ICPC SOUTH PACIFIC REGIONAL FINALS

DECEMBER 17, 2022

Contest Problems

A: A Menace 2 $society
B: Begrudging Friendship 2
C: Cakes and Ogres
D: Distinct Platforms
E: Emergency Plan
F: Finding Clusters
G: Great Trails
H: Holy Stone of Destiny
I : Iguana Honeymoon
J : Jay and Jay's son ja$on
K: Keeping Arrays Menacing
L: Land Tax

This contest contains twelve problems. Good luck.

For problems that state "*Your answer should have an absolute or relative error of less than* 10^{-9}", your answer, x, will be compared to the correct answer, y. If $|x-y| < 10^{-9}$ or $\frac{|x-y|}{|y|} < 10^{-9}$, 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 ($a, b, \ldots, z$).

Definition 4

Unless otherwise specified, the distance between two points (x_0, y_0) and (x_1, y_1) is defined as its Euclidean distance:

$$\sqrt{(x_0 - x_1)^2 + (y_0 - y_1)^2}.$$

Problem A
A Menace 2 $society
Time limit: 3 seconds

Jay and Jay's son ja$on are playing a game together. This game involves counting occurrences of a string J in another string S. But, ja$on has gotten so good at this game, Jay is at a loss for how to keep it interesting.

Jay is a menace to society and has decided to make the game more fun for Jay's son ja$on by adding some new rules. On top of S, Jay will define $K + 1$ more strings $S_0, S_1, \ldots, S_K$. S_i can be created by taking the first character in S, skipping the next i characters, then taking the next character, then skipping the next i characters, and so on. For example, if S is `southpacific`, then S_2 is `staf`. Note that S_0 and S are the same string.

Further, in honour of Jay's son ja$on's name, Jay will use letters from a language called 'Engli$h', which contains all lowercase English letters, except that the letter `s` is replaced with the letter `$`. So `$outh`, `pacific`, `programming`, `conte$t` are valid words in Engli$h, but `south` and `contest` are not.

In Jay's new game, he must find the string J in each of the $K + 1$ words. For example, if S is `aacbc` and J is `ac`, then S_1 is `acc`, and S_2 is `ab`. J is found once in both S_0 and S_1, but not S_2.

Given the strings S and J, as well as the value of K, determine the number of occurrences of J in each S_i.

Input

The first line of the input contains a single integer K ($0 \leq K < 1\,000\,000$), which is the number of strings to check.

The second line of the input contains a single non-empty string S that has strictly more than K and at most $1\,000\,000$ characters, consisting of Engli$h letters.

The third line of the input contains a single non-empty string J, consisting of Engli$h letters. The length of J is at most the length of S.

Output

Display the number of occurrences of J in $S_0, S_1, \ldots, S_K$.

Sample Input 1

```
3
ca$hca$h
cc
```

Sample Output 1

```
0
0
0
1
```

Sample Input 2

```
0
abc
abc
```

Sample Output 2

```
1
```

Sample Input 3

```
2
aacbc
ac
```

Sample Output 3

```
1
1
0
```

Sample Input 4	Sample Output 4
3 ff$f $f	3 0 1 0

Problem B
Begrudging Friendship 2
Time limit: 4 seconds

In Divisionals, we learned about Kyle and Leonard's pseudo-friendship. Today, they are out again at a restaurant where they are sharing a meal, which consists of N food items.

Leonard noticed that Kyle bought him a really bad drink last time they were out and would like to return the favour. Every item of food in the meal has a tastiness value. Leonard will give some of the food items to Kyle and the rest to himself. Both people must receive at least one food item, and no food item can be eaten by both of them. The total tastiness value of a person's food items is the sum of all of their food items' tastiness values.

If the total tastiness value of Leonard's food items is ℓ and the total tastiness value of Kyle's food items is k, then Leonard's *pettiness* value is $\ell - k$. Leonard will divide the food so that his pettiness value is maximized.

Given the tastiness values of all the food items, what is the largest pettiness value that Leonard can achieve?

