Problem 8: Edit Distance

The *edit distance* is a string metric quantifying how dissimilar two strings are from each other. The "distance" is the sum of the cost of each edit operation required to transform one string into the other.
In this problem, the edit operations are:

- insert a single character,
- delete a single character, and
- replace one character with another character.

The cost of each operation depends on the ASCII value of the characters involved. Insert and delete operations cost the ASCII value of the character. The replacement cost is the absolute difference of the ASCII values of the characters involved.
The following pseudocode defines a recursive function to compute the ASCII edit distance, given two strings.

```
function first(s) → integer
  if s is empty then
    return 0
  return the ASCII value of the first character of s

function rest(s) → string
  if s is empty then
    return empty string
  return all of s except the first character

function distance(a,b) → integer
  if a is empty and b is empty then
    return 0
  if a is empty then
    return first(b) + dist(a,rest(b))
  if b is empty then
    return first(a) + dist(rest(a),b)
  if first(a) = first(b) then
    return distance(rest(a),rest(b))
  return minimum of
    first(a) + distance(rest(a),b), or
    first(b) + distance(a,rest(b)), or
    | first(a) - first(b) | + distance(rest(a),rest(b))
```

Create a program to determine the ASCII edit distance between strings.
The first line of input contains a single integer (1 ≤ N ≤ 100) indicating the number of distances to compute. The next N line-pairs contain pairs of strings to use. Each string contains at most 100 printable ASCII characters whose values are between 33 and 126, inclusive.

(over)

For each pair of strings, output one line containing an integer indicating the ASCII edit distance between the strings.

Sample input 1

```
5
abcde
cbcde
abcde
abade
lmnopqr
lmnpqr
TUVWXYZ
STUVWXY_
STUVWXY_
TUVWXYZ
```

Sample output 1

```
2
2
111
88
88
```

In the first pair, **a** is replaced by **c**; thus, the ASCII difference is 2.
In the second pair, **c** is replaced by **a**; again, the ASCII difference is 2.
In the third pair, **o** is deleted; its ASCII value is 111.
In the fourth pair, **S** (83) is inserted and **Z** (90) is replaced with **_** (95). Thus, the total distance is 88.
In the final pair, **S** (83) is deleted and **_** (95) is replaced with **Z** (90). Again, the total is 88.