Kickstart Practice Round 2 2017

A. Diwali lightings

B. Safe Squares

C. Beautiful Numbers

D. Watson and Intervals

Questions asked

Submissions

Diwali lightings

5pt Not attempted **89/141 users** correct (63%)

8pt Not attempted **62/87 users** correct (71%)

Safe Squares

6pt Not attempted **55/58 users** correct (95%)

13pt Not attempted **25/53 users** correct (47%)

Beautiful Numbers

6pt Not attempted **51/63 users** correct (81%)

15pt Not attempted **14/39 users** correct (36%)

Watson and Intervals

8pt Not attempted **12/15 users** correct (80%)

17pt Not attempted **7/10 users** correct (70%)

Top Scores

Benq	78
1717374	78
yubowenok	78
gridnevvvit	78
LiCode	65
Yash....	53
YourRatzon	53
broncos.billy	53
cmroz	53
sam1373	50

Problem A. Diwali lightings

Small input 5 points — Solve A-small

Large input 8 points — Solve A-large

Problem

Diwali is the festival of lights. To celebrate it, people decorate their houses with multi-color lights and burst crackers. Everyone loves Diwali, and so does Pari. Pari is very fond of lights, and has transfinite powers, so she buys an infinite number of red and blue light bulbs. As a programmer, she also loves patterns, so she arranges her lights by infinitely repeating a given finite pattern **S**.

For example, if **S** is BBRB, the infinite sequence Pari builds would be BBRBBBRBBBRB. . .

Blue is Pari's favorite color, so she wants to know the number of blue bulbs between the **I**th bulb and **J**th bulb, inclusive, in the infinite sequence she built (lights are numbered with consecutive integers starting from 1). In the sequence above, the indices would be numbered as follows:

```
B  B  R  B  B  B  R  B  B  B  R  B...
1  2  3  4  5  6  7  8  9  10 11 12
```

So, for example, there are 4 blue lights between the 4th and 8th positions, but only 2 between the 10th and 12th.

Since the sequence can be very long, she wrote a program to do the count for her. Can you do the same?

Input

The first line of the input gives the number of test cases, **T**. **T** test cases follow.
First line of each test case consists of a string S, denoting the initial finite pattern.
Second line of each test case consists of two space separated integers **I** and **J**, defined 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 number of blue bulbs between the **I**th bulb and **J**th bulb of Pari's infinite sequence, inclusive.

Limits

$1 \le \mathbf{T} \le 100$.
$1 \le$ length of $\mathbf{S} \le 100$.
Each character of **S** is either uppercase B or uppercase R.

Small dataset

$1 \le \mathbf{I} \le \mathbf{J} \le 10^6$.

Large dataset

$1 \le \mathbf{I} \le \mathbf{J} \le 10^{18}$.

Sample

Input	Output
3 BBRB 4 8 BBRB 10 12 BR 1 1000000	Case #1: 4 Case #2: 2 Case #3: 500000

Cases #1 and #2 are explained above.

In Case #3, bulbs at odd indices are always blue, and bulbs at even indices are always red, so there are half a million blue bulbs between positions 1 and 10^6.

Kickstart Practice Round 2 2017

[A. Diwali lightings](#)

B. Safe Squares

[C. Beautiful Numbers](#)

[D. Watson and Intervals](#)

[Questions asked](#)

Submissions

Diwali lightings

5pt Not attempted **89/141 users** correct (63%)

8pt Not attempted **62/87 users** correct (71%)

Safe Squares

6pt Not attempted **55/58 users** correct (95%)

13pt Not attempted **25/53 users** correct (47%)

Beautiful Numbers

6pt Not attempted **51/63 users** correct (81%)

15pt Not attempted **14/39 users** correct (36%)

Watson and Intervals

8pt Not attempted **12/15 users** correct (80%)

17pt Not attempted **7/10 users** correct (70%)

Top Scores

Benq	78
1717374	78
yubowenok	78
gridnevvvit	78
LiCode	65
Yash....	53
YourRatzon	53
broncos.billy	53
cmroz	53
sam1373	50

Problem B. **Safe Squares**

This contest is open for practice. You can try every problem as many times as you like, though we won't keep track of which problems you solve. Read the [Quick-Start Guide](#) to get started.

