icpc | north america contests

The 2023 ICPC Rocky Mountain Regional Contest

Official Problem Set

nac icpc rocky mountain 2023 2024

JET BRAINS

icpc global sponsor programming tools

upsilon pi epsilon honor society

NOVEMBER 4, 2023

Contest Problems

This contest contains 12 problems over 32 pages. Good luck.

For problems that state "*Your answer should have a relative error of less than* 10^{-9}", your answer, x, will be compared to the correct answer, y. If $\frac{|x-y|}{|y|} < 10^{-9}$, then your answer will be considered correct.

For problems that state "*Your answer should have an absolute error of less than* 10^{-9}", your answer, x, will be compared to the correct answer, y. If $|x - y| < 10^{-9}$, then your answer will be considered correct.

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
Attendance
Time limit: 1 second

Taking attendance in your class is a tedious task. You call out the names of students one at a time in alphabetical order. If the student is present, they respond by saying "`Present!`" before you call the next name.

This is such a boring task that you sometimes zone out and don't keep a proper record of attendance. Write a program to help you summarize the absences!

Input

The first line of input contains a single integer N $(1 \leq N \leq 200)$ indicating the number of "callouts". Then N lines follow indicating the callouts that were made in the order they were made. A single line consists of either a student's name or of the response `Present!`. A student's name will consist of between 2 and 10 characters, the first always being an uppercase letter (`'A'-'Z'`) and the remaining characters always being lowercase letters (`'a'-'z'`).

The student names will appear in alphabetical order in this input and a line with the response `Present!` will only appear if the previous line was the name of a student. In particular, the response `Present!` will never appear as the first callout.

Output

Output the names of all students that are absent in the order they were called, each on a separate line. If no students were absent, simply output the message `No Absences`

Sample Input 1

```
6
Buckley
Burnadette
Present!
Chad
Present!
Erin
```

Sample Output 1

```
Buckley
Erin
```

Sample Input 2

```
3
Alice
Bob
Charlie
```

Sample Output 2

```
Alice
Bob
Charlie
```

Sample Input 3	Sample Output 3

`8`
`Gregory`
`Present!`
`Maureen`
`Present!`
`Milton`
`Present!`
`Xavier`
`Present!`
	`No Absences`

Sample Input 4	Sample Output 4

`5`
`Gift`
`Present!`
`Present`
`Treat`
`Present!`
	`Present`

Problem B
Add or Multiply
Time limit: 7 seconds

Your younger sister is playing with a wooden math toy consisting of *number* and *operator* blocks. Each number block is printed with a single digit from 1 to 9, and operator blocks are double-sided; $+$ and $\times$ are printed on each side. She has just built a math expression by alternating number and operator blocks. The first block is a number, the second is an operator, the third is a number, and so on. She asks, "can you answer the value of the expression?"

Initial input of Sample 1.

Further, she repeatedly makes one of the following *moves*.

- `s` i j (swap): Swaps the ith and jth number blocks.
- `f` i (flip): Flips the ith operator block. Its operator alternates between $+$ and $\times$.
- `a` (all flip): Flips all the operator blocks in the entire math expression.

After her move, you have to quickly answer the updated value (possibly the same as the previous one). Remember that you must perform multiplications earlier than additions.

Input

The first line of input contains two integers N ($2 \leq N \leq 2 \times 10^5$) and M ($1 \leq M \leq 2 \times 10^5$), where N is the number of integers and M is the number of queries. The second line represents the initial input of $2N - 1$ characters. Single digits between 1 and 9 are at odd positions, and operators $+$ ("`+`") or $\times$ ("`*`") are at even positions. Each of the next M lines represents your sister's move. The swap move is described as "`s` i j" with $1 \leq i, j \leq N$, where i and j are the 1-indexed positions to swap. If $i = j$, then no blocks are swapped. The flip move is described as "`f` i" with $1 \leq i \leq N - 1$, meaning your sister flips the ith operator. Finally, the all-flip move is described as "`a`", without additional parameters.

Output

The first line of output should be the value of the initial expression. Then, output M more lines, one for each move. Since values can be large, print the value modulo $10^9 + 7$ ($= 1\,000\,000\,007$).

