation of both the strings described in problem. The string will only contain small letters and without any spaces.

Output Format
An integer corresponding to each test case is printed in a different line i.e., the number of changes required for each test case. Print ‘-1’ if it is not possible.

Constraints

$1 \leq T \leq 100$

$1 \leq a+b \leq 10{,}000$

Sample Input

```
5
aaabbb
ab
abc
mnop
xyyx
```

Sample Output

```
3
1
-1
2
0
```

Explanation
In the five test cases

1. One string must be “aaa” and the other “bbb”. The lengths are $a=3$ and $b=3$, so the difference is less than 1. No characters are common between the strings, so all three must be changed.
2. One string must be “a” and the second “b”. The lengths are $a=1$ and $b=1$, so the difference is less than 1. One character must be changed to them the same.
3. Since the string lengths a and b must differ by no more than 1, the lengths are either $a=1$ and $b=2$ or $a=2$ and $b=1$. No sequence of substitutions will make the two anagrams of one another.
4. One string must be “mn" and other be “op”. The length are $a=2$ and $b=2$, so the difference is less than 1. No characters are common between the strings, so both must be changed.
5. One string must be “xy” and the other be “yx”. The length are $a=2$ and $b=2$, so the difference is less than 1. No changes are needed because the second string is already an anagram of the first.

Question - 3

The Huffman Decoder

A fragment of text is picked up from Wikipedia Each distinct character in this fragment is then assigned a unique numeric code using a Huffman encoding scheme. Huffman codes are used for compressing text by giving the characters with the highest occurrence frequency the shortest possible code. One requirement of such encoding schemes is that no code can be a prefix of another. For instance, it is not possible for us to encode the letter 'a' as 01 and then encode 'b' as 011.

Task

Given the dictionary of the encoding scheme and the compressed encoded version of the Wikipedia text fragment, recover the original text.

Input Format

- The first input line contains an integer ***N***, the number of characters in the Huffman dictionary.
- Each of the next ***N*** lines contains a character ***c,*** followed by a *tab*, followed by the character's Huffman encoding, a sequence of ones and zeroes. The newline character *‘\n’* is a special case, which will be represented by the string [newline] in the dictionary.
- Finally there is a single line which is the encoded version of the text, based on the encoding dictionary.

Constraints

- $1 \le N \le 100$
- The original text will contain no more than *7,000* characters.

Output Format

The original block of text. Please note that it may contain more than one line.

Constraints

1 ≤ N ≤ 100

The text fragments used will not exceed 5,000 characters.

