Problem J1: Dog Treats

Problem Description

Barley the dog loves treats. At the end of the day he is either happy or sad depending on the number and size of treats he receives throughout the day. The treats come in three sizes: small, medium, and large. His happiness score can be measured using the following formula:

$$1 \times S + 2 \times M + 3 \times L$$

where S is the number of small treats, M is the number of medium treats and L is the number of large treats.

If Barley's happiness score is 10 or greater then he is happy. Otherwise, he is sad. Determine whether Barley is happy or sad at the end of the day.

Input Specification

There are three lines of input. Each line contains a non-negative integer less than 10. The first line contains the number of small treats, S, the second line contains the number of medium treats, M, and the third line contains the number of large treats, L, that Barley receives in a day.

Output Specification

If Barley's happiness score is 10 or greater, output `happy`. Otherwise, output `sad`.

Sample Input 1

```
3
1
0
```

Output for Sample Input 1

```
sad
```

Explanation of Output for Sample Input 1

Barley's happiness score is $1 \times 3 + 2 \times 1 + 3 \times 0 = 5$, so he will be sad.

Sample Input 2

```
3
2
1
```

Output for Sample Input 2

```
happy
```

Explanation of Output for Sample Input 2

Barley's happiness score is $1 \times 3 + 2 \times 2 + 3 \times 1 = 10$, so he will be happy.

La version française figure à la suite de la version anglaise.

Problem J2: Epidemiology

Problem Description

People who study epidemiology use models to analyze the spread of disease. In this problem, we use a simple model.

When a person has a disease, they infect exactly R other people but only on the very next day. No person is infected more than once. We want to determine when a total of more than P people have had the disease.

(This problem was designed before the current coronavirus outbreak, and we acknowledge the distress currently being experienced by many people worldwide because of this and other diseases. We hope that including this problem at this time highlights the important roles that computer science and mathematics play in solving real-world problems.)

Input Specification

There are three lines of input. Each line contains one positive integer. The first line contains the value of P. The second line contains N, the number of people who have the disease on Day 0. The third line contains the value of R. Assume that $P \leq 10^7$ and $N \leq P$ and $R \leq 10$.

Output Specification

Output the number of the first day on which the total number of people who have had the disease is greater than P.

Sample Input 1

```
750
1
5
```

Output for Sample Input 1

```
4
```

Explanation of Output for Sample Input 1

The 1 person on Day 0 with the disease infects 5 people on Day 1. On Day 2, exactly 25 people are infected. On Day 3, exactly 125 people are infected. A total of $1 + 5 + 25 + 125 + 625 = 781$ people have had the disease by the end of Day 4 and $781 > 750$.

Sample Input 2

```
10
2
1
```

Output for Sample Input 2

```
5
```

La version française figure à la suite de la version anglaise.

Explanation of Output for Sample Input 2

There are 2 people on Day 0 with the disease. On each other day, exactly 2 people are infected. By the end of Day 4, a total of exactly 10 people have had the disease and by the end of Day 5, more than 10 people have had the disease.

La version française figure à la suite de la version anglaise.

Problem J3: Art

Problem Description

Mahima has been experimenting with a new style of art. She stands in front of a canvas and, using her brush, flicks drops of paint onto the canvas. When she thinks she has created a masterpiece, she uses her 3D printer to print a frame to surround the canvas.

Your job is to help Mahima by determining the coordinates of the smallest possible rectangular frame such that each drop of paint lies inside the frame. Points on the frame are not considered inside the frame.

Input Specification

The first line of input contains the number of drops of paint, N, where $2 \leq N \leq 100$ and N is an integer. Each of the next N lines contain exactly two positive integers X and Y separated by one comma (no spaces). Each of these pairs of integers represents the coordinates of a drop of paint on the canvas. Assume that $X < 100$ and $Y < 100$, and that there will be at least two distinct points. The coordinates $(0, 0)$ represent the bottom-left corner of the canvas.

For 12 of the 15 available marks, X and Y will both be two-digit integers.

Output Specification

Output two lines. Each line must contain exactly two non-negative integers separated by a single comma (no spaces). The first line represents the coordinates of the bottom-left corner of the rectangular frame. The second line represents the coordinates of the top-right corner of the rectangular frame.