Explanation of Sample 1

- $2+3\times 4=14$: initial expression.
- $3+2\times 4=11$: swaps the first number (2) and the second number (3).
- $3\times 2+4=10$: flips all the operators.
- $3\times 2\times 4=24$: flips the second operator ($+$).
- $3+2+4=9$: again, flips all the operators.

Sample Input 1

```
3 4
2+3*4
s 1 2
a
f 2
a
```

Sample Output 1

```
14
11
10
24
9
```

Problem C
Guess Who
Time limit: 1 second

Guess Who is a two-player board game in which a number of characters are uniquely identified by a set of attributes (e.g. wearing glasses), and each player attempts to guess the other player's hidden character by asking a number of yes/no questions such as "does the person wear glasses?"

In our variation of this game, there are N characters, each of which is uniquely identified by a set of M attributes. The value of each attribute is either YES or NO. You will be provided with a list of the characters and their attributes, together with a series of Q questions and the corresponding responses. Your task is to determine the hidden character, if possible.

Input

Input begins with three space-separated integers N, M, and Q, satisfying $1 \leq N \leq 1\,000$, $1 \leq M \leq 15$, and $1 \leq Q \leq M$. The next N lines each contains a string of M characters that are either `Y` or `N`. The ith line specifies the values of the M attributes of the ith character. Each of the next Q lines contains an integer $1 \leq A \leq M$, followed by a space, followed by a single character `Y` or `N`. This indicates the question is about attribute A, and the response that the attribute of the hidden character is YES or NO. There is at most one question for each of the attribute. At least one of the listed characters has attributes consistent with the responses of the queries.

Output

In the first line, output one of `unique` or `ambiguous`, indicating if there is a uniquely identifiable hidden character, or if there are multiple possible hidden characters.

In the case in which the hidden character is uniquely identifiable, output on the second line the index (between 1 and N) of the hidden character. If there are multiple possible hidden characters, output on the second line the number of possible hidden characters.

Sample Input 1	Sample Output 1
5 5 3 YYYYY NNNNN YNYNY YYYNN NNYYY 1 N 2 N 3 Y	unique 5

Sample Input 2	Sample Output 2
5 5 3 YYYYY NNNNN YNYNY YYNNN NNNYY 1 Y 5 Y 3 Y	ambiguous 2

Problem D
Seven Up
Time limit: 13 seconds

The one-player game of *Seven Up* is played with a standard deck of 52 cards - each card has one of thirteen possible *faces* which we denote by `A`, `2`, `3`, `4`, `5`, `6`, `7`, `8`, `9`, `T`, `J`, `Q`, and `K`. There are exactly four cards of each face.

Initially, seven cards are dealt face down in positions numbered 1 through 7. The ace (`A`) has value 1, the cards with faces `2`–`7` have corresponding values 2 through 7, and the remaining cards do not have any value.

A single turn consists of the player drawing a card from the top of the deck (initially having $52 - 7 = 45$ cards). The following steps are repeated until the turn is **ended**:

- if the card has no value (i.e. the face is not one of `A`, `2`, `3`, `4`, `5`, `6`, or `7`), the turn has **ended**,
- otherwise if the card in the position corresponding to the value of the card held by the player is already face up, the turn has **ended**,
- otherwise the player swaps the card they are holding with the card in the corresponding position except the card they placed in this position is now face up, the current turn continues

At the end of a turn, if all seven positions have a face-up card, the game ends.

If the remaining 45 cards are randomly permuted so each ordering is equally likely, what is the expected number of turns until the game completes?

More specifically, if for any $1 \leq k \leq 45$ we let p_k denote the probability (over the random ordering of the remaining 45 cards) the game finishes after k turns are completed. You should compute $\sum_{k=1}^{45} k \cdot p_k$.

Input

Input consists of a single string consisting of exactly 7 characters from `A23456789TJQK` denoting the faces of the 7 cards that were dealt face down into their corresponding positions. No face will appear more than four times in this string.

Output

Output a single value giving the expected number of turns until the game completes. Your answer should have an absolute or relative error of less than 10^{-6}.

