CCSC

Consortium for Computing Sciences in Colleges
Central Plains Region

2025 CCSC
Central Plains Region Programming Contest

Official Problem Set

April 5, 2025

CCSC

Consortium for Computing Sciences in Colleges
Central Plains Region

APRIL 5, 2025

Contest Problems

A: The Ants Go Marching
B: Frequency Product
C: Hey Diddle Diddle
D: Legal Maze Path
E: Polly Had a Dolly
F: Polly Put the Kettle On
G: Reincarnation
H: Jack and Jill Count Primes

CCSC
Consortium for Computing Sciences in Colleges
Central Plains Region

This contest contains 7 problems over 24 pages. Good luck.

Before the CCSC competition, the judges will have solved all problems in languages from at least two of the three distinct language groups (Java/Kotlin, C/C++, and Python3).

Definition 1

For problems that ask for a result modulo m:
If the correct answer to the problem is the integer b, then you should display the unique value a such that:

- $0 \leq a < m$

  and

- $(a - b)$ is a multiple of m.

Definition 2

A string $s_1 s_2 \cdots s_n$ is lexicographically smaller than $t_1 t_2 \cdots t_\ell$ if

- there exists $k \leq \min(n, \ell)$ such that $s_i = t_i$ for all $1 \leq i < k$ and $s_k < t_k$

  or

- $s_i = t_i$ for all $1 \leq i \leq \min(n, \ell)$ and $n < \ell$.

Definition 3

- Uppercase letters are the uppercase English letters $(A, B, \ldots, Z)$.
- Lowercase letters are the lowercase English letters $(a, b, \ldots, z)$.

This page is intentionally left blank.

Problem A
The Ants Go Marching

> The ants go marching one by one. Hurrah! Hurrah!
> The ants go marching one by one. Hurrah! Hurrah!
> The ants go marching one by one.
> The little one stops to suck his thumb.
> And they all go marching down to the ground
> to get out of the rain, BOOM! BOOM! BOOM!

Additional verses include (2) tie a shoe, (3) climb a tree, (4) shut the door, (5) take a dive, (6) pick up sticks, (7) pray to heaven.

The march will be triangular rather than rectangular. The first row has 1 ant, the second row has 2 ants, etc. It is quite possible that the very last row will not have a full contingent of ants if there are not enough ants to fill the row.

Given the total number of ants, you are to determine how many ants are in the last row. For example, with 12 ants, the first row has 1 ant, the second has 2, the third has 3, and the fourth has 4, leaving only 2 remaining ants for the fifth and last row.

Input

The input will consist of a single non-negative integer representing the number of ants going on the march. This number will not be larger than 10^9.

Output

Print a single integer indicating the number of ants in the last row.

Sample Input 1	Sample Output 1
`12`	`2`

Sample Input 2	Sample Output 2
`10`	`4`

This page is intentionally left blank.

Problem B
Frequency Product

Knowing your interest in programming contests, your folks bought you a book of coding puzzles for your birthday, You eagerly run off to your room with your friends, sit down at your computer and open the book.

For one of the puzzles, you are given lines of alphanumerics ('a'-'z', 'A'-'Z', '0'-'9') as well as other characters such as punctuation and white space. You are asked to find the product of all non-zero frequencies of each alphanumeric character modulo $10^9 + 7$.

In the first sample, the input has 5 a's, 4 b's, 3 c's, 2 d's, and one e, in that order. There are no additional characters. The answer is therefore $5 * 4 * 3 * 2 * 1 = 120$, which is small enough that the modular division does not have an impact.

Input

The only input line contains a list of alphanumerics along with other characters as described above. The line will not be longer than 1 million characters.

Output

Print the product of all non-zero frequencies of each alphanumeric modulo $10^9 + 7$.

Sample Input 1	Sample Output 1
aaaaabbbbcccdde	120

Sample Input 2

```
asdioguaposidguoipasdugpoiaudopiguoipsuadgoipadusgopiuoi1902348093180 9
qierupoqieuoriqpuiofgdasug98e5093280958e0gdsogaoifubpoiafuidt5093285r9
0qewf8vba0df9b8a09fb8oe9ip623o8ief8aod9f8b09df8609q8w0a9vbafodibudpo68
098
```

Sample Output 2

```
534130650
```

This page is intentionally left blank.

Problem C
Hey, Diddle Diddle

Hey diddle diddle,
The cat and the fiddle,
The cow jumped over the moon.
The little dog laughed to see such sport,
And the dish ran away with the spoon.

