

## Challenge 2 - Hidden numbers

[« PREV](#)[NEXT »](#)

The Nazi air force is trying to transmit numbers (maybe representing army units or available stock) over the radio and the Enigma code has just been broken, so they try to find out a pen and paper solution to send those numbers in a sort of secret manner. They decide to encode them on an arbitrary base using arbitrary symbols to represent the number. The good thing is that we only care about the order of magnitude of such numbers and their minimum and maximum possible values.

You're given a string with characters a-z representing a number. You don't know for sure what base it's in and what digit on that base each character represents. But you know, however, that all digits in the base are present just one time in the input string.

You have to calculate the difference between the minimum and maximum possible values. Bear in mind that the numbers never have leading zeroes.

### Input

The first line has one integer, **T**, which is the number of test cases. For each case, there is a line with a string of lowercase characters, **S**.

### Output

For each line, output "Case #x: " followed by the difference between the minimum and maximum possible values, as a decimal value.

### Limits

$$2 \leq \text{length}(S) \leq 26$$

### Sample input

```
3
bfd
xwyz
qwerty
```

### Sample output

Case #1: 10

Case #2: 153

Case #3: 36445

In the first case (bfd), these are all the possibilities and their values. Each character from bfd represents a digit in base 3, so the possible digits are 0, 1 and 2. Now we substitute each of the letters by a digit (for example, b->1, d->2, f->0 gives you 102). To convert 102 from base 3 to base 10, the calculation is  $1*3^2 + 0*3 + 2 = 11$ . Following the same logic, we can create a table with all the values.

Number	Decimal value
102	11
120	15
210	21
201	19

The maximum value is 21 and the minimum is 11, so the solution is 10. Remember that numbers with leading zeroes are not allowed, so both 012 and 021 are not valid.

## Test your code

You can test your program against both the input provided in the test phase and the input provided in the submit phase. A nice output will tell you if your program got the right solution or not. You can try as many times as you want to. Be careful with extra whitespaces, the output should be exactly as described.

### Test your program against the input provided in the test phase

[Download test input](#)

Program output:

No file chosen

### Test your program against the input provided in the submit phase

[Download input](#)

Program output:

No file chosen