Sample Input 1	Sample Output 1
`A9Q22T5`	`19.88129713423831`

Sample Input 2	Sample Output 2
`JQQQKKK`	`20.91314505643106`

Problem E
Gas Station
Time limit: 1 second

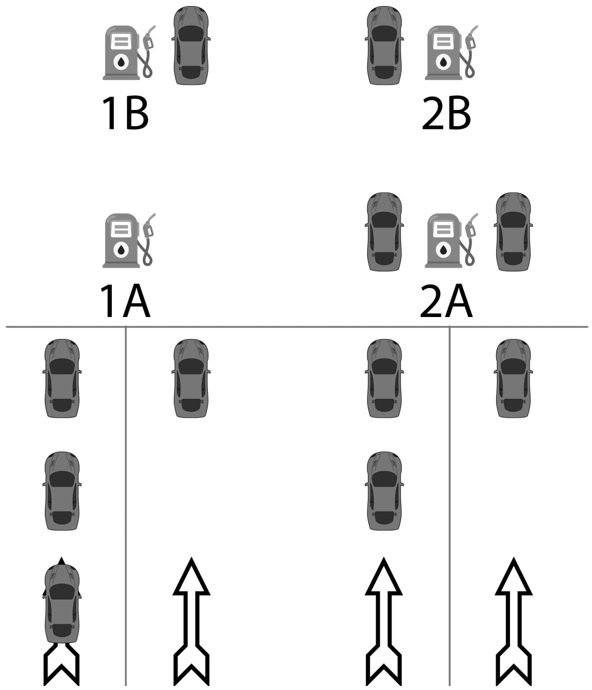
You are a late night attendant at a busy gas station. You can only go home after all the cars have topped up their fuel tanks and left the gas station. The gas station you work at has P gas pump columns, counting from left to right, and each column has two gas pumps, pump A and pump B. A pump can serve cars with left side fuel doors on its right side, or cars with right side fuel doors on its left side. A pump can serve a car on its left side and a car on its right side simultaneously. Thus, each pump column has two lanes for cars.

Cars arriving at the gas station follow strict rules. A car will go to the leftmost open lane that is suitable for its fuel door side. If there are no open lanes, the car will queue up for the suitable lane with the shortest queue. If there are multiple lanes with the shortest queue, the car will queue up in the leftmost one. Once a car has joined a queue, it cannot switch to a different one. After cars leave after refueling and a lane becomes open, the next car in the queue for that lane will go to use a pump.

A lane is open if pump A is available. If pump B is available but pump A is occupied, the lane is not open. When a car goes to an open lane, if both pumps are available, the car will go to pump B, otherwise it will go to pump A. If a car arrives at the same time as some other cars have just finished filling up and left, the new car waits for all other cars in the queues to move to the open pumps (if any) before deciding where to queue. When a car leaves from a pump A, but the pump B ahead of it is occupied, the car is stil able to leave immediately.

You know that Car i arrives at time t_i, requires just under f_i minutes to fill up and leave, and has its fuel door on the s_i side.

Knowing all this, you want to know when you will be able to go home.



Input

The first line of input contains two space separated integers, $1 \leq P \leq 10$, the number of gas pump columns, and $1 \leq N \leq 1000$, the total number of cars that will arrive.

The next N lines each contains two integers $1 \leq t_i \leq 10^5$, $1 \leq f_i \leq 100$, and a single character `R` or `L` for s_i, all separated by a space.

It is guaranteed that $t_i < t_{i+1}$ for all $1 \leq i < N$.

Output

Output N lines containing the time when each car leaves the gas station, in the order that the cars are listed in the input.

Sample Input 1	Sample Output 1
1 4 1 9 L 2 5 L 3 10 L 4 10 L	10 7 17 27

Sample Input 2

```
1 4
1 9 L
2 9 L
3 10 L
4 10 L
```

Sample Output 2

```
10
11
21
21
```

Sample Input 3

```
1 2
8 11 R
9 10 L
```

Sample Output 3

```
19
19
```

Sample Input 4

```
2 10
1 10 R
2 3 R
3 10 R
4 12 R
5 1 R
6 5 R
7 10 R
10 2 R
11 7 R
13 4 R
```

Sample Output 4

```
11
5
13
16
6
11
21
18
18
22
```

Problem F
Thwack!
Time limit: 5 seconds

The game *Thwack* is played on a 1-dimensional grid of cells: the *game board*. Each grid cell contains either a black stone, a white stone, or is empty.