Sample Input

```
5
44,62
34,69
24,78
42,44
64,10
```

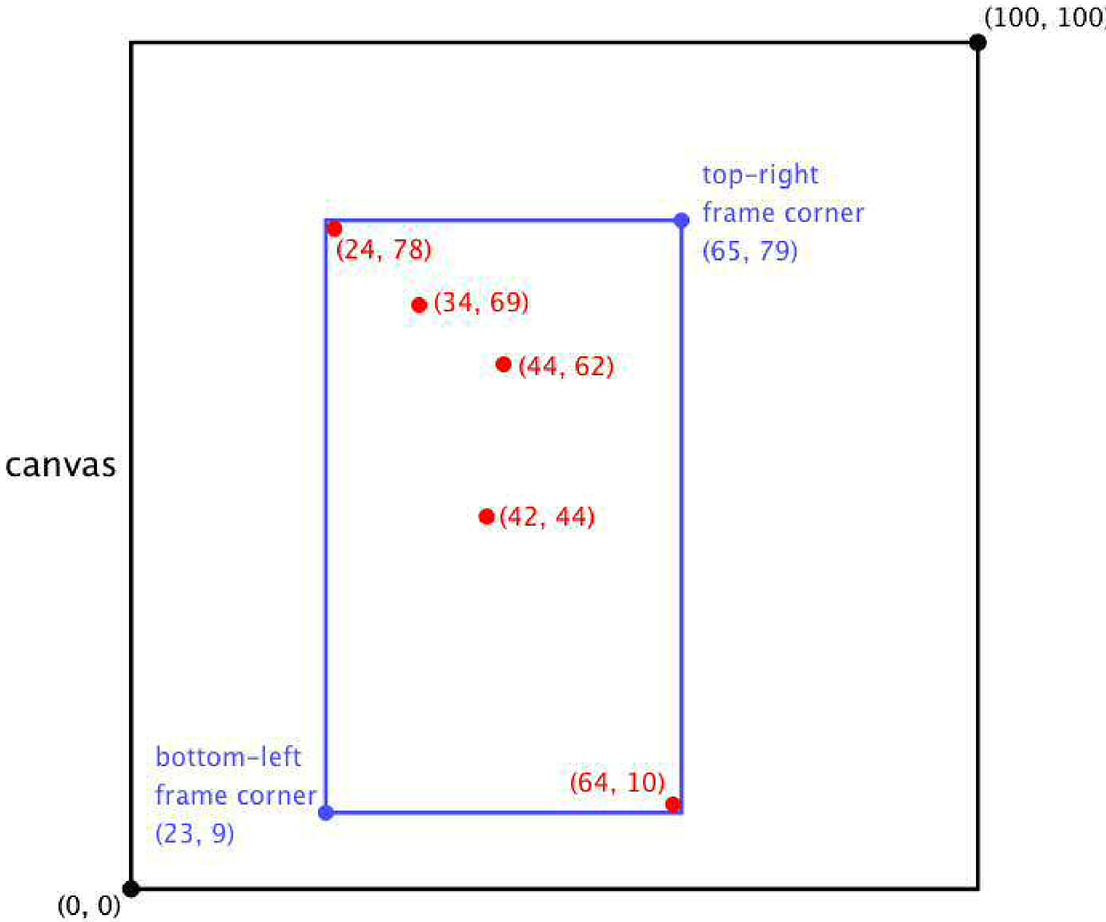
Output for Sample Input

```
23,9
65,79
```

Explanation of Output for Sample Input

The bottom-left corner of the frame is $(23, 9)$. Notice that if the bottom-left corner is moved up, the paint drop at $(64, 10)$ will not be inside the frame. (See the diagram on the next page.) If the corner is moved right, the paint drop at $(24, 78)$ will not be inside the frame. If the corner is moved down or left, then the frame will be larger and no longer the smallest rectangle containing all the drops of paint. A similar argument can be made regarding the top-right corner of the frame.

La version française figure à la suite de la version anglaise.



La version française figure à la suite de la version anglaise.

Problem J4: Cyclic Shifts

Problem Description

Thuc likes finding cyclic shifts of strings. A *cyclic shift* of a string is obtained by moving characters from the beginning of the string to the end of the string. We also consider a string to be a cyclic shift of itself. For example, the cyclic shifts of `ABCDE` are:

`ABCDE`, `BCDEA`, `CDEAB`, `DEABC`, and `EABCD`.

Given some text, T, and a string, S, determine if T contains a cyclic shift of S.

Input Specification

The input will consist of exactly two lines containing only uppercase letters. The first line will be the text T, and the second line will be the string S. Each line will contain at most 1000 characters.

For 6 of the 15 available marks, S will be exactly 3 characters in length.

Output Specification

Output `yes` if the text, T, contains a cyclic shift of the string, S. Otherwise, output `no`.

Sample Input 1

```
ABCCDEABAA
ABCDE
```

Output for Sample Input 1

```
yes
```

Explanation of Output for Sample Input 1

`CDEAB` is a cyclic shift of `ABCDE` and it is contained in the text `ABC`**`CDEAB`**`AA`.

Sample Input 2

```
ABCDDEBCAB
ABA
```

Output for Sample Input 2

```
no
```

Explanation of Output for Sample Input 2

The cyclic shifts of `ABA` are `ABA`, `BAA`, and `AAB`. None of these shifts are contained in the text `ABCDDEBCAB`.

La version française figure à la suite de la version anglaise.

Problem J5/S2: Escape Room

Problem Description
You have to determine if it is possible to escape from a room. The room is an M-by-N grid with each position (cell) containing a positive integer. The rows are numbered $1, 2, \ldots, M$ and the columns are numbered $1, 2, \ldots, N$. We use (r, c) to refer to the cell in row r and column c.

You start in the top-left corner at $(1, 1)$ and exit from the bottom-right corner at (M, N). If you are in a cell containing the value x, then you can jump to any cell (a, b) satisfying $a \times b = x$. For example, if you are in a cell containing a 6, you can jump to cell $(2, 3)$.

