

A - Copying DNA

Input: dna.in

Output: standard output

Evolution is a seemingly random process which works in a way which resembles certain approaches we use to get approximate solutions to hard combinatorial problems. You are now to do something completely different.

Given a DNA string S from the alphabet $\{A, C, G, T\}$, find the minimal number of copy operations needed to create another string T . You may reverse the strings you copy, and copy both from S and the pieces of your partial T . You may put these pieces together at any time. You may only copy contiguous parts of your partial T , and all copied strings must be used in your final T .

Example: From $S = \text{"ACTG"}$ create $T = \text{"GTACTATTAAT"}$

1. Get GT..... by copying and reversing "TG" from S .
2. Get GTAC..... by copying "AC" from S .
3. Get GTAC...TA.. by copying "TA" from the partial T .
4. Get GTAC...TAAT by copying and reversing "TA" from the partial T .
5. Get GTACAATTAAT by copying "AAT" from the partial T .

Input

The first line of input gives a single integer, $1 \leq t \leq 100$, the number of test cases. Then follow, for each test case, a line with the string S of length $1 \leq m \leq 18$, and a line with the string T of length $1 \leq n \leq 18$.

Output

Output for each test case the minimal number of copy operations needed to create T from S , or "impossible" if it cannot be done.

Sample Input

```
5
ACGT
GTAC
A
C
ACGT
TGCA
ACGT
TCGATCGA
A
```

AAAAAAAAAAAAAAAAAAAA

Sample Output

2
impossible
1
4
6