Two players alternate taking turns. A turn consists of choosing two adjacent stones of opposite colours and then choosing one of these stones to *capture* the other. That is, one stone eliminates the other stone by moving to its position. A player loses when it is their turn to play but there are no available moves, i.e. there is no pair of adjacent stones of different colours.

The interesting thing about Thwack is that there is no default "initial setup" of the game board. Any game can be made by simply placing stones randomly on the grid.

Given the initial configuration of a game, your job is to list all the opening moves that would be winning moves for the first player assuming both players play optimally.

Input

The first line of input contains a single integer N $(1 \leq N \leq 100)$ indicating the number of cells on the game board.

The second line contains a single string of length N consisting only of characters `B`, `W`, or `.` which indicate the initial contents of the cells on the game board at the start of the game.

Output

The first line of output displays the number W of possible opening moves for the first player that would result in them winning, assuming both players play perfectly. Then W lines follow, the ith of which contains two integers A_i, D_i $(1 \leq A_i, D_i \leq N, |A_i - D_i| = 1)$ indicating that moving the stone at A_i to capture the stone at position D_i is one possible opening move that would result in the first player winning. These lines should be presented in lexicographical ordering, i.e. for $1 \leq i < N$ we have either a) $A_i < A_{i+1}$ or b) $A_i = A_{i+1}$ and $D_i < D_{i+1}$.

Sample Input 1	Sample Output 1
4 BWBW	2 2 3 3 2

Sample Input 2	Sample Output 2
11 .BW.WB..W..	0

Sample Input 3	Sample Output 3
8 BBBBW.BW	1 5 4

Sample Input 4	Sample Output 4
9 BBBBBW.BW	0

Sample Input 5	Sample Output 5
12 WBBWBWWBBWBW	5 2 1 4 5 5 4 10 11 11 10

Sample Input 6	Sample Output 6
11 WBBWWWBBWBW	5 1 2 4 3 9 10 10 9 11 10

Problem G
Bombardment
Time limit: 7 seconds

You are designing an old-school game you call *Bombardment* where the goal is to destroy a number of points by bombarding them. You do not yet know the theme of your game, just that the core mechanics should involve a bombardment.

The points to be destroyed are located on the real number line, that is each point is simply an x-coordinate. A bombardment is an attack that will destroy all points within some fixed distance R from the center of the bombardment. More specifically, a single bombardment is specified by picking an integer point X (the center). All points lying in the interval $[X - R, X + R]$ will be destroyed.

You decide to playtest a basic version of this game before you go through the effort of designing a theme, adding nice graphics, etc. Interestingly, most testers seemed to employ a greedy strategy: each bombardment is chosen to destroy the maximum number of points in that single bombardment. Sometimes, this causes players to use more bombardments than the minimum possible number of bombardments. You want to design a program that will simulate this strategy, this will help you design interesting levels.

That is, your job is to write a program that will simulate the following process. While there are still points remaining, choose a value X for a bombardment that will destroy the maximum number of remaining points. If there are multiple values X for the center of such a bombardment, you will choose the least such X each time.

Input

The first line of input contains two integers N ($1 \leq N \leq 5 \times 10^5$) and R ($1 \leq R \leq 10^8$). The next line contains N integers describing the x-coordinates of the points, each lying in the range $[-10^8, 10^8]$. Multiple points may share the same x-coordinate. You are also guaranteed that the difference between the maximum x-coordinate and the minimum x-coordinate is at most $40 \cdot R$.

Output

The first line of output contains a single integer B indicating the number of bombardments that were performed. The second line consists of B integers describing the x-coordinates of each bombardment in the order they were performed.

Sample Input 1	Sample Output 1
7 1 1 2 3 3 4 4 5	3 3 0 4

Sample Input 2	Sample Output 2
6 1 5 -2 5 0 1 2	3 1 4 -3

Sample Input 3	Sample Output 3
6 2 5 -2 5 0 1 2	2 0 3

Sample Input 4	Sample Output 4
6 3 5 -2 5 0 1 2	2 2 -5

Problem H
Scientific Grading
Time limit: 1 second

You recently started working as a TA (teaching assistant) for your university's *Scientific Computing* class. Today, Professor introduced the *scientific notation*, where numbers are written in the form $m \times 10^n$ with a real number m (the *significand*) and an integer n (the *exponent*). At the end of class, she gave students the following assignment.