Note that from a cell containing a 6, there are up to four cells you can jump to: $(2, 3)$, $(3, 2)$, $(1, 6)$, or $(6, 1)$. If the room is a 5-by-6 grid, there isn't a row 6 so only the first three jumps would be possible.

Input Specification
The first line of the input will be an integer M ($1 \leq M \leq 1000$). The second line of the input will be an integer N ($1 \leq N \leq 1000$). The remaining input gives the positive integers in the cells of the room with M rows and N columns. It consists of M lines where each line contains N positive integers, each less than or equal to $1\,000\,000$, separated by single spaces.

For 1 of the 15 available marks, $M = 2$ and $N = 2$.

For an additional 2 of the 15 available marks, $M = 1$.

For an additional 4 of the 15 available marks, all of the integers in the cells will be unique.

For an additional 4 of the 15 available marks, $M \leq 200$ and $N \leq 200$.

Output Specification
Output `yes` if it is possible to escape from the room. Otherwise, output `no`.

Sample Input
```
3
4
3 10  8 14
1 11 12 12
6  2  3  9
```

Output for Sample Input
```
yes
```

La version française figure à la suite de la version anglaise.

Explanation of Output for Sample Input

Starting in the cell at $(1,1)$ which contains a 3, one possibility is to jump to the cell at $(1,3)$. This cell contains an 8 so from it, you could jump to the cell at $(2,4)$. This brings you to a cell containing 12 from which you can jump to the exit at $(3,4)$. Note that another way to escape is to jump from the starting cell to the cell at $(3,1)$ to the cell at $(2,3)$ to the exit.

Notes

1. The online grader begins by testing submissions using the sample input. All other tests are skipped if the sample test is not passed. If you are only attempting the first three subtasks (the first 7 marks), then you might want to handle the specific values of the sample input as a special case.

2. For the final subtask (worth 2 marks), if you are using Java, then `Scanner` will probably take too long to read in the large amount of data. A much faster alternative is `BufferedReader`.

La version française figure à la suite de la version anglaise.

Problem S1: Surmising a Sprinter's Speed

Problem Description

Trick E. Dingo is trying, as usual, to catch his nemesis the Street Sprinter. His past attempts using magnets, traps and explosives have failed miserably, so he's catching his breath to gather observational data and learn more about how fast Street Sprinter is.

Trick E. Dingo and Street Sprinter both inhabit a single straight west-east road with a particularly famous rock on it known affectionately as The Origin. Positions on this straight road are measured numerically according to the distance from The Origin, and using negative numbers for positions west of The Origin and positive numbers for positions east of The Origin.

The observations by Trick E. Dingo each contain two numbers: a time, and the value of Street Sprinter's position on the road at that time. Given this information, what speed must Street Sprinter must be capable of?

Input Specification

The first line contains a number $2 \leq N \leq 100\,000$, the number of observations that follow. The next N lines each contain an integer $0 \leq T \leq 1\,000\,000\,000$ indicating the time, in seconds, of when a measurement was made, and an integer $-1\,000\,000\,000 \leq X \leq 1\,000\,000\,000$ indicating the position, in metres, of the Street Sprinter at that time. No two lines will have the same value of T.

For 7 of the 15 available marks, $N \leq 1000$.

Output Specification

Output a single number X, such that we can conclude that Street Sprinter's speed was at least X metres/second at some point in time, and such that X is as large as possible. If the correct answer is C, the grader will view X as correct if $|X - C|/C < 10^{-5}$.

Sample Input 1

```
3
0 100
20 50
10 120
```

Output for Sample Input 1

```
7.0
```

Explanation of Output for Sample Input 1

Since the Street Sprinter ran from position 100 to position 120 between time 0 and time 10, we know its speed must have been at least 2 at some point in time: if it was always less than 2, then the distance of 20 could not be covered in 10 seconds. Likewise, the speed must have been at least 7 in order to travel between position 120 and 50 in 10 seconds.

La version française figure à la suite de la version anglaise.

Sample Input 2

```
5
20 -5
0 -17
10 31
5 -3
30 11
```

Output for Sample Input 2

```
6.8
```

La version française figure à la suite de la version anglaise.

Problem S3: Searching for Strings

Problem Description

You're given a string N, called the *needle*, and a string H, called the *haystack*, both of which contain only lowercase letters "a".."z".

Write a program to count the number of distinct permutations of N which appear as a substring of H at least once. Note that N can have anywhere between 1 and $|N|!$ distinct permutations in total – for example, the string "aab" has 3 distinct permutations ("aab", "aba", and "baa").

Input Specification

The first line contains N ($1 \le |N| \le 200\,000$), the needle string.

The second line contains H ($1 \le |H| \le 200\,000$), the haystack string.

For 3 of the 15 available marks, $|N| \le 8$ and $|H| \le 200$.

For an additional 2 of the 15 available marks, $|N| \le 200$ and $|H| \le 200$.

For an additional 2 of the 15 available marks, $|N| \le 2000$ and $|H| \le 2000$.

