Lab Assignment 1 CS109 2024Spring

Problem 1. Numbers That Must Not Be Spoken (30 pts) [Easy]

Qi(小七) is a first-grade student at Kata Star Primary School. He and his friends are playing a game involving the number 7. They sit in a circle and take turns counting, incrementing by 1. However, when they encounter a number that contains a 7 or is a multiple of 7, they must avoid speaking out loud and instead clap their hands.

Please help Qi count how many numbers cannot be spoken within a given range of values.

Input Format

The input contains several lines.

The first line is an integer n ($1 \leq n \leq 2^{20}$), indicating the number of test cases.

The following n lines represent n sets of test data.

Each set of test data occupies one line, consisting of 2 integers l and r ($0 \leq l \leq r \leq 2^{30}$), defining the range of integers to be queried in that set, denoted as $[l, r]$.

We guarantee that in all test cases, the constraint $l \leq r$ is satisfied; thus, no validity checks need to be used for the input.

Output Format

For each set of test data, output an integer m, representing the number of integers within the range that contain a 7 or are multiples of 7.

When the same number satisfies both conditions, it should not be counted twice.

Then, output a newline character.

Followed by the output for the next set of test data.

Samples

Sample 1

Input

```
1
0 10
```

output

```
2
```

Explanation

Within the range $[0, 10]$, the numbers 0 and 7 are divisible by 7, so the output is 2.

Sample 2

Input

```
2
0 10
14 17
```

output

```
2
2
```

Explanation

Within the range $[0, 10]$, the numbers 0 and 7 are divisible by 7, so the output is 2.

Then, a newline character is output.

Within the range $[14, 17]$, the number 14 is a multiple of 7, meeting the condition, and the number 17 contains the digit 7, also meeting the condition, so the output is 2.

Hint

Negative integers are **not** considered in this problem.

The number 0 is considered a multiple of 7.

Problem 2. What's the Number? (30 pts) [Medium]

Qi(小七) and his friends are playing a game of numeric conversion.

In our daily counting, we use the decimal counting system, where we advance one unit after ten. In computing, binary counting is used, where we advance one unit after two. In the field of computer science, octal and hexadecimal counting systems are also commonly used. The numeric conversion game played by Qi and his friends is a form of base conversion game.

In this game, Qi's friends need to convert decimal numbers into septenary (base-7) numbers.

In the septenary counting system, we advance one unit after seven. In other words, when the unit digit is `6`, and one more number is added, it carries over to the "tens place". For example, instead of writing `7`, it's written as `10`.

In the septenary counting system, the valid digits are only 0, 1, 2, 3, 4, 5, and 6.

Please help Qi complete this game.

Input Format

The input contains several lines.

The first line is an integer n ($1 \leq n \leq 2^{20}$), indicating the number of test cases.

The following n lines represent n sets of test data.

Each set of test data occupies one line, and each test data contains only one decimal integer d.

Output Format

For each set of test data, output a septenary integer m, representing the septenary integer corresponding to the decimal integer d in the input.

Samples

Sample 1

Input

```
1
7
```

output

```
10
```

Explanation

The decimal integer 7, when converted to septenary, is 10, hence the output is 10.

Sample 2

Input

```
2
7
1
```

output

```
10
1
```

Explanation

The decimal integer 7, when converted to septenary, is 10 hence the output is 10.

Then, a newline character is output.

The decimal integer 1, when converted to septenary, is 1 hence the output is 1.

Sample 3

Input

```
2
225
20240225
```

output

```
441
334016255
```

Explanation

The decimal integer 225, when converted to septenary, is 441 hence the output is 441.

Then, a newline character is output.

The decimal integer 20240225, when converted to septenary, is 334016255 hence the output is 334016255.

Hint

Negative numbers are **NOT** considered in this problem.

Problem 3. Looking Forward to Summer Vacation (40 pts) [Hard]

After the semester started, Qi(小七) felt anxious and eager to know when summer vacation would come.

The time span of a year on Kata Star is quite long, totaling approximately 916 days. The Kata Star people divide these 916 days into 15 months, represented as Month 1, Month 2, Month 3, ..., Month 15.

Among the 15 months, Month 5 has 62 days, while the other months have only 61 days.

Kata Star also differentiates between regular years and leap years. A regular year consists of 916 days, whereas a leap year consists of 917 days. In leap years, Month 3 also becomes a particular month with 62 days.

Starting from year 0 of the calendar, Kata Star experiences a leap year every 13 years. That means the years AD 0, AD 13, AD 26, etc., are leap years.

Due to Qi's limited calculation ability, please help him calculate how many days are left until summer vacation comes.

Input Format

The input contains several lines.

The first line is a positive integer n ($1 \leq n \leq 2^{20}$), indicating the number of test cases.

The following n lines are the test cases.

Each test case occupies one line, consisting of 6 numbers represented as a string, indicating the current date and the start date of summer vacation.

Each date is given in the format `YEAR MONTH DAY`, where the first number represents the year, the second number represents the month, and the third number represents the day. There is a single space between each number. For example, `11206 3 32` represents the date 32nd, Month 3, 11206 in the Kata Star calendar.

Two dates are also separated by a `space`. For example, input `11206 3 32 11206 7 53` represents the current date as 32nd, Month 3, 11206, and the start date of summer vacation is 53rd, Month 7, 11206.

We guarantee that all input dates for the test cases are in this format and are valid; thus, the second date **will not be earlier than** the first date, no validity checks need to be used.

Output Format

For each test case, output an integer m, representing the number of days between the first date and the second date.

Then, output a newline character.

Followed by the output for the next test case.

Samples

Sample 1

Input

```
1
11206 3 32 11206 3 33
```

output

```
1
```

Explanation

There is a difference of 1 day between 32nd, Month 3, 11206, and 33rd, Month 3, 11206; thus, the output is 1.

Sample 2

Input

```
2
11206 3 32 11206 3 33
11206 3 32 11206 3 34
```

output

```
1
2
```

Explanation

There is a difference of 1 day between 32nd, Month 3, 11206, and 33rd, Month 3, 11206; thus, the output is 1.

Then a newline character is output.

There is a difference of 2 days between 32nd, Month 3, 11206, and 34th, Month 3, 11206; thus, the output is 2.

Sample 3

Input

```
3
65662 3 32 65662 7 33
65663 3 32 65663 7 33
65662 3 32 65663 7 33
```

output

```
246
247
1163
```

Explanation

Year 65662 is a common year, so Month 3, Month 4, and Month 7 have 61 days, and Month 5 has 62 days. There is a difference of 246 day between 32nd, Month 3, 65662, and 33rd, Month 7, 65662; thus, the output is 246.

Then a newline character is output.

Year 65663 is a leap year, so Month 4 and Month 7 have 61 days, Month 3 and Month 5 has 62 days. There is a difference of 247 day between 32nd, Month 3, 65663, and 33rd, Month 7, 65663; thus, the output is 247.

Then a newline character is output.

Year 65662 is a common year, so Month 5 of 65662 has 62 days while other months have 61 days. Year 65663 is a leap year, so Month 3 and Month 5 of 65663 has 62 days while other months have 61 days. There is a difference of 1163 day between 32nd, Month 3, 65662, and 33rd, Month 7, 65663; thus, the output is 1163.

Hint

In the input data, the start date and end date may span different years; remember to handle this case.

We guarantee that the second date **will not be earlier than** the first date for all inputs.

We guarantee that all output results will not exceed the maximum value that can be represented by a 32-bit integer.