Input

The first line of input contains a single integer N ($2 \leq N \leq 300\,000$), which is the number of food items.

The next line contains N integers $t_1, t_2, \ldots, t_N$ ($-10^9 \leq t_i \leq 10^9$), which are the tastiness values of the food items.

Output

Display the largest pettiness value that Leonard can achieve.

Sample Input 1	Sample Output 1
4 10 15 5 7	27

Sample Input 2	Sample Output 2
2 10 5	5

Sample Input 3	Sample Output 3
4 1 -3 2 1	7

This page is intentionally left (almost) blank.

Problem C
Cakes and Ogres
Time limit: 4 seconds

Cakes and Ogres are alike in one important way: They both have layers.

Fiona is planning her wedding and is trying to assemble a delicious cake out of a collection of possible cake layers. Each layer has a weight, a carrying capacity, and a deliciousness. Layers can be stacked on top of one another in any order to create the cake. The cake will collapse if the carrying capacity of a layer is lower than the sum of the weight of that layer plus all the weights of the layers above that layer. That is, a layer must carry itself and all the layers above it.

Fiona wants to maximise the sum of deliciousness values for the layers in the cake such that the cake does not collapse.

Input

The first line of input contains the integer N ($1 \leq N \leq 10$), the number of layers.

The next N lines describe the layers. The ith of these lines contains three integers W_i ($1 \leq W_i \leq 10^9$), C_i ($W \leq C_i \leq 10^9$), and D_i ($1 \leq D_i \leq 10^9$), which are the weight, carrying capacity, and deliciousness of the ith layer respectively.

Output

Display the maximum sum of deliciousness values.

Sample Input 1

```
3
3 4 5
1 9 2
1 1 1
```

Sample Output 1

```
8
```

Sample Input 2

```
2
2 2 2
3 3 3
```

Sample Output 2

```
3
```

Sample Input 3

```
3
1 2 2
9 10 2
1 1 1
```

Sample Output 3

```
4
```

This page is intentionally left (almost) blank.

Problem D
Distinct Platforms
Time limit: 10 seconds

You are playing a new platforming game on a 2D grid. There are N platforms numbered from 1 to N spread out in the world. Platform i covers the range from L_i to R_i. You can start at any platform you choose and then must jump between platforms. You can jump from platform a to platform b if their ranges overlap. That is, there is some integer x such that $L_a \leq x \leq R_a$ and $L_b \leq x \leq R_b$.

Once you visit a platform (by either starting there or jumping to it), you can never jump onto that platform again. You win the game by visiting all of the platforms.

Find out if it is possible to win the game. If it is, show a way to visit all of the platforms.

Input

The first line of input contains a single integer N ($1 \leq N \leq 500\,000$), which is the number of platforms.

The next N lines describe the platforms. Each of these lines contains two integers L_i ($1 \leq L_i \leq 10^9$) and R_i ($L_i \leq R_i \leq 10^9$). This represents that platform i covers the range from L_i to R_i.

Output

If it's not possible to visit each platform exactly once, display `NO`. Otherwise, display `YES` followed by N integers representing the order to visit each platform. If there are multiple solutions, any will be accepted.

Sample Input 1

```
5
4 14
5 20
10 16
1 5
19 23
```

Sample Output 1

```
YES
4 1 3 2 5
```

Sample Input 2

```
3
1 10
12 20
1 10
```

Sample Output 2

```
NO
```

This page is intentionally left (almost) blank.

Problem E
Emergency Plan
Time limit: 10 seconds

The town of Bloomsville is always prepared for an emergency. The houses in Bloomsville are arranged in a single straight line running west to east, and numbered from 1 to n, inclusive, in that order. There is exactly one person per house. In case of a fire at a house, the occupant of that house is allowed to seek shelter at any house that is strictly west of their house. Since the westernmost house has no houses to its west, they are given a special firehose to put out the fire. Every other person must decide where they will seek shelter in case of an emergency.