Output Specification

Output consists of one integer, the number of distinct permutations of N which appear as a substring of H.

Sample Input

```
aab
abacabaa
```

Output for Sample Input

```
2
```

Explanation of Output for Sample Input

The permutations "aba" and "baa" each appear as substrings of H (the former appears twice), while the permutation "aab" does not appear.

La version française figure à la suite de la version anglaise.

Problem S4: Swapping Seats

Problem Description
There are N people sitting at a circular table for a long session of negotiations. Each person belongs to one of the three groups: A, B, or C. A group is *happy* if all of its members are sitting contiguously in a block of consecutive seats. You would like to make all groups happy by some sequence of *swap* operations. In each swap operation, two people exchange seats with each other. What is the minimum number of swaps required to make all groups happy?

Input Specification
The input consists of a single line containing N ($1 \leq N \leq 1\,000\,000$) characters, where each character is A, B, or C. The i-th character denotes the group of the person initially sitting at the i-th seat at the table, where seats are numbered in clockwise order.

For 4 of the 15 available marks, the input has no C characters and $N \leq 5\,000$.

For an additional 4 of the 15 available marks, the input has no C characters.

For an additional 4 of the 15 available marks, $N \leq 5\,000$.

Output Specification
Output a single integer, the minimum possible number of swaps.

Sample Input
```
BABCBCACCA
```

Output for Sample Input
```
2
```

Explanation of Output for Sample Input
In one possible sequence, the first swap results in the seating layout AABCBCBCCA. After the second swap, the layout is AABBBCCCCA.

La version française figure à la suite de la version anglaise.

Problem S5: Josh's Double Bacon Deluxe

Problem Description
On their way to the contest grounds, Josh, his coach, and his $N-2$ teammates decide to stop at a burger joint that offers M distinct burger menu items. After ordering their favourite burgers, the team members line up, with the coach in the first position and Josh last, to pick up their burgers. Unfortunately, the coach forgot what he ordered. He picks a burger at random and walks away. The other team members, in sequence, pick up their favourite burger if available, or a random remaining burger if there are no more of their favourite burger. What is the probability that Josh, being last in line, will get to eat his favourite burger?

Input Specification
The first line contains the number N ($3 \leq N \leq 1\,000\,000$), the total number of people and burgers. The next line contains N numbers, the i-th being b_i ($1 \leq b_i \leq M \leq 500\,000$), denoting the item number of the i-th person's favourite burger. The first person in line is the coach, and the N-th person is Josh.

For 4 of the 15 available marks, $N \leq 100\,000$ and $M \leq 1000$

For an additional 5 of the 15 available marks, $M \leq 5000$.

Output Specification
Output a single number P, the probability that Josh will get to eat his favourite burger, b_N. If the correct answer is C, the grader will view P correct if $|P - C| < 10^{-6}$.

Sample Input 1
```
3
1 2 3
```

Output for Sample Input 1
```
0.5
```

Explanation of Output for Sample Input
The coach randomly chooses between the three burgers. With probability $1/3$, he chooses his favourite burger (burger 1), and Josh gets to eat his favourite burger (burger 3). With probability $1/3$, he chooses Josh's favourite burger, and Josh fails to eat his favourite burger. With probability $1/3$, he chooses the second person's burger, there is a $1/2$ chance that the second person chooses Josh's burger, denying Josh the pleasure of eating his favourite burger.

Sample Input 2
```
7
1 2 3 1 1 2 3
```

La version française figure à la suite de la version anglaise.

Output for Sample Input 2

```
0.57142857
```

La version française figure à la suite de la version anglaise.

2020 Canadian Computing
Problem 1
A Game with Grundy

Time Limit: 1 second

Problem Description
Grundy is playing his favourite game - hide and seek.

His N friends stand at locations on the x-axis of a two-dimensional plane - the i-th one is at coordinates $(x_i, 0)$. Each friend can see things in a triangular wedge extending vertically upwards from their position – the i-th friend's triangular wedge of vision will be specified by two lines: one with slope of v_i/h_i and the other with slope $-v_i/h_i$. A friend cannot see a point that lies exactly on one of these two lines.

Grundy may choose to hide at any location (a, Y), where a is an integer satisfying $L \leq a \leq R$, and L, R, and Y are given integer constants.

Each possible location may be in view of some of Grundy's friends (namely, strictly within their triangular wedge of vision).

Grundy would like to know in how many different spots he can stand such that he will be in view of at most i of his friends, for every possible value of i from 0 to N.

Input Specification
The first line of input contains the integer N ($1 \leq N \leq 100\,000$).

The next line contains three integers: L, R and Y ($-1\,000\,000\,000 \leq L \leq R \leq 1\,000\,000\,000, 1 \leq Y \leq 1\,000\,000$).

Each of the next N lines contain three integers: the i-th such line contains x_i ($L \leq x_i \leq R$), the x-value of the position of friend i followed by two integers v_i and h_i ($1 \leq v_i, h_i \leq 100$). The slopes v_i/h_i and $-v_i/h_i$ define the triangular wedge of vision for friend i.

For 15 of the 25 marks available, $-1\,000\,000 \leq L \leq R \leq 1\,000\,000$.