Given two numbers x, y in scientific notation, perform the following four arithmetic operations:

- $x + y$
- $x - y$
- $x \times y$
- x/y

As a strict grader, you decided to write a program to grade students' answers. You mark a solution correct if and only if *both* relative and absolute errors are *less than* 10^{-9} (not including 10^{-9}). If the correct answer is 0, then 0 is the only acceptable answer. Otherwise, a student's answer z will be compared to the correct answer $\tilde{z}$, and the relative and absolute errors are computed as $\frac{|z-\tilde{z}|}{|\tilde{z}|}$ and $|z - \tilde{z}|$, respectively.

Input

The first line of input contains the value of x, and the second line contains the value of y. The next four lines contain a student's answer to $x + y$, $x - y$, $x \times y$, and x/y. All numbers are in the form of `<SIGNIFICAND>e<EXPONENT>`. The significand m starts with a sign ($+$ or $-$), followed by one digit, a period (`.`), and *exactly* nine digits. The exponent n also starts with a sign ($+$ or $-$) and is followed by an integer between 0 and 10^9, inclusively. The value is computed by $m \times 10^n$. The value 0 is always represented as `+0.000000000e+0`, and for any nonzero values the first digit of their significand is not 0. It is guaranteed that x and y are both nonzero.

Output

For each student solution, output `Correct` if it is considered correct and `Incorrect` otherwise. The first line of output indicates if the student's solution to $x + y$ is correct, the second line indicates if their solution to $x - y$ is correct, the third line indicates if their solution to $x \times y$ is correct, and the fourth line indicates if their solution to x/y is correct.

Sample Input 1	Sample Output 1
+2.000000000e+1 +3.000000000e+2 +3.200000000e+2 -2.800000000e+2 +6.000000000e+3 +6.666666667e-2	Correct Correct Correct Correct

Sample Input 2	Sample Output 2
+1.000000000e-1 +1.000000000e-1 +2.000000003e-1 +1.000000000e-18 +1.000000002e-2 +1.000000001e+0	Incorrect Incorrect Incorrect Incorrect

Problem I
Building Roads
Time limit: 2 seconds

A multi-billionaire has a vision to build a completely new city from scratch. After much research and consultations, locations have been selected for all the houses, shopping malls, restaurants, etc. Roads now have to be added to ensure that every location is reachable by any other location, but only the minimum number of roads should be built. For environmental reasons, it is also desirable to minimize the longest travel between two locations. Each road must connect two locations, but roads may cross each other by adding overpasses (so traffic cannot switch to a different road between locations).

What is the minimum length in the longest travel in the road network designed?

Input

The first line of input contains an integer $2 \leq N \leq 200$ specifying the number of locations to follow. Each of the next N lines contains two integers x_i and y_i ($-5000 \leq x_i, y_i \leq 5000$), specifying the coordinates of the ith location. All coordinates are specified in meters, and all locations are distinct.

Output

Output the minimum possible length (in meters) of the longest travel between two locations. Your answer should have a relative or absolute error of less than 10^{-3}.

Sample Input 1

```
3
0 0
10 0
0 10
```

Sample Output 1

```
20.0000000000
```

Sample Input 2

```
9
0 0
10 0
0 10
-10 0
0 -10
10 10
10 -10
-10 10
-10 -10
```

Sample Output 2

```
28.2842712475
```

This page is intentionally left blank.

Problem J
Don't Be Fake
Time limit: 1 second

DontBeFake is an app where people get a notification at a random point in time, and they have to take a picture of what they are doing right now.

You have N friends on DontBeFake. Each friend has a set of intervals of seconds $[L, R]$ when they are available. If they get a notification at second s and they are available, they will take a picture and you will get to see it. Because the DontBeFake app is slow and so is taking a picture, even if they are available a second later, they will not have time to take the picture. All friends live in the same time zone and will get one notification at exactly the same time in the day. All times are specified as seconds from midnight.

You want to know what is the maximum number of pictures you can view in the day, over all possible seconds that the notification can arrive. In addition, you also want to know how many different seconds the notification could arrive for the maximum number of pictures to be taken.