Each house has an emergency level, and each person starts with an initial emergency level equal to their house's emergency level. They will seek shelter at a house with a similar emergency level to their own emergency level. However, the emergency level of a person decreases as they walk. For every house that a person passes, their emergency level decreases by one.

For example, consider 3 houses with emergency levels of $[3, 6, 5]$ (listed from west to east). The third house has an emergency level of 5, so the person there will initially have an emergency level of 5. If they take shelter in the middle house, their emergency level would be 4. If they go to the westernmost house, it would be 3.

Each person will choose the house that has the closest emergency level to theirs when they arrive at that house (they will only ever walk west). In the example above, the person in the third house will choose the first house since $|3-3| < |4-6|$. If there are two houses that provide the same absolute difference, they will choose the one where they walk the least.

Given the emergency levels of the houses, determine which house each of the people (except the westernmost) will go to in an emergency.

Input

The first line of input contains a single integer n ($2 \leq n \leq 300\,000$), which is the number of houses.

The next line contains n integers $e_1, e_2, \ldots, e_n$ ($0 \leq e_i \leq 10^9$), which are the emergency levels of the houses (which is also the initial emergency level of the person in each of them) in order from west to east.

Output

Display which houses (by number) each of the $n-1$ people (not including the person initially in the westernmost house) will go to in an emergency, in order from west to east of each person's initial location.

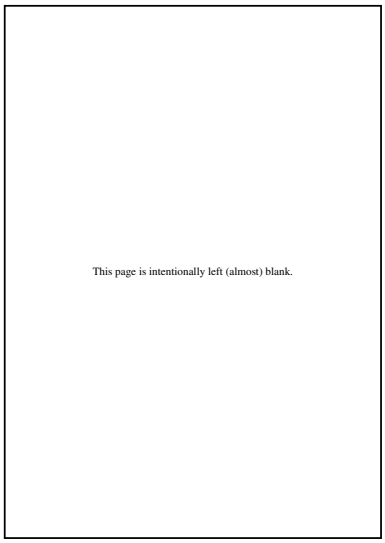
Sample Input 1	Sample Output 1
3 3 6 5	1 1

Sample Input 2	Sample Output 2
11 3 1 4 1 5 9 2 6 5 3 5	1 1 2 3 1 4 4 7 7 10

Sample Input 3	Sample Output 3
2 100 0	1

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

Problem F
Finding Clusters
Time limit: 6 seconds



You just came up with the brilliant PageBlank algorithm, which drastically improves the quality of web search results. To avoid manipulation, it needs to detect and analyse artificial, highly interlinked clusters of web pages.

The web is modelled as a set of n web pages with links between them. A non-empty set S of web pages is a *cluster* if for every page p in the web (not just for the web pages in S), p is in the set S if and only if there is a page q in the set S such that q contains a link to p.

Two clusters can be compared to determine which is more suspicious. Given two clusters S and T, let p be the smallest identifier in exactly one of S and T. If p is in S but not in T, then T is more suspicious; otherwise, S is more suspicious.

Out of all clusters, which is the most suspicious?

Input

The first line of the input contains a single integer n ($1 \leq n \leq 10^5$), which is the number of web pages.

Each of the next n lines describes the links contained in a web page. Each line starts with the number of links m_i ($1 \leq m_i \leq 10^6$), followed by m_i distinct integers a_{ij} in ascending order, which are the linked web pages ($1 \leq a_{ij} \leq n$). Web pages are numbered from 1 to n in the order they are described in the input. The total number of links is at most 10^6.

Output

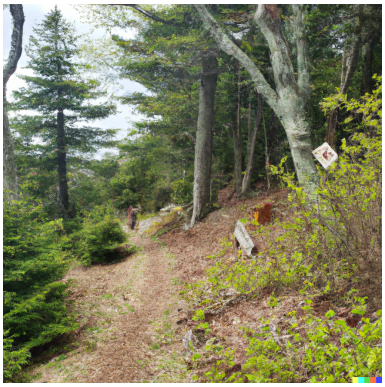
Display the web pages in the most suspicious cluster. List these pages using their identifiers in ascending order. It can be proven that at least one cluster exists given the constraints for the problem.