Output Specification
The output is $N + 1$ lines, where each line i ($0 \leq i \leq N$) contains the integer number of positions in which Grundy can stand and be in view of at most i of his friends.

Sample Input
```
3
-7 7 3
0 2 3
```

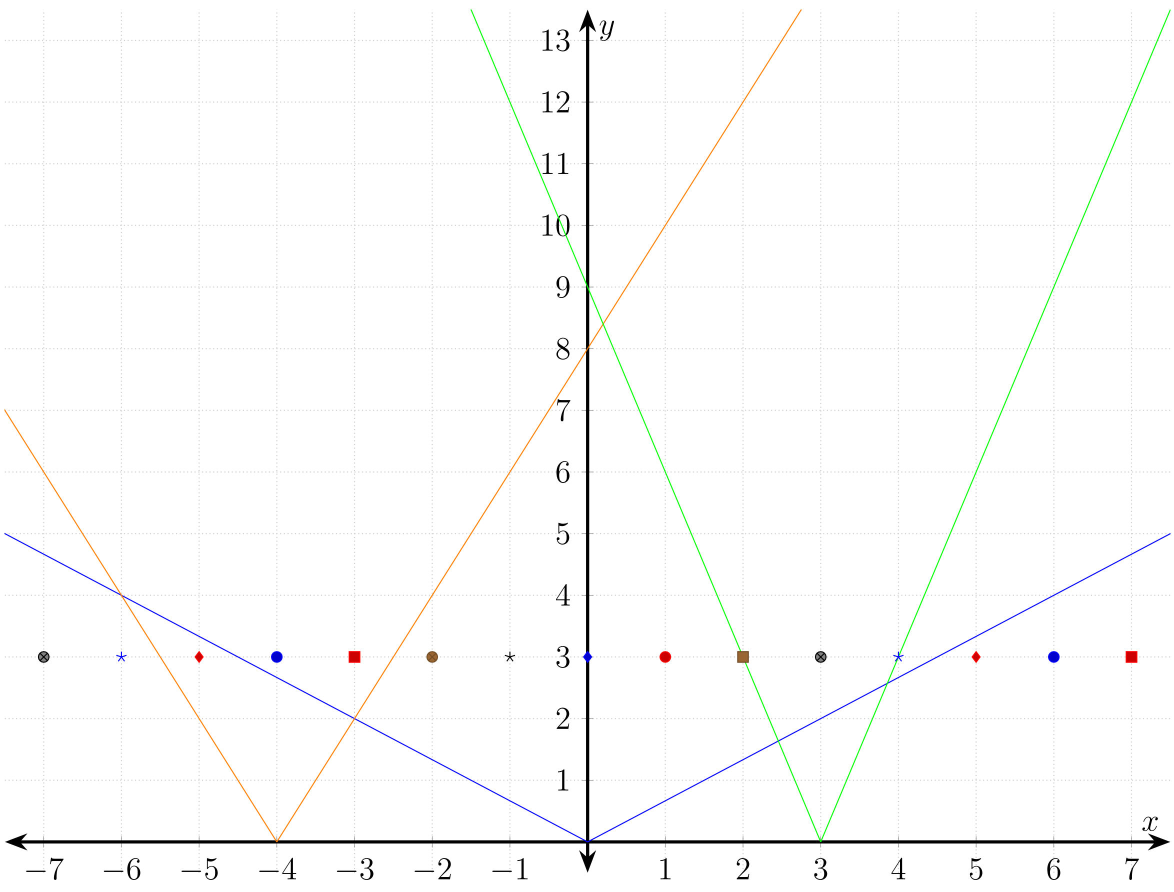
```
-4 2 1
3 3 1
```

Output for Sample Input

```
5
12
15
15
```

Explanation of Output for Sample Input

There are three friends with the following triangular wedges of vision, along with the possible positions that Grundy can be placed, as shown in the diagram below:



Notice the points $(2, 3)$ and $(4, 3)$ are visible only by the friend at position 0, since they lie on the boundary of the triangular wedge of vision for the friend at position 3.

2020 Canadian Computing
Problem 2
Exercise Deadlines

Time Limit: 1 second

Problem Description

Bob has N programming exercises that he needs to complete before their deadlines. Exercise i only takes one time unit to complete, but has a deadline d_i ($1 \leq d_i \leq N$) time units from now.

Bob will solve the exercises in an order described by a sequence $a_1, a_2, \ldots, a_N$, such that a_1 is the first exercise he solves, a_2 is the second exercise he solves, and so on. Bob's original plan is described by the sequence $1, 2, \ldots, N$. With one *swap* operation, Bob can exchange two adjacent numbers in this sequence. What is the minimum number of swaps required to change this sequence into one that completes all exercises on time?

Input Specification

The first line consists of a single integer N ($1 \leq N \leq 200\,000$). The next line contains N space-separated integers $d_1, d_2, \ldots, d_N$ ($1 \leq d_i \leq N$).

For 17 of the 25 marks available, $N \leq 5000$.

Output Specification

Output a single integer, the minimum number of swaps required for Bob to solve all exercises on time, or `-1` if this is impossible.

