Cars painted in different colors are moving in a row on the road as shown in Figure 1. The color of each car is represented by a single character and the distance of two adjacent cars is assumed to be 1. Figure 1 shows an example of such cars on the road. For convenience, the numbers in Figure 1 represent the locations of each car.

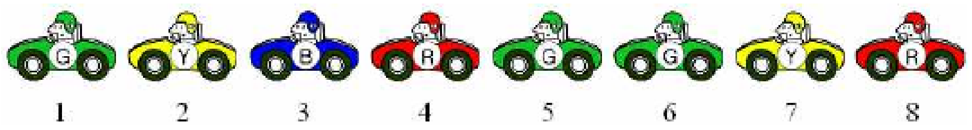


Figure 1. Cars in different colors on the road

For any color c, $location(c)$ represents set of the locations of the cars painted in color c and color length $L(c)$ is defined as follows:

$$L(c) = \max location(c) - \min location(c)$$

For example, according to Figure 1, $location(G) = \{1, 5, 6\}$, $location(Y) = \{2, 7\}$, $location(B) = \{3\}$, and $location(R) = \{4, 8\}$. Hence the color length of each color and the sum of the color lengths are as follows:

Color	G	Y	B	R	**Sum**
$L(c)$	5	5	0	4	14

In Gyeongju City, almost all the roads including the main street of the city were constructed more than 500 years ago. The roads are so old that there are a lot of puddles after rain. Visitors have complained about the bad condition of the roads for many years. Due to the limited budget, the mayor of the city decided to repair firstly the main street of the city, which is a four-lane road, two lanes for each direction.

However, since the main street is a backbone of the city, it should not be blocked completely while it is under repair, or it is expected that serious traffic jams will occur on almost all the other roads in the city. To allow cars to use the main street during the repair period, the city decided to block only two lanes, one lane for each direction. Hence, the cars in the two lanes for each direction should merge into a single lane before the blocked zone.

For instance, as shown in Figure 2, cars in the two lanes merge into a single lane as shown in Figure 3. To differentiate the cars in the same color, a unique identifier is assigned to each car.

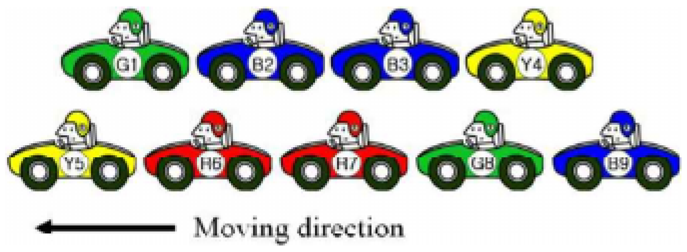
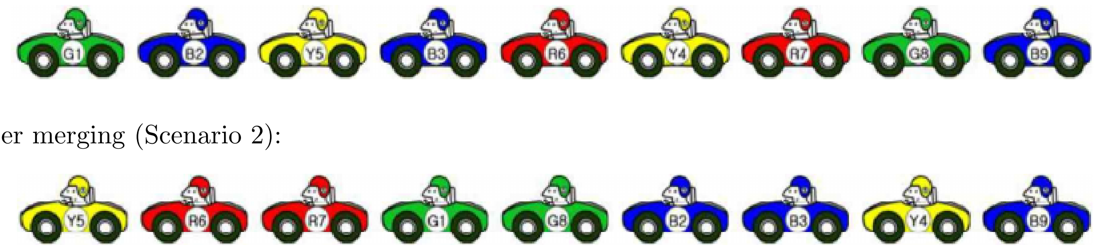


Figure 2. Cars moving in two lanes before merging

Figure 3 shows two different merging scenarios after merging the cars from the two lanes. As shown in Figure 3, cars in the two lanes do not necessarily merge one by one from each lane. The distance between two adjacent cars after merging is also assumed 1.

After merging (Scenario 1):



After merging (Scenario 2):

Figure 3. Two different merging scenarios

For each merging scenario shown in Figure 3, the color length for each color and the sum of the color lengths are as follows:

Color	G	Y	B	R	**Sum**
$L(c)$: Scenario 1	7	3	7	2	19
$L(c)$: Scenario 2	1	7	3	1	12

As you can imagine, there are many different ways of merging other than the two examples shown in Figure 3.

Given two character strings which represent the color information of the cars in the two lanes before merging, write a program to find the sum of color lengths obtained from the character string, which is the color information of cars after merging, such that the sum of color lengths is minimized.

Input

Your program is to read from standard input. The input consists of T test cases. The number of test cases T is given in the first line of the input. Each test case consists of two lines. In the first line, a character string of length n ($1 \le n \le 5,000$) that is the color information of the cars in one lane before merging is given. In the second line, a character string of length m ($1 \le m \le 5,000$) that is the color information of the cars in the other lane is given. Every color is represented as an uppercase letter in English, hence the number of colors is less than or equal to 26.

Output

Your program is to read from standard input. Print exactly one line for each test case. The line should contain the sum of color lengths after merging the cars in the two lanes optimally as described above.

The following shows sample input and output for two test cases.

Sample Input

```
2
AAABBCY
ABBBCDEEY
GBBY
YRRGB
```

Sample Output

```
10
12
```