Input

Input begins with a line containing the integer N ($1 \leq N \leq 50$). The next N lines each describes the set of available intervals for the N friends. Each such line starts with an integer M ($1 \leq M \leq 10$) followed by M pairs of integers L_i, R_i ($L_i \leq R_i$) meaning that the friend is available between L_i and R_i seconds, inclusive. It is guaranteed that $R_i < L_{i+1}$ for all $1 \leq i < M$, so that the intervals do not overlap. It is also guaranteed that $0 \leq L_i, R_i < 86\,400$.

Output

Output on the first line the maximum number of pictures you can view in the day. On the second line, output the total number of seconds the notification could arrive for the maximum number of pictures to be taken.

Sample Input 1	Sample Output 1
3 1 0 20000 2 10000 20000 40000 60000 1 15000 80000	3 5001

Sample Input 2	Sample Output 2
3 1 0 10000 2 10000 20000 30000 40000 2 20000 30000 50000 80000	2 3

This page is intentionally left blank.

Problem K
A Complex Problem
Time limit: 3 seconds

There are many problems in the field of computer science, and some are harder than others. Computer scientists have accordingly categorized problems using complexity classes, and like to analyze these classes to see how they interact with each other.

A complexity class is a simply a set of problems; for example, the complexity class P is the set of decision problems which are solvable with an algorithm whose runtime scales as a polynomial function of the input size. We know some results about the relations between complexity classes: for example, every problem in the complexity class P also belongs to the complexity class NP of decision problems for which "yes"-instances can be verified in polynomial time. However, we don't know if every problem in NP is also in P (and we won't ask you to figure this out for today). Therefore, P and NP could be either one or two distinct complexity classes. On the other hand, we know that the Halting Problem is in ALL (the set of all decision problems) but not in R (the set of decision problems solvable by a Turing machine). Therefore, ALL and R are two distinct complexity classes.

Given a series of relations between complexity classes, can you find the minimum and maximum number of distinct complexity classes? Two complexity classes are distinct if and only if there is some problem that exists in one class but not in the other.

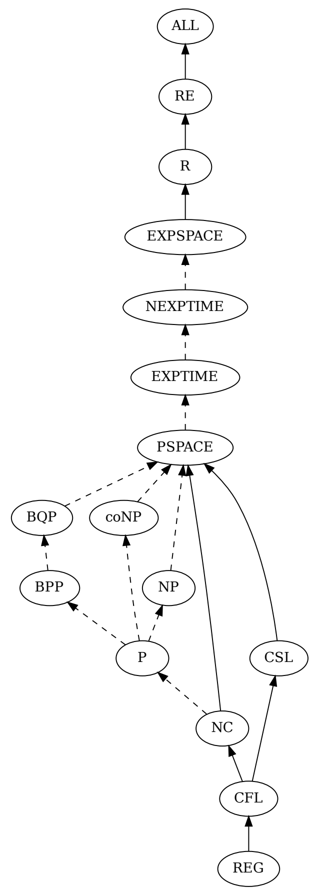


Illustration of Sample 4, where dashed lines indicate subset relations ($\subseteq$) and solid lines indicate proper subset relations ($\subsetneq$)

Input

Input begins with two space-separated integers $0 \leq M, N \leq 10^5$ such that $M + N > 0$. Each of the next M lines consists of two distinct space-separated complexity classes A_i and B_i, where a complexity class is a set of problems, denoted by one to eight uppercase or lowercase English letters. Each of these M lines indicates that $A_i \subseteq B_i$, meaning that every problem in A_i is also in B_i. Similarly, each of the next N lines consists of two distinct space-separated complexity classes A_j and B_j. Each of these N lines indicates that $A_j \subsetneq B_j$, meaning that every problem in A_j is also in B_j **and** that at least one problem in B_j is not in A_j.

There is at most one relation between any two distinct complexity classes specified in the input, and the relations between complexity classes will not imply any logical contradiction.

Output

Output two space-separated integers on a single line: the minimum and maximum number of distinct complexity classes among the specified complexity classes given in the relations in the input.