Sample Input 1

```
4
4 4 3 2
```

Output for Sample Input 1

```
3
```

Explanation of Output for Sample Input 1

One valid sequence is $(1, 4, 3, 2)$, which can be obtained from $(1, 2, 3, 4)$ by three swaps.

Sample Input 2

```
3
1 1 3
```

Output for Sample Input 2

```
-1
```

Explanation of Output for Sample Input 2

There are two exercises that are due at time 1, but only one exercise can be solved by this time.

2020 Canadian Computing
Problem 3
Mountains and Valleys

Time Limit: 7 seconds

Problem Description

You are planning a long hiking trip through some interesting, but well-known terrain. There are N interesting sites you would like to visit and M trails connecting pairs of sites. Each trail has a difficulty level indicated as a positive integer.

The trail system is a bit special, however. There are exactly $N-1$ trails with difficulty level 1 (these are completely flat trails), and the rest of the trails all have a difficulty level of at least $\left\lceil \frac{N}{3} \right\rceil$ (these are very mountainous trails). (The ceiling of x, denoted as $\lceil x \rceil$, is the smallest integer greater than or equal to x.)

Additionally, it is possible to travel between any two sites using only the trails with difficulty level 1.

You would like to visit every site, starting your walk from any site of your choice and finishing at some other site, such that you visit each site at least once and the total sum of difficulty levels is minimum among all walks. Note that walking a trail k times with difficulty level d contributes a value of $k \cdot d$ to the sum of difficulty levels.

Input Specification

The first line of input contains two space-separated integers N ($4 \leq N \leq 500\ 000$) and M ($N-1 \leq M \leq 2\ 000\ 000$).

The next M lines contain three space-separated integers x_i, y_i, and w_i describing the trail between sites x_i and y_i with difficulty level w_i ($1 \leq i \leq M$; $0 \leq x_i, y_i \leq N-1$; $x_i \neq y_i$). Note that there is at most one trail between every pair of sites, and that $w_i = 1$ or $\left\lceil \frac{N}{3} \right\rceil \leq w_i \leq N$.

For 1 of the 25 marks available, $N \leq 6$ and $M \leq 10$.

For an additional 2 of the 25 marks available, $N \leq 20$ and $M \leq 40$.

For an additional 2 of the 25 marks available, $N \leq 5\,000$, $M \leq 10\,000$, and either $w_i = 1$ or $\left\lceil \frac{N}{2} \right\rceil \leq w_i \leq N$.

For an additional 6 of the 25 marks available, $N \leq 100$ and $M \leq 200$.

For an additional 2 of the 25 marks available, $N \leq 500$ and $M \leq 1\,000$.

For an additional 3 of the 25 marks available, $N \leq 5\,000$ and $M \leq 10\,000$.

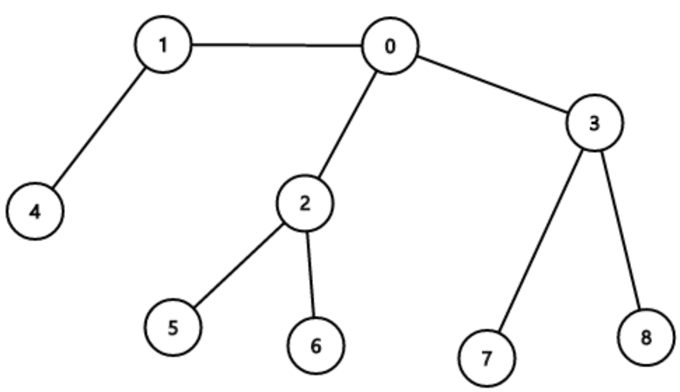
For an additional 5 of the 25 marks available, $N \leq 80\,000$ and $M \leq 160\,000$.

Output Specification
Output one integer, which is the minimum sum of difficulty levels taken for all trails walked to visit each site at least once.

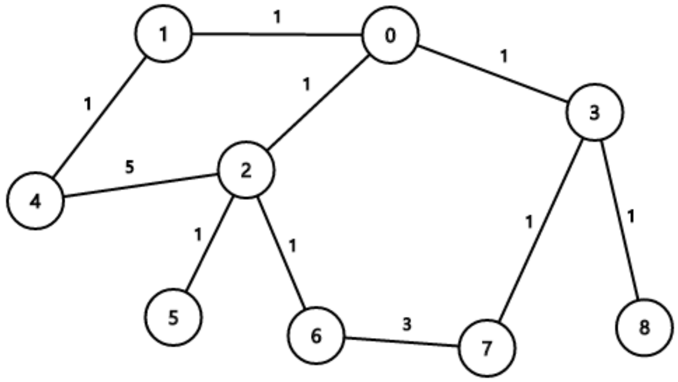
Sample Input
```
9 10
0 1 1
0 2 1
0 3 1
1 4 1
2 5 1
2 6 1
3 7 1
3 8 1
2 4 5
6 7 3
```