Sample Input 1	Sample Output 1
3 1 1 1 2 1 3	3

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

This page is intentionally left (almost) blank.

Problem G
Great Trails
Time limit: 10 seconds



Alex is in charge of planning the walking trails in a nature reserve. The nature reserve comprises M trails between N intersection points. The intersection points are labelled with unique integers from the inclusive range between 1 and N. Each trail runs from one intersection point to another intersection point and can only be traversed in one direction to prevent hikers from bumping into one another. There are no two trails between the same pair of intersection points going in the same direction, though there may be one in each direction. No trail can start and end at the same intersection point. The length of each trail is a positive number of metres.

A *tour* of the nature reserve is a sequence of intersection points that are connected by trails that go from intersection point 1 to intersection point N. Hikers always take the shortest tour they can. If there are many shortest tours, they pick one arbitrarily. If there are no tours, the hikers go home. Two tours are considered different if the sequences of intersection points they visit are different.

Alex needs to close one of the trails to save on costs. Hikers will complain if their tours change, so Alex must pick carefully. Let the length of the shortest tour be L. If there is no tour, define L to be infinitely large. For each edge individually, Alex wants to determine the number of distinct tours whose length is equal to L after that edge is deleted. Note that if there are no tours initially (L is infinitely large), then there are always zero tours whose length is equal to L after deleting an edge. Note that edge deletions are independent and should not be applied cumulatively.

Input

The first line of input contains two integers N ($2 \leq N \leq 200\,000$) and M ($1 \leq M \leq \min(200\,000, N \cdot (N-1))$), which are the number of intersection points and trails respectively. The next M lines describe the trails. The ith of which contains three integers u_i ($1 \leq u_i \leq N$), v_i ($1 \leq v_i \leq N$ and $u_i \neq v_i$), and l_i ($1 \leq l_i \leq 10^9$), the start intersection point, the end intersection point, and the length respectively.

Output

For each edge in input order, display the number of tours of length L there are after deleting that edge. Since the number of tours can be large, display them modulo $1\,000\,000\,007$.

Sample Input 1	Sample Output 1
5 6 1 2 1 2 3 2 1 4 1 4 5 2 1 5 66 4 2 1	1 1 0 0 1 1

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

Sample Input 3	Sample Output 3
3 1 1 2 1	0

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

Problem H
Holy Stone of Destiny
Time limit: 4 seconds

It is time: the alignment of all eight planets (yes, eight—sorry, Pluto) is approaching. The three stones of destiny must be put in place at the time of the alignment.

The first two stones have already been placed at integer coordinates. The third stone, named the *holy stone*, must be placed at another point with integer coordinates to finish the triangle. The triangle may not be a straight line (it must have a positive area). The holy stone will activate only if it is at a point where the angle inside the triangle (formed at the holy stone) is as large as possible among all integer points.

Find a point where the holy stone would activate.

Input

The input starts with an integer T ($1 \leq T \leq 10\,000$), which is the number of test cases.

The next T lines describe the test cases. Each line contains four integers x_1, y_1, x_2, y_2 ($-10^9 \leq x_1, y_1, x_2, y_2 \leq 10^9$), indicating that the first two stones are at locations (x_1, y_1) and (x_2, y_2). It is guaranteed that $(x_1, y_1) \neq (x_2, y_2)$.

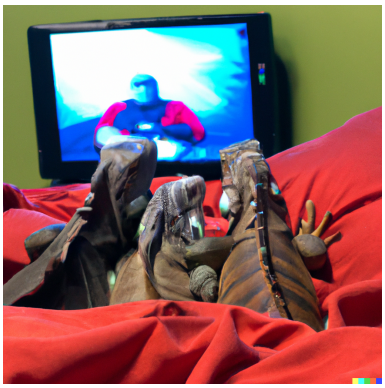
Output

For each test case, display the $x-$ and $y-$coordinate of a point with integer coordinates where the holy stone would activate. If there are multiple solutions, any will be accepted.

