2820 Gap Punishment Alignment Problem

Consider two strings $X = x_1x_2 \ldots x_m$ and $Y = y_1y_2 \ldots y_n$ over an alphabet set $\sum = \{A, G, C, T\}$. Denote $\sum^\star = \sum \cup \{-\}$, where "$-$" (dash) is the symbol that represents a space (or blank) in strings. A *string alignment* is to align X and Y and form two strings $X^\star, Y^\star$ over the alphabet $\sum^\star$ such that:

1. the two strings $X^\star, Y^\star$ have the same lengths, and
2. ignoring dashes, the string $X^\star$ is the same as the string X, and the string $Y^\star$ is the same as the string Y .

As an example, an alignment of two strings "`GATCCGA`" and "`GAAAGCAGA`" is as follows:

```
G-A--TCCGA
GAAAG-CAGA.
```

There are three gaps in the above alignment; here a *gap* is defined as a string of consecutive dashes. Now, let us consider the following alignment:

```
GA---TCCGA
GAAAG-CAGA.
```

Here are two gaps within this alignment. The rule of measuring the *intermittent gap punishment alignment score* (abbreviated by GPS) is as follows:

- If x_i is aligned with y_j, the score $\sigma(x_i, y_j)$ is

$$\sigma(x_i, y_j) = \begin{cases} 2 & \text{if } x_i = y_j \\ -1 & \text{if } x_i \neq y_j \end{cases}$$

- If a (consecutive) subsequence of x_i's (or y_j's) is aligned with a gap of length k, the score is defined as $-(4 + k)$.

That is, in the first alignment example given above, its GPS is

$$2 - (4 + 1) + 2 - (4 + 2) - (4 + 1) + 2 - 1 + 2 + 2 = -7.$$

For the second alignment, its GPS is

$$2 + 2 - (4 + 3) - (4 + 1) + 2 - 1 + 2 + 2 = -3.$$

Given two strings, the problem we would like to solve is to find an alignment such that its GPS is maximized. Thus, in our example, the best alignment is

```
GA--TCCGA
GAAAGCAGA.
```

Its GPS is $2 + 2 - (4 + 2) - 1 + 2 - 1 + 2 + 2 = 2$.

In our problem, m and n are at most 500. Furthermore, it is required that no space in one sequence is aligned with a space in another.

Input

The input file format is as follows:

1. The first line contains an integer n of sequence pairs; the number n is at most 50.

2. The 2nd line is the sequence X of the first pair.

3. The 3rd line is the other sequence Y of the first pair.

⋮

2i. The $(2i)$-th line is the sequence X of the i-th pair.

2i+1. The $(2i+1)$-th line is the other sequence Y of the i-th pair.

⋮

2n. The $(2n)$-th line is the sequence X of the n-th pair.

2n+1. The $(2n+1)$-th line is the other sequence Y of the n-th pair.

Output

For each pair of sequences, output the *maximum* GPS in one line.

Sample Input

```
3
ACGGCTTAGATCCGAGAGTTAGTAGTCCTAAGCTTGCA
AGCTTAGAAAGCAGACACTTGATCCTGACGGCTTGAA
TTGAGTAGTGTTTTAGTCCTACACGACACATCAAATTCGGACAAGGCCTAGCT
TTCAAGTCCTACAATGTGTGTCAAATTCGCTTGGCCGAAAGCC
TTTGGGAACGTGTGTAGACTTCCCCATGCGATGG
AACACACACGGACTTCATGCTGG
```

Sample Output

```
18
20
2
```