Output for Sample Input
```
11
```

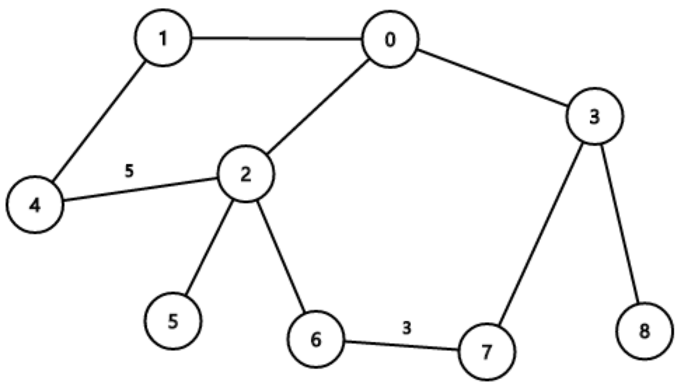
Explanation of Output for Sample Input
This is the set of flat trails:



This is the entire set of trails with all the difficulty levels.



This is the entire set of trails, with trails with difficulty level 1 being omitted.



An optimal walk for this set of trails is $4 \to 1 \to 0 \to 2 \to 5 \to 2 \to 6 \to 7 \to 3 \to 8$ with a total cost of $1 + 1 + 1 + 1 + 1 + 1 + 3 + 1 + 1 = 11$. There is no way to make a walk that visits all the sites at least once with a lower total difficulty level cost.

2020 Canadian Computing
Problem 4
Travelling Salesperson

Time Limit: 7 seconds

Problem Description

In the city of RedBlue, every pair of buildings is connected by a road, either red or blue. To switch from travelling along red roads to blue roads or vice versa costs one ticket. The length of a route is the number of buildings that are visited. For example, the following route has a length of five and costs one ticket.

1 — 2 — 3 — 4 — 3

If we wanted to travel on a blue road again after visiting vertex 3 for the second time, we would need another ticket, for a total of two tickets:

1 — 2 — 3 — 4 — 3 — 2

You are a travelling salesperson visiting the city of RedBlue, and you wish to visit each building at least once, while minimizing repeated visits of the same buildings. You have not yet decided which building you are starting your route from, so you would like to plan out all possible routes. Furthermore, you only have access to one ticket. For each building, you would like to find a route of minimum length that begins at that building, visits all the buildings at least once, and uses at most one ticket.

Input Specification

The first line will contain a single integer N $(2 \leq N \leq 2\,000)$, the number of buildings in RedBlue.

Lines 2 to N each contain a string, with line i containing the string C_i, representing the colours of the roads connected to building i. The string $C_i = C_{i,1}C_{i,2}...C_{i,i-1}$ has a length of $i-1$ and consists only of the characters `R` and `B`. If $C_{i,j}$ is `R`, then the road between buildings i and j is red. Otherwise, it is blue.

Output Specification

Output $2N$ lines. Lines $2i-1$ for $1 \leq i \leq N$ should contain a single integer M_i, representing the length of the travel plan starting at building i. Lines $2i$ for $1 \leq i \leq N$ should each contain M_i space separated integers, describing the order in which you visit the buildings, starting at building i.

Scoring

For every one of your travel plans, a score is computed. Let K_i be the length of the optimal route starting at each building, and let M_i be the length of your route. If M_i is greater than $2K_i$, then your score will be 0, and you will receive a verdict of Wrong Answer. If M_i is equal to K_i, then your score will be 25. Otherwise you will a receive a score of $\lfloor 8 + 8 \times \frac{2K_i - M_i}{K_i - 1} \rfloor$. Your score for the test case is the minimum score for each travel plan.

If any of your plans are invalid, your score will be 0, and you will receive a verdict of Wrong Answer.

Your submission's score is the minimum score over all test cases.

Sample Input

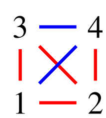
```
4
R
RR
BRB
```

Possible Output for Sample Input

```
5
1 4 2 1 3
6
2 3 1 2 3 4
5
3 1 2 3 4
4
4 3 1 2
```

Explanation of Possible Output for Sample Input

RedBlue looks like this:



The route starting from building 3 has an optimal length of 4 by visiting the buildings in the order $3, 2, 1, 4$. The solution's route has a length of 5, meaning the score is equal to $\lfloor 8+8\times \frac{2\times 4-5}{4-1} \rfloor = 16$.

2020 Canadian Computing
Problem 5
Interval Collection

Time Limit: 3.5 seconds

Problem Description

Altina is starting an interval collection. An interval is defined as two positive integers $[l, r]$ such that $l < r$. We say that the length of this interval is $r - l$. Additionally, we say that an interval $[l, r]$ contains another interval $[x, y]$ if $l \leq x$ and $y \leq r$. In particular, each interval contains itself.

For a non-empty set S of intervals, we define the set of common intervals as all the intervals $[x, y]$ that are contained within every interval $[l, r]$ in S. If the set of common intervals is non-empty, then we say the **greatest common interval** of S is equal to the common interval with the largest length.

For the same set S, we define the set of enclosing intervals as all the intervals $[x, y]$ that contain every interval $[l, r]$ in S. Note that this set is always non-empty, so we say the **least enclosing interval** of S is equal to the enclosing interval with the smallest length.