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

This page is intentionally left (almost) blank.

Problem I
Iguana Honeymoon
Time limit: 2 seconds



The legendary iguanas Izzy and Iggy just got married and are heading to their honeymoon. The newlyweds are working on picking a movie to 'watch' during their honeymoon. Izzy really likes movies about cats and/or pirates. Iggy really likes movies about pirates and/or skeletons. As such, during each scene of a movie, Izzy is interested if the scene is about cats or pirates and is very interested if the scene is about cats and pirates. Similarly, Iggy is interested if the scene is about pirates or skeletons and is very interested if the scene is about both.

Izzy and Iggy enjoy watching movies together but prefer scenes where they are both equally interested (either both very interested, both interested, or both not interested).

Can you help Izzy and Iggy pick the movie that has the most scenes where Izzy and Iggy would be equally interested?

Input

The first line of input contains a single integer N ($1 \leq N \leq 100$), which is the number of movies Izzy and Iggy are considering.

Then, N movie descriptions follow. Each movie description starts with a line containing a string of uppercase letters T_i ($1 \leq |T_i| \leq 100$), which is the title of movie i, and an integer M_i ($1 \leq M \leq 100$), which is the number of scenes in the movie. The following M_i lines describe the scenes in movie i. Each of these lines contains three strings that are either `Y` or `N` (shortened versions of 'yes' and 'no'). The first string says whether or not this scene is about cats, the second says if the scene is about pirates, and the third says if the scene is about skeletons.

No two movies have the same title.

Output

Display the title of the movie that has the most scenes where Izzy and Iggy would be equally interested. If there are multiple such movies, choose the title that comes lexicographically first.

Sample Input 1	Sample Output 1
3 CATSOFTHECARIBBEAN 3 Y N N Y Y N N N N FORESTCATS 2 Y N N Y N N FINDINGSKELETONS 3 Y N Y N Y Y N N Y	CATSOFTHECARIBBEAN

This page is intentionally left (almost) blank.

Problem J

Jay and Jay's son ja$on

Time limit: 15 seconds

Jay and Jay's son ja$on live in Rome. As we all know, all roads lead to Rome. Every city in the country (other than Rome) has exactly one outgoing road. No matter which city you start in, you can repeatedly take the outgoing roads and eventually arrive back in Rome.

Jay and ja$on love road trips and scrapbooks. Each year, they take a road trip. They start by picking a city that they have never visited before at any point on any previous road trip (this city cannot be Rome). Their road trip plan is simple for each year: continually take the roads visiting cities along the way until they arrive back in Rome. It is possible they will visit cities that they have visited in previous years, but they may not start the road trip at one of these places.

They alternate who selects the starting city each year (with Jay selecting the city the first year). After each road trip, the person who chose the starting city makes a scrapbook of their road trip. They then brag about it until the next road trip.

Because of this, both Jay and ja$on want to make the last scrapbook ever so that they can brag about it for the rest of their lives. Both Jay and ja$on will optimally choose their cities each year so they can be the forever-bragger.

Given a description of the roads, determine whether Jay or ja$on gets to brag for the rest of their lives.

Input

The first line of the input contains a single integer N ($1 \leq N \leq 200\,000$), which is the number of cities in the country other than Rome. The cities are numbered 1 to N, and Rome is number 0.

The next N lines describe the outgoing roads. The ith line (1-based) contains a single integer P_i ($0 \leq P_i < i$) denoting the outgoing road from city i goes to city P_i.

Output

Display the name of the forever-bragger: `Jay` or `ja$on`.

Sample Input 1

```
7
0
1
1
2
2
3
0
```

Sample Output 1

```
ja$on
```

Sample Input 2

```
6
0
1
1
2
2
3
```

Sample Output 2

```
Jay
```

Sample Input 3	Sample Output 3
3 0 1 1	Jay

Problem K
Keeping Arrays Menacing
Time limit: 10 seconds

Jay is back at it again. A real menace to society, he now wants to find the most menacing menace array.