Sample Input

```
54
[newline] 010100
2 010101
4 010110
9 010111
: 011000
G 011001
K 011010
M 011011
V 011100
j 011101
q 011110
1 011111
6 100000
B 100001
C 100010
I 100011
W 100100
x 100101
" 100110
5 100111
8 101000
E 101001
F 101010
k 101011
R 101100
S 101101
0 101110
A 101111
L 110000
- 110001
. 110010
, 110011
v 110100
b 110101
p 110110
g 110111
w 111000
y 111001
m 111010
f 111011
```

```
u 111100
c 111101
s 111110
h 111111
d 00000
l 00001
r 00010
i 00011
o 00100
a 00101
n 00110
t 00111
e 01000
  01001
1001000001100111111111101001001111111101000010
0111110100100000010000100101110110111110010000
1001001001110110100110110000100111010001010011
0010010001011110000001010000100100011001100100
1001111111110100001001010000010100010000011110
0101001100111001111111110100111110101000001100
0111111100000101110011100110100111000000100001
1000000001000011001001111101010000010111111001
0000000001001001110010001001110101010000100100
1010100111110100101110110000110011100101000010
1001001010011000000010010011111111101000010011
1100000101000010000101000000000100111110100011
0011111100101001001001110110100111000000100001
1000000000110011000011111100111010010011110000
0101111110010010100011101111110110100011101001
1100011110100010000000111100101001001011101010
0101001100000000100001100100000000110011010010
0101000010011111111100100111100110111111111010
0110110000100111010001010011001001111101000111
1010000011000010001111111000101001110001100100
0011001001111111111110000110110111010010010001
1001001000110011001001001111111110100001001101
1010011101001011011001010001000111000110011011
0001000110011011000100111111111010001100011010
1000011010000000100000111110010010010100010010
0000101010011111000011000111000110000101001001
0100010001001111000011000000010010101101001111
0111011001001001101010000100010011101001001001
0100010001001111000011000000010011001111011101
0111011001101001001010011001001101111001101101
1100001001001100011011010010110010100100001100
1001111110010000011100111000010100011101001000
0011000111010011010110011000100111000001100100
1001011111100100111000011110000110000000100000
1101110000001111110101001000000100011010001000
0000100100110110010000000001001000110011001001
0011111111101000010011111100010111101001000010
0100101000100100000101110011010011111100000100
0111101111111110011100001111001010010011100100
0100100111111111010000100111100001000111110001
1101001001001110110100100111111111010000100100
1000000100000010011011000010011101000101001100
1001111101000110011111100111001001001100001111
0010100100101110101001001111000011101001100000
1010001011101100110100100011001110100111111100
1010000001001000100100011010000011110100010000
0000100111111011110011101111110110001111110100
0110100000110001110000111100101001001110010001
0011101010100011110100100111010010000100100101
0100111101000101011101001000001001001110110001
0000010001111100110100100101000010011111111100
1001111001101111111110100100111111111010000001
0010000100100011111110010010000100011001110011
1000010100001001010001101000001100000010000011
0111101010000100100100111011010010000100101000
1011011101000110001111110111101001010000101000
0100111011000010001000000011110011110100111000
1100100001100100100100111011010011101110010000
1000000011111011001001001101010000100010011101
0010010011111111101000010011010000101011011101
1111001001001111111110100001001001110010011100
0001100100100000010001111010000100011001100100
0000000100111010101000111101001011111001111100
1000010010010011101101001000100100011011001000
0010100111010000000001001011100000111010110001
1001101101110100100101001110011100101111101101
0111111101100110100100101001100000001001001111
1111101000010011011110011011011100001001001100
0110110100101100101001000011001001100010111111
```

```
0001000100001100001111110100001010000100100010
0100011110100100000100000001000000000100100111
1111110010100111010010001100111010011110000010
1111110010011001100001001000111011001001111000
0110000000100000000100110010011101011110010100
1101111000011110110001001000000000100100111111
1110100001001011001000100100000101001110100100
0110011001001101000101000100000110010010011011
110001011110111111100101010000001000000010010011
0111000111111010010100001010010001001000111110
0100000101000101111011111110100111111011111100
1001110001111100100100111111111001010011101001
0011111111100011111110010010001100110110100001
00000011101000100000000001001001011101010010100
1100000000100001101110100100000110001110100100
1001110110100111000011110000110000000100000110
111000000011111101010010010100110000000010010010
1010010001001000110100000111101000010100001010
0100100111011010010000100011111011010000100100
10100110000000100100111000100010100000001000010
0111100000011001111111110001100110010010011111
11110100001001001000000010000001001101100001001
1101000101001100100111100000101000010000111111
011001001001110000000100001100000000010000110010
0100111111111010000011001001110111000100100011
10000100111111000001001001110000000011110010100
1111100001100011100011000010100100101110101001
0011110000111010010101111001111011101100110100
1001011110110011101000000100100111100011111100
0111111011111110100100101111101001110001111010
0000110011111100101001000110011011110100010010
00001011111100100000000001001000000001000101111
0100010100111000111111010010100001000011110011
1001001001100001111001010010011111111101000010
01011111011111100111111111010011111010100000110
00111111110000010111001110011010011100000010000
1100000000100001100100111100000101111110010011
1010101000111001001000011000000010010010100001
0000101001111101001001110101101100010100010000
1111111000100001100100100111111111010000100100
0010010100010110111010001111100011101001001110
0100111000001100100100011001100100110100100110
1101110000100101001100000011001001001100100010
0011111000111111010000110011011111000111010000
0010010011011111101011101010100011100111001101
0010001001000110101111100000110000100111010010
0011001100100100111111111010000100110110000100
1110100010100110010001111100111101111000100001
0011111100011111100100001010000100111010111100
1010010110100001100110110111010011010010000011
10000010100010000000010010011111111101000010011
0001000100001101110110100011111011111000100000
1011001101001111000001011111100100100100001100
1000010010010011101101001001111111110100001001
11011100010001010011000000001000111110001110100
1111101111111111100000101111011111110100011111
0010010001100110010011010011111000001000100110
1100100011001001001100100000110011011110111111
10100011111000111010000000100100111111100101000
0001001110110000100100011010000011001001111001
1111000001111001010011101010100001000001100100
1001111111110100001001111101001011101100001100
1110010100001010010010011101101001101111001101
1011100001001001100011011010010110010100100001
1001001101001001101101110000100101001100000011
0011010011101011111000011101001111011000100010
0111010010010011111111100011111110010010011100
0111110100100001001001000011011001101001110000
0010000110000000010000110010011101010100011110
1001011110100100001001001111111110100001001111
0100010100011001100100111101100100000101111001
1101001001111011001000001001001111011001000001
0010000001111011100110010010011100010001010000
0010000001011111001001001010011000000010010011
1111111010000100111010100101111110010000100111
101100100000010010010000000100011101101000001101
1110101000010010001100110010010011100011111010
0100001001001001110110100111100000101000101100
1001001100011001100100100111111111010000100111
0100000110100011100001001001001110110100110101
0000100010100110101011010011011010011101000001
1000111001000011001100001001100110100011001110
```

```
10011111111001010000000100100111111111010000100 1
```

Sample Output

```
With the collapse of Roman rule in the early 5th century,
London ceased to be a capital and the walled city of
Londinium was effectively abandoned, although Roman
civilisation hung on in the St Martin-in-the-Fields area until
around 450. From around 500, an Anglo-Saxon settlement
known as Lundenwic developed in the same area, slightly
to the west of the old Roman city. By about 680, it had
revived sufficiently to become a major port, although there
is little evidence of large-scale production of goods. From
the 820s the town declined because of repeated Viking
attacks, and the Anglo-Saxon Chronicle recorded that it
was "refounded" by Alfred the Great in 886. Archaeological
research shows that this involved abandonment of
Lundenwic and a revival of life and trade within the old
Roman walls. London then grew slowly until about 950,
after which activity increased dramatically. By the 11th
century, London was beyond all comparison the largest
town in England. Westminster Abbey, rebuilt in the
Romanesque style by King Edward the Confessor, was one
of the grandest churches in Europe. Winchester had
previously been the capital of Anglo-Saxon England, but
from this time on, London became the main forum for
foreign traders and the base for defence in time of war. In
the view of Frank Stenton: "It had the resources, and it
was rapidly developing the dignity and the political self-
consciousness appropriate to a national capital."
```

Explanation

Consider the first word of the text (and the space which follows it) "With ". From the encoding dictionary 'W', 'i', 't', 'h' and ' ' correspond to '100100','00011','00111','111111','01001' respectively. So, the encoded version of the text begins with: 100100000110011111111101001.