Initially, Altina owns no intervals in her collection. There are Q events that change the set of intervals she owns.

The first type of event is when Altina adds an interval $[l, r]$ to her collection. Note that this interval could have the same $[l, r]$ as another interval in her collection. They should be treated as separate intervals.

The second type of event is when Altina removes an existing interval $[l, r]$ from her collection. Note that if Altina has more than one interval with the same $[l, r]$, she removes exactly one of them.

After each event, Altina chooses a non-empty subset S of intervals she owns in her collection that satisfy the following conditions:

- Among all sets S Altina could choose, she chooses one that has **no** greatest common interval, if possible. If this is impossible, then she chooses one which has the length of its greatest common interval as small as possible.
- Among all sets S that satisfy the previous condition, she chooses one which has the length of its least enclosing interval as small as possible.

Your task is to determine the length of the least enclosing interval of the set S Altina chose after each event.

Input Specification
The first line of input contains Q ($1 \leq Q \leq 500\,000$), the number of add and remove operations in total. The next Q lines are in one of the following forms:

- `A` l r: add the interval $[l, r]$ to Altina's collection.
- `R` l r: remove one of the instances of the interval $[l, r]$ from Altina's collection . It is guaranteed the interval to be removed exists and that the collection will be non-empty after the interval is removed.

For all subtasks, $1 \leq l < r \leq 1\,000\,000$.

For 3 of the 25 marks available, $Q \leq 500$.

For an additional 8 of the 25 marks available, $Q \leq 12\,000$.

For an additional 7 of the 25 marks available, $Q \leq 50\,000$.

For an additional 4 of the 25 marks available, the following condition holds after each event: for every two separate intervals $[l_1, r_1]$ and $[l_2, r_2]$ in Altina's collection, either $r_1 < l_2$ or $r_2 < l_1$.

Output Specification
The output consists of Q lines, each line containing the length of the least enclosing interval for Altina's choice of S as described in the problem description.

Sample Input
```
5
A 1 5
A 2 7
A 4 6
A 6 8
R 4 6
```

Output for Sample Input
```
4
6
5
4
7
```

Explanation of Output for Sample Input
After the interval $[1, 5]$ is added, there is only one interval, so $S = \{[1, 5]\}$ is the only valid choice and the least enclosing interval is $[1, 5]$.

After the interval $[2, 7]$ is added, $S = \{[1, 5], [2, 7]\}$ has the greatest common interval $[2, 5]$ and least enclosing interval $[1, 7]$.

After the interval $[4, 6]$ is added, $S = \{[1, 5], [4, 6]\}$ has the greatest common interval $[4, 5]$ and least enclosing interval $[1, 6]$.

After the interval $[6, 8]$ is added, $S = \{[4, 6], [6, 8]\}$ has no greatest common interval and its least enclosing interval $[4, 8]$. Note that $S = \{[1, 5], [6, 8]\}$ also has no greatest common interval but its least enclosing interval $[1, 8]$ has a greater length than $[4, 8]$.

After the interval $[4, 6]$ is removed, $S = \{[1, 5], [6, 8]\}$ has no greatest common interval and least enclosing interval $[1, 8]$.

2020 Canadian Computing
Problem 6
Shopping Plans

Time Limit: 2 seconds

Problem Description
You are shopping from a store that sells a total of N items. The i-th item has a *type* a_i which is an integer between 1 and M. A feasible shopping plan is a subset of these items such that for all types j, the number of items of type j is in the interval $[x_j, y_j]$.

The i-th item in the store has a cost of c_i, and the cost of a shopping plan is the sum of the costs of items in the plan. You are interested in the possible costs of feasible shopping plans. Find the costs of the K cheapest feasible shopping plans. Note that if there are two different shopping plans with the same cost, they should be counted separately in the output.

Input Specification
The first line consists of three space-separated integers N, M, and K ($1 \leq N, M, K \leq 200\,000$). N lines follow, the i-th of which contains two space-separated integers a_i and c_i ($1 \leq a_i \leq M$, $1 \leq c_i \leq 10^9$). M lines follow, the j-th of which contains two space-separated integers x_j and y_j ($0 \leq x_j \leq y_j \leq N$).

For 5 of the 25 marks available, $x_j = y_j = 1$ and $N, M, K \leq 4000$.

For an additional 5 of the 25 marks available, $x_j = y_j = 1$ and $N, M, c_i \leq 4000$.

For an additional 5 of the 25 marks available, $x_j = y_j = 1$.

For an additional 5 of the 25 marks available, $x_j = 0$.

Output Specification
Output K lines. On the i-th line, output the cost of the i-th cheapest feasible shopping plan, if one exists, or `-1` if there are fewer than i feasible shopping plans.

Sample Input 1
```
5 2 7
1 5
1 3
2 3
1 6
2 1
1 1
1 1
```

Output for Sample Input 1

```
4
6
6
7
8
9
-1
```

Explanation of Output for Sample Input 1

A feasible shopping plan must combine exactly one item with a cost in $\{5, 3, 6\}$ with exactly one item with a cost in $\{3, 1\}$.