Problem E - Ew, Not That

Daniel is ordering pizza for ACM practice, and he's taking requests again. He's learned his lesson after taking too many requests last time, so this time he's only going to take 3 requests.

- *"I want something like the **PINESEEDNAPPLETACO** pizza!"*
- *"I'd prefer anything similar to the **PINEAPPLETACO** pizza!"*
- *"Just order any pizza that doesn't have **TACO** in its name!"*

Daniel doesn't think these requests are representative of everyone's tastes, so he plans on ordering a single pizza that respects all their requests. He figures that if he get's a pizza with a name that contains the longest common subsequence of the pizza names request by the first two that does not contain the name of the pizza rejected by the third person as a contiguous substring, then everyone will be happy. Fortunately PanaJohn's selection of pizzas is vast, so they have a pizza of every possible name Daniel can dream up. In the above case, Daniel can just order **PINEAPPLEACO** pizza.

Daniel was wondering about the number of characters in the name of the pizza will order.

Input

The first line contains a single integer, T specifying the number of test cases.

Each test case consists of three lines. The first line has S_1. The second line has S_2. The third line has S_{REJECT}. All strings are non-empty with length ≤ 100, and contain only uppercase English letters.

Output

For each test case, output the integer representing the size of the longest common subsequence of S_1 and S_2 that does not contain S_{REJECT} as a substring. If there is no valid longest common subsequence, ouput 0.

Sample Input

```
4
PINESEEDNAPPLETACO
PINEAPPLETACO
TACO
BCBCBC
BCBCBC
BC
CAM
EAT
A
BDDDBBBABABDAACACCACDDABDACDDCBDCDCBCCBABBBADDADACCCAADCDBCDBBBBDDBBBBDCCCCCDDACACCABCACABDBCACDDDAC
BBBDDCBCBDDBADCBAACCDBBABDCDDBACADADCABADBABCAABBDBADDDABDCDCCBDBABADCCBBBCCDCCBDACABDDDCCDACDBDBCAC
DBBDBABAAAC
```

Sample Output

```
12
3
0
64
```