**Code Jam 2016 - World Finals**

# Integeregex

## Problem

In this problem, a valid regular expression is one of the following. In the following descriptions, $E_1$, $E_2$, etc. denote (not necessarily different) valid regular expressions.

- A decimal digit: that is, one of `0 1 2 3 4 5 6 7 8 9`.
- Concatenation: $E_1E_2$.
- Disjunction: `(`$E_1$`|`$E_2$ `|...|`$E_N$`)`, for at least two expressions. Note that the outer parentheses are required.
- Repetition: `(`$E_1$`)*`. Note that the outer parentheses are required.

For example, `7`, `23`, `(7)*`, `(45)*`, `(1|2|3)`, `((2)*|3)`, `(1|2|3)`, and `((0|1))*` are valid expressions. `(7)`, `4|5`, `4*`, `(1|)`, and `(0|1)*` are not.

We say that an expression $E$ matches a string of digits $D$ if and only if at least one of the following is true:

- $E = D$.
- $E = E_1E_2$ and there exist $D_1$ and $D_2$ such that $D = D_1D_2$ and $E_i$ matches $D_i$.
- $E =$ `(`$E_1$`|` $E_2$`|...|` $E_N$`)` and at least one of the $E_i$ matches $D$.
- $E =$ `(`$E_1$`)*` and there exist $D_1$, $D_2$, ..., $D_N$ for some non-negative integer N such that $D = D_1D_2...D_N$ and $E_1$ matches each of the $D_i$. In particular, note that `(`$E_1$`)*` matches the empty string.

For example, the expression `((1|2))*3` matches `3`, `13`, `123`, and `2221123`, among other strings. However, it does *not* match `1234`, `3123`, `12`, or `33`, among other strings.

Given a valid regular expression **R**, for how many integers between **A** and **B**, inclusive, does **R** match the integer's base 10 representation (with no leading zeroes)?

## Input

The first line of the input gives the number of test cases, **T**. **T** test cases follow; each consists of two lines. The first line has two positive integers **A** and **B**: the inclusive limits of the integer range we are interested in. The second has a string **R** consisting only of characters in the set `0123456789()|*`, which is guaranteed to be a valid regular expression as described in the statement above.

## Output

For each test case, output one line containing `Case #x: y`, where `x` is the test case number (starting from 1) and `y` is the number of integers in the inclusive range [**A**, **B**] that the the regular expression **R** matches.

## Limits

Time limit: 20 seconds per test set.
Memory limit: 1 GB.

$1 \le \mathbf{T} \le 100$.
$1 \le \mathbf{A} \le \mathbf{B} \le 10^{18}$.
$1 \le$ length of $\mathbf{R} \le 30$.

**Small dataset (Test Set 1 - Visible)**

**R** contains no | characters.

**Large dataset (Test Set 2 - Hidden)**

No additional limits.

## Sample

### Sample Input

```
8
1 1000
(0)*1(0)*
379009 379009
379009
1 10000
(12)*(34)*
4 5
45
1 100
((0|1))*
1 50
(01|23|45|67|23)
1 10000000000000000000
((0|1|2|3|4|5|6|7|8|9))*
1 1000
1(56|(((7|8))*9)*)
```

### Sample Output

```
Case #1: 4
Case #2: 1
Case #3: 5
Case #4: 0
Case #5: 4
Case #6: 2
Case #7: 10000000000000000000
Case #8: 6
```

Note that sample cases 5 through 8 would not appear in the Small dataset.

In sample case 1, the matches in range are 1, 10, 100, and 1000.

In sample case 2, the match in range is 379009.

In sample case 3, the matches in range are 12, 34, 1212, 1234, and 3434.

In sample case 4, there are no matches in range.

In sample case 5, the matches in range are 1, 10, 11, and 100.

In sample case 6, the matches in range are 23 and 45.

In sample case 7, it is possible to form any number in the range.

In sample case 8, the matches in range are 1, 19, 156, 179, 189, and 199.