Sample Input 1	Sample Output 1
1 0 P NP	1 2

Sample Input 2	Sample Output 2
0 1 R ALL	2 2

Sample Input 3	Sample Output 3
3 0 QMIP MIPstar MIPstar RE RE QMIP	1 1

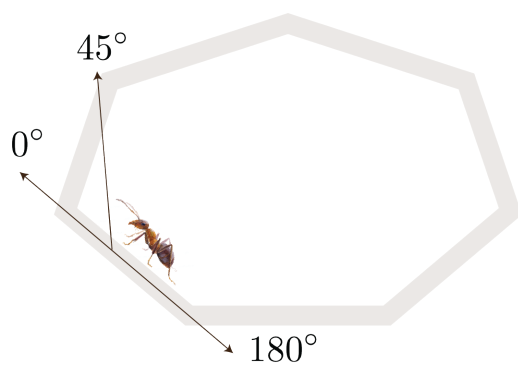
Sample Input 4	Sample Output 4
11 8 NC P P BPP P coNP P NP BPP BQP coNP PSPACE BQP PSPACE NP PSPACE PSPACE EXPTIME EXPTIME NEXPTIME NEXPTIME EXPSPACE REG CFL CFL NC CFL CSL NC PSPACE CSL PSPACE EXPSPACE R R RE RE ALL	7 16

Problem L
Ribbon Road
Time limit: 1 second

Christie is an ant that likes to crawl on a closed ribbon (i.e., it forms a polygon with start and end points attached). Her owner, Cindy would place a sugar cube on the ribbon for her to enjoy. Cindy always puts the sugar cube on the exterior side of the ribbon, so Christie wants to crawl on the exterior side in order to expect a sugar cube somewhere down the ribbon road. However, because Christie is so tiny, she usually has trouble finding out if she is crawling on the inside or on the outside of the ribbon. If she is on the inside, then she would need to crawl to the outside carefully.

Fortunately, Christie's antenna can emit a signal, which is a ray. The rays emitted by Christie travel from her location to some other point (x, y), and the ray may pass through the ribbon. The rays she emits span $[0°, 180°]$ with the line segment she is current on—the direction parallel to the line segment she stands on pointing forward being $0°$, the span of the ray goes above her, and direction parallel to herself pointing away from her being $180°$. The ribbon does not twist (i.e., the ribbon does not rotate about itself), so that the ribbon is not a Möbius band. Therefore, whether Christie is on the inside or the outside is always the same as she walks along the ribbon.

The ray can be used to determine if Christie is crawling on the inside or the outside of the ribbon. Note that if the ray happens to be parallel to the line segment she stands, then she will not be able to determine if she is inside or outside. Also, if she stands on a vertex of the polygon, then she *might* not be to determine the answer either. Please help Christie find out if she is on the inside of the ribbon road.



An illustration of Christie's antenna range, from which you can see if antenna is emanating $0°$ or $180°$, then Christie would not be able to tell she is inside or outside.

Input

The first line of input contains a single integer N ($3 \le N \le 10^5$), which is the number of vertices in the polygon. The next N lines each contains a pair of integers (x_i, y_i), indicating the ith vertex of the polygon in order ($1 \le i \le N$). The ribbon road is a closed simple polygon so the last vertex is connected back to the first vertex. No three consecutive vertices are colinear. The final line contains four integers (x_c, y_c)

and (x, y), denoting the location of Christie and a point that Christie's signal passes through, respectively. Christie always stays on the ribbon in a direction parallel to the ribbon, but the location of (x, y) has no restrictions. If Christie stays on a vertex of the ribbon, then Christie can be parallel to either edge adjacent to that vertex. It is guaranteed that $(x_c, y_c) \neq (x, y)$.

All coordinates satisfy $-10^6 \leq x_i, y_i, x_c, y_c, x, y \leq 10^6$.

Output

Output on a single line `inside` if Christie is inside, `outside` if Christie is outside of the ribbon road, or `?` if it cannot be determined.

Sample Input 1	Sample Output 1
4 0 0 2 0 2 2 0 2 1 0 1 1	inside

Sample Input 2	Sample Output 2
4 0 0 0 2 2 2 2 0 1 0 -1 0	?

Sample Input 3	Sample Output 3
4 0 0 5 0 5 5 0 5 4 0 1 0	?

Sample Input 4	Sample Output 4
4 0 0 2 0 2 2 0 2 0 0 1 1	inside

Sample Input 5	Sample Output 5
4 0 0 2 0 2 2 0 2 0 0 -1 -1	outside

Sample Input 6	Sample Output 6
4 0 0 1 0 1 1 0 1 0 0 1 -1	?