Imagine viewing the moon which happens to be somewhat above the horizon, and that there is a cow standing a distance ahead of you, seemingly next to the moon. When the cow jumps, she does so with a running leap, and appears to be jumping over the moon. Your task is to calculate how high the cow has to jump, given your distance to the cow and the apparent angle from your eyes to the top of the moon. Assume the ground is level, that your eyes are 1 1/2 yards above the ground, the cow's belly starts out 1/2 yard above the ground and during the jump her legs are stretched out straight, so they are not a factor in the answer. Therefore the effective jumping height is the highest belly height minus the 1/2 yard initial belly height. The distances to the horizon and the moon are sufficiently large compared to the cow that you may neglect the effect of your eye height on the angle to the top of the moon - 0 degrees is essentially the angle to the horizon.

Input

You will be given two integers, each on a separate line. The first of these indicates your distance to the cow, given in yards. This number will be positive and never more than 2^{30}. (Ignore the curvature of the earth and distance to the moon.) The second is the number of degrees, $0 < d < 90$, that the top of the moon is above the horizon at the time of the jump.

Output

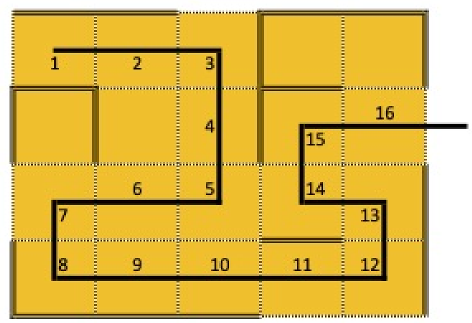
Output the minimum number of whole yards that the cow must jump in order to appear to jump over the moon. Use the following format:

"The cow needs to jump at least x yards into the air."

Sample Input 1	Sample Output 1
100 5	The cow needs to jump at least 10 yards into the air.

Sample Input 2	Sample Output 2
10 1	The cow needs to jump at least 2 yards into the air.

Problem D
Legal Maze Path



The maze specified in the Sample.

A bricklayer is constructing a maze that is being laid out on a grid in the neighborhood playground. As he moves rectilinearly from cell to cell in a meandering fashion, possibly crossing and/or retracing his path, he may choose to build a wall section to his left and/or to his right, or neither (assuming his next step is to go straight ahead). In case he plans to turn to his left or right, he obviously can not build a wall section in the direction he plans to turn, but instead can choose to build wall sections directly ahead and either to the left or right, depending on the direction he is not turning. If he comes back to a completed wall section, even if from the other side of the wall, he just recognizes its presence and should not attempt to climb over or remove it. Similarly, he does not return to a location in order to build a wall section that he could have done earlier - wall sections are always constructed at the first opportunity to do so.

The figure shows only a portion of the completed maze along with a portion of the path that the bricklayer took. Where there are wall sections that are not adjacent to the path, you can assume that his path had taken him there at some point during construction. The only move he does not make is a full 180 degree u-turn and he should not climb over a completed or intended wall section. In other words, the path he follows during construction will remain an open path after completion. Also, if the last move puts him in a cul-de-sac, it would indicate it truly was the final move! We don't care how he exits the completed maze. But finding himself in a cul-de-sac during construction would invalidate any later moves, meaning the path description is inconsistent.

You will be given step by step details of the maze construction, following the turns he takes and any walls he builds along the path. Remember this is just a portion of the final maze and a portion of his complete path. The rest of the maze is not relevant for this problem. Your task is to determine if the reported walls are consistently specified. For example, path segments 6 and 9 should be in agreement that there is no wall separating them and that segment 4 and corner 15 do have a wall separating them. Of course, there should be no wall segment blocking any portion of his path. The length of a path will be at most 1000 steps. The maze is not necessarily rectangular, but you may assume he never strays more than 16 steps from the initial specified location, as measured by the Manhattan distance formula ($\Delta x + \Delta y$).

Input

The first line contains an integer representing the number of steps in the path. Each subsequent line contains a 3-character string that describes one of the steps. The second character of the line will be either 'S' to go straight, 'L' to turn left, or 'R' to turn right. The first and third characters indicate either the presence of a wall '1' or no wall '0' as determined by the direction of the step:

S: The first represents the edge to the left, and the third represents the edge to the right.

L: The first represents the edge to the right prior to turning, and the third represents the edge to the right after turning.

R: The first represents the edge to the left prior to turning, and the third represents the edge to the left after turning.