An array A contains many *menace arrays*. The menace arrays of an array A are all subsequences of A that can be constructed using the following process: First, pick a subarray (a subarray is a contiguous subsequence) of A that has more than K elements. Then, delete exactly K elements from that subarray. We are only interested in the sum of each of these menace arrays. The *menace value* of A is the largest sum over all the menace arrays in A, or it is zero if there are fewer than $K+1$ elements in A to begin with.

For example, if $A = [1, 2, 3, 4, 5]$ and $K = 2$, then $[1, 3]$ is a menace array, since you could select $[1, 2, 3, 4]$ and delete 2 and 4. However, $[1, 5]$ is not a menace array since the only subarray that contains both 1 and 5 is the whole array and we need to delete exactly 2 elements.

Jay is going to play a game. He begins with an empty array, and then he performs a sequence of Q queries. Each query is one of two types: an insertion query or a menace query. In an insertion query, Jay inserts a number at either the beginning or end of the array. In a menace query, Jay determines the menace value of his current array.

Perform the sequence of queries for Jay.

Input

The first line of the input contains two integers Q ($1 \leq Q \leq 200\,000$), which is the number of queries, and K ($0 \leq K \leq 10$).

The next Q lines describe the queries. Each line starts with a single character c (c is one of `L`, `R`, or `M`). If $c = $ `M`, this is a menace query. Otherwise, it is an insert query, and this line contains is a second integer V ($-1\,000\,000 \leq V \leq 1\,000\,000$), which is the value to insert to the array. If $c = $ `L`, then V should be inserted at the beginning. If $c = $ `R`, then V should in inserted at the end.

Output

For each menace query in the order given in the input, display the menace value for the current array.

Sample Input 1	Sample Output 1
8 0 L 5 M R 5 M L -5 M L 20 M	5 10 10 25

Sample Input 2	Sample Output 2
8 1 L 5 M R 5 M L -5 M L 20 M	0 5 10 30

Sample Input 3	Sample Output 3
7 1 L 5 R 5 L -5 L 20 L -5 R -5 M	30

Sample Input 4	Sample Output 4
3 0 M M M	0 0 0

Sample Input 5	Sample Output 5
3 1 M L 5 M	0 0

Sample Input 6	Sample Output 6
2 0 L -5 M	0

Problem L
Land Tax
Time limit: 6 seconds

Sir Elong Muskrat does not like to pay taxes and has been evading them for years using shell companies and creative accounting. The government has finally decided to crack down on his tax evasion.

Sir Elong owns lots of rental properties all of which are located consecutively on the same street. There are N properties on the street, all of which are owned by Sir Elong, and they are numbered from 1 to N. The ith property has a value v_i and a size s_i. The tax paid on the properties is the sum of their values.

Sir Elong has decided to minimise his tax by merging properties. Two directly adjacent properties can be merged by knocking down the fence between them. The value of the merged property is the maximum of their values ($\max(v_i, v_j)$) and the size is the sum of their sizes ($s_i + s_j$). Sir Elong cannot create a property whose size is larger than the legal limit L. Sir Elong can repeatedly merge properties as many times as he likes otherwise. He can merge properties that were created in a previous merger.

What is the minimum amount of tax Sir Elong can pay?

Input

The first line of input contains two integers N ($1 \leq N \leq 200\,000$) and L ($1 \leq L \leq 10^9$), which are the number of properties and the legal property size limit respectively.

The next N lines describe the properties. The ith of these describes property number i and contains two integers v_i ($0 \leq v_i \leq 10^9$), this property's value, and s_i ($1 \leq s_i \leq L$), this property's size.

Output

Display the minimum amount of tax payable by Sir Elong.

Sample Input 1	Sample Output 1
3 3 1 1 2 2 3 3	5

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

This page is intentionally left (almost) blank.