Small input 6 points	Solve B-small
Large input 13 points	Solve B-large

Problem

Codejamon trainers are actively looking for monsters, but if you are not a trainer, these monsters could be really dangerous for you. You might want to find safe places that do not have any monsters!

Consider our world as a grid, and some of the cells have been occupied by monsters. We define a *safe square* as a grid-aligned $\mathbf{D} \times \mathbf{D}$ square of grid cells (with $\mathbf{D} \geq 1$) that does not contain any monsters. Your task is to find out how many safe squares (of any size) we have in the entire world.

Input

The first line of the input gives the number of test cases, **T**. **T** test cases follow. Each test case starts with a line with three integers, **R**, **C**, and **K**. The grid has **R** rows and **C** columns, and contains **K** monsters. **K** more lines follow; each contains two integers $\mathbf{R_i}$ and $\mathbf{C_i}$, indicating the row and column that the i-th monster is in. (Rows are numbered from top to bottom, starting from 0; columns are numbered from left to right, starting from 0.)

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 the total number of safe zones for this test case.

Limits

$1 \leq \mathbf{T} \leq 20$.

$(\mathbf{R_i}, \mathbf{C_i}) \neq (\mathbf{R_j}, \mathbf{C_j})$ for i ≠ j. (No two monsters are in the same grid cell.)

$0 \leq \mathbf{R_i} < \mathbf{R}$, i from 1 to **K**

$0 \leq \mathbf{C_i} < \mathbf{C}$, i from 1 to **K**

Small dataset

$1 \leq \mathbf{R} \leq 10$.

$1 \leq \mathbf{C} \leq 10$.

$0 \leq \mathbf{K} \leq 10$.

Large dataset

$1 \leq \mathbf{R} \leq 3000$.

$1 \leq \mathbf{C} \leq 3000$.

$0 \leq \mathbf{K} \leq 3000$.

Sample

Input	Output

```
2
3 3 1
2 1
4 11 12
0 1
0 3
0 4
0 10
1 0
1 9
2 0
2 4
2 9
2 10
3 4
3 10
```

```
Case #1: 10
Case #2: 51
```

The grid of sample case #1 is:

```
0 0 0
0 0 0
0 1 0
```

Here, 0 represents a cell with no monster, and 1 represents a cell with a monster. It has 10 safe squares: 8 1x1 and 2 2x2.

The grid of sample case #2 is:

```
0 1 0 1 1 0 0 0 0 0 1
1 0 0 0 0 0 0 0 0 1 0
1 0 0 0 1 0 0 0 0 1 1
0 0 0 0 1 0 0 0 0 0 1
```

Note that sample case #2 will only appear in the Large dataset. It has 51 safe squares: 32 1x1, 13 2x2, 5 3x3, and 1 4x4.

Submissions

Diwali lightings

5pt Not attempted **89/141 users** correct (63%)

8pt Not attempted **62/87 users** correct (71%)

Safe Squares

6pt Not attempted **55/58 users** correct (95%)

13pt Not attempted **25/53 users** correct (47%)

Beautiful Numbers

6pt Not attempted **51/63 users** correct (81%)

15pt Not attempted **14/39 users** correct (36%)

Watson and Intervals

8pt Not attempted **12/15 users** correct (80%)

17pt Not attempted **7/10 users** correct (70%)

Top Scores

Benq	78
1717374	78
yubowenok	78
gridnevvvit	78
LiCode	65
Yash....	53
YourRatzon	53
broncos.billy	53
cmroz	53
sam1373	50

Problem C. Beautiful Numbers

This contest is open for practice. You can try every problem as many times as you like, though we won't keep track of which problems you solve. Read the Quick-Start Guide to get started.

Small input 6 points	Solve C-small
Large input 15 points	Solve C-large

Problem

We consider a number to be *beautiful* if it consists only of the digit 1 repeated one or more times. Not all numbers are beautiful, but we can make any base 10 positive integer beautiful by writing it in another base.

Given an integer **N**, can you find a base B (with $B > 1$) to write it in such that all of its digits become 1? If there are multiple bases that satisfy this property, choose the one that maximizes the number of 1 digits.