Hopefully the following will make it clearer. The numbers in the figure are located at the centers of the grid cells. Number 1 is the starting position of the path segment which happens to be at a corner. Number 16 is the start of the final step which happens to lead out of the maze segment. A turn is a pivot that takes place entirely inside a cell. If the bricklayer is in a cell getting ready to turn right, you will see a corner in the path. Here is some annotated input for the beginning of the sample case:

```
16  (length of the path)
1S1 (1 Left wall, Right wall, Step to 2)
1S0 (2 Left wall, Right open, Step to 3)
0R1 (3 Left open, Right turn, Left wall, Step to 4)
1S0 (4 Left wall, Right open, Step to 5)
0R0 (5 Left open, Right turn, Left open, Step to 6)
0S0 (6 Left open, Right open, Step to 7)
0L0 (7 Right open, Left turn, Right open, Step to 8)
1L1 (8 Right wall, Left turn, Right wall, Step to 9)
0S1 (9 Left open, Right open, Step to 10)
0L1 ...
```

Output

Print the string "Yes" or "No" depending on the maze path description being consistent. This means that the presence or absence of walls follows from the description/construction steps as would be discoverable by repeat visits to the same edge, and that the bricklayer does not attempt to cross an edge that currently has or in the future will have a wall.

Sample Input 1	Sample Output 1
16 1S1 1S0 0R1 1S0 0R0 0S0 0L0 1L1 0S1 0S1 1S0 0L1 1L0 1R0 1R1 0S0	Yes

Problem E
Polly Had a Dolly

> Miss Polly had a dolly who was sick, sick, sick.
> So she phoned for the doctor to be quick, quick, quick.
> The doctor came with his bag and his hat
> And he knocked at the door with a rat-a-tat-tat.
> He looked at the dolly and he shook his head
> And he said “Miss Polly, put her straight to bed!”
> He wrote on a paper for a pill, pill, pill
> “I’ll be back in the morning yes I will, will, will.”

For this problem you are to detect and count repetitions of words within each line of verse. For each line print the most common word or "(many)" if more than one word is equally common, followed by the number of repetitions. Let the start of a line, any white space, and the end of a line delineate words. To obtain the words for comparing and counting, first remove all non-alphabetic characters and reduce to all lower case. As an example from the sample input, "I’ll" becomes "Ill", then "ill".

Print the number of repetitions that occurred most frequently for each line. In the sample verse we have eight lines with these counts and words: 3: "sick", 3: "quick", 2: "his", 1: "(many)", 2: "he", 1: "(many)", 3: "pill", and 3: "will".

Input

The first line will contain the number of lines of text following it. This will not excede 100. The remaining lines will each represent one line of text to be processed. No line will be longer than 1000 characters.

Output

For each line, print the word that appears the most often, in all lower case. Then print a space and the number of repetitions of that word. If more than one word occurred with the same maximum frequency, print out "(many)" instead of any of the words for that line.

Sample Input 1	Sample Output 1

`8`
`Miss Polly had a dolly who was sick, sick, sick.`
`So she phoned for the doctor to be quick, quick, quick.`
`The doctor came with his bag and his hat`
`And he knocked at the door with a rat-a-tat-tat.`
`He looked at the dolly and he shook his head`
`And he said "Miss Polly, put her straight to bed!"`
`We wrote on a paper for a pill, pill, pill`
`"I'll be back in the morning yes I will, will, will."`
	`sick 3`
	`quick 3`
	`his 2`
	`(many) 1`
	`he 2`
	`(many) 1`
	`pill 3`
	`will 3`

Problem F
Polly Put the Kettle On

Polly put the kettle on,
Polly put the kettle on,
Polly put the kettle on,
We'll all have tea.

The question before us today is how many full cups of tea can be served, given the capacity of a cup, the capacity of the kettle, the starting temperature of the water, the rate of heat transfer into the kettle of water, the rate of water loss due to evaporation from boiling, and when each cup is poured.

It requires one calorie of heat to raise the temperature of one ml of water one degree Celcius. Once the water reaches 100°Celsius, the boiling point, it will be evaporating rather than continuing to get hotter. Evaporation will occur at a rate of one ml per 540 calories added to the kettle. (You may ignore any evaporation loss other than from boiling.)

A cup of water will be poured from the kettle after a specified time interval, starting when the water reaches a specified temperature and repeated with each succeeding time interval. A single cup will be poured at a time until there is not enough water left to pour a complete cup.

Input

There will be one integer per line, as indicated in the following list.

1. Capacity of a cup (1-1000 ml)
2. Capacity of the kettle (1-10,000 ml)
3. Initial temperature of the water in the kettle (0-100°Celsius)
4. Rate of heat transfer into the kettle (1-1,000,000 calories per minute)
5. Minimum temperature at which the first cup is to be poured (0-100°Celsius)
6. Time interval between attempted cup pours (0-60 minutes) This applies after the first cup is poured.

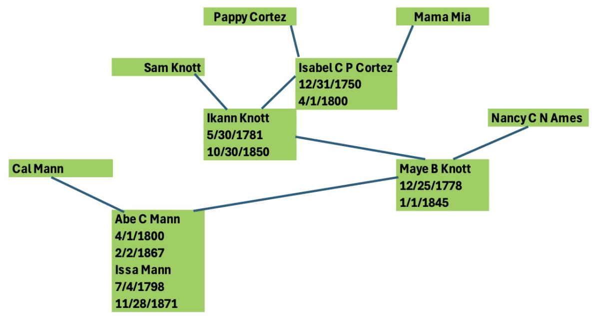
Output

Print out the number of whole cups that will be successfully poured from the kettle.

Sample Input 1	Sample Output 1
100 1000 10 10000 95 2	7

Problem G
Reincarnation

Some cultures hold a belief called reincarnation that after death, one's soul can be reborn into another body, sometimes repeatedly and not necessarily as the same kind of creature. For this problem we will restrict reincarnation to people in a family tree. Given a family tree containing names, parent-child relationships, and birth and death dates, your task is to determine the maximum number of times a particular person could have been reborn after his/her death.



Family tree as specified in the sample

Input

The first line of input will contain the number of people in a family tree. Each subsequent set of 4 lines will contain first the full name of one of the persons, secondly the full name of the father, thirdly the full name of the mother, and fourthly the birth date and death date of the person, each date formatted as month (1-12), day (1-31), and year (0-2025). A final line of input will contain the name of the person for whom we wish to determine the potential number of subsequent rebirths within the family tree. You may assume that a rebirth could take place as soon as the date of death. The order of entry of persons in the family tree is arbitrary.

Output

Print the name of the person in question followed by the potential number of rebirths following that person's death. If the person's death is not given, print 0.

Sample Input 1	Sample Output 1
5 Abe C Mann Cal Mann Maye B Knott 4 1 1800 2 2 1867 Maye B Knott Ikann Knott Nancy C N Ames 12 25 1778 1 1 1845 Ikann Knott Sam Knott Isabel C P Cortez 5 30 1781 10 30 1850 Isabel C P Cortez Pappy Cortez Mama Mia 12 31 1750 4 1 1800 Issa Mann Cal Mann Maye B Knott 7 4 1798 11 29 1871 Isabel C P Cortez	Isabel C P Cortez 1

Sample Input 2	Sample Output 2
5 Abe C Mann Cal Mann Maye B Knott 4 1 1800 2 2 1867 Maye B Knott Ikann Knott Nancy C N Ames 12 25 1778 1 1 1845 Ikann Knott Sam Knott Isabel C P Cortez 5 30 1781 10 30 1850 Isabel C P Cortez Pappy Cortez Mama Mia 12 31 1750 4 1 1800 Issa Mann Cal Mann Maye B Knott 7 4 1798 11 29 1871 Abe C Mann	Abe C Mann 0

Sample Input 3	Sample Output 3
5 Abe C Mann Cal Mann Maye B Knott 4 1 1800 2 2 1867 Maye B Knott Ikann Knott Nancy C N Ames 12 25 1778 1 1 1845 Ikann Knott Sam Knott Isabel C P Cortez 5 30 1781 10 30 1850 Isabel C P Cortez Pappy Cortez Mama Mia 12 31 1750 4 1 1800 Issa Mann Cal Mann Maye B Knott 7 4 1798 11 29 1871 Nancy C N Ames	Nancy C N Ames 0

Problem H
Jack and Jill Count Primes

Jack and Jill went up the hill
To fetch a pail of primes.
Jack fell down and lost his count,
And Jill must count this time.

Jack had a special number in mind, but forgot how many prime numbers came **before** it. Jill, being the clever one, needs your help to figure it out.

A **prime number** is a number greater than 1 that has no positive divisors other than 1 and itself. Jill will give you Jack's number, and you must tell her how many prime numbers are **less than** that number.

Input

You will be given one positive integer n, with n no greater than 10,000. This number will be provided on a single line of input.

Output

Output a single line in the following format:
"Jack forgot there are x prime numbers less than n."
where x is the number of prime numbers strictly less than the given number n.

Sample Input 1	Sample Output 1
`10`	`Jack forgot there are 4 prime numbers less than 10.`

Sample Input 2	Sample Output 2
`2`	`Jack forgot there are 0 prime numbers less than 2.`

This page is intentionally left blank.