Input

The first line of the input gives the number of test cases, **T**. **T** test cases follow. Each test case consists of one line with an integer **N**.

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 base described in the problem statement.

Limits

$1 \leq \mathbf{T} \leq 100$.

Small dataset

$3 \leq \mathbf{N} \leq 1000$.

Large dataset

$3 \leq \mathbf{N} \leq 10^{18}$.

Sample

Input	Output
2 3 13	Case #1: 2 Case #2: 3

In case #1, the optimal solution is to write 3 as 11 in base 2.

In case #2, the optimal solution is to write 13 as 111 in base 3. Note that we could also write 13 as 11 in base 12, but neither of those representations has as many 1s.

Problem D. **Watson and Intervals**

This contest is open for practice. You can try every problem as many times as you like, though we won't keep track of which problems you solve. Read the Quick-Start Guide to get started.

Small input 8 points	Solve D-small
Large input 17 points	Solve D-large

Problem

Sherlock and Watson have mastered the intricacies of the language C++ in their programming course, so they have moved on to algorithmic problems. In today's class, the tutor introduced the problem of merging one-dimensional intervals. **N** intervals are given, and the ith interval is defined by the inclusive endpoints [$\mathbf{L_i}$, $\mathbf{R_i}$], where $\mathbf{L_i} \le \mathbf{R_i}$.

The tutor defined the *covered area* of a set of intervals as the number of integers appearing in at least one of the intervals. (Formally, an integer p contributes to the covered area if there is some j such that $\mathbf{L_j} \le p \le \mathbf{R_j}$.)

Now, Watson always likes to challenge Sherlock. He has asked Sherlock to remove exactly one interval such that the covered area of the remaining intervals is minimized. Help Sherlock find this minimum possible covered area, after removing exactly one of the **N** intervals.

Input

Each test case consists of one line with eight integers **N**, $\mathbf{L_1}$, $\mathbf{R_1}$, **A**, **B**, $\mathbf{C_1}$, $\mathbf{C_2}$, and **M**. **N** is the number of intervals, and the other seven values are parameters that you should use to generate the other intervals, as follows:

First define $x_1 = \mathbf{L_1}$ and $y_1 = \mathbf{R_1}$. Then, use the recurrences below to generate x_i, y_i for i = 2 to **N**:

- $x_i = (\mathbf{A}*x_{i-1} + \mathbf{B}*y_{i-1} + \mathbf{C_1})$ modulo **M**.
- $y_i = (\mathbf{A}*y_{i-1} + \mathbf{B}*x_{i-1} + \mathbf{C_2})$ modulo **M**.

We define $\mathbf{L_i} = \min(x_i, y_i)$ and $\mathbf{R_i} = \max(x_i, y_i)$, for all i = 2 to **N**.

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 minimum possible covered area of all of the intervals remaining after removing exactly one interval.

Limits

$1 \le \mathbf{T} \le 50$.
$0 \le \mathbf{L_1} \le \mathbf{R_1} \le 10^9$.
$0 \le \mathbf{A} \le 10^9$.
$0 \le \mathbf{B} \le 10^9$.
$0 \le \mathbf{C_1} \le 10^9$.
$0 \le \mathbf{C_2} \le 10^9$.
$1 \le \mathbf{M} \le 10^9$.

Small dataset

$1 \le \mathbf{N} \le 1000$.

Large dataset

$1 \le \mathbf{N} \le 5 * 10^5$(500000).

Sample

Input	Output
3 1 1 1 1 1 1 1 1 3 2 5 1 2 3 4 10 4 3 4 3 3 8 10 10	Case #1: 0 Case #2: 4 Case #3: 9

In case 1, using the generation method, the set of intervals generated are: {[1, 1]}. Removing the only interval, the *covered area* is 0.

In case 2, using the generation method, the set of intervals generated are: {[2, 5], [3, 5], [4, 7]}. Removing the first, second or third interval would cause the covered area of remaining intervals to be 5, 6 and 4, respectively.

In case 3, using the generation method, the set of intervals generated are: {[3, 4], [1, 9], [0, 8], [2, 4]}. Removing the first, second, third or fourth interval would cause the covered area of remaining intervals to be 10, 9, 9 and 10, respectively.