ICPC SOUTH PACIFIC REGIONALS

NOVEMBER 24, 2018

Contest Problems

A: Aligned Typesetting
B: Bad Keming
C: Collecting Stars
D: Dropping Blocks
E: Explosive Wiring
F: Far, Far Away
G: Good Cable Management
H: Hazardous Driving
I : Interesting World of Arrays
J : Judging Divisionals
K: Knights and Dragons
L: Laps

This contest contains twelve problems over 26 pages. 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 ($a, b, \ldots, z$).

Problem A
Aligned Typesetting
Time limit: 4 seconds

For this problem, a *sentence* with n words is a sequence of n non-empty strings $[w_1, w_2, \ldots, w_n]$. Given a sentence, a *valid typesetting* of length L is a string of length exactly L which is formed by concatenating all the words in the sentence and inserting a positive number of spaces between each adjacent pair of words.

An *aligned typesetting* is a valid typesetting such that the number of spaces in between each adjacent pair of words is equal.

For example, given the sentence [`harry`, `ron`, `hermione`] and using `_` to indicate a space:

- the string `__harry_ronhermione` is not a valid typesetting;
- the string `harry_ron____hermione` is a valid typesetting of length 22 but it is not an aligned typesetting;
- the string `harry___ron___hermione` is an aligned typesetting of length 22.

Source: Pexels

Darcy was given a sentence of n words and the desired length of typesetting L. Can you help him to figure out whether it is possible to construct an aligned typesetting of the desired length?

Input

The first line contains two integers n ($1 \leq n \leq 10^6$), which is the number of words, and L ($0 \leq L \leq 10^6$), which is the desired length of typesetting.

The next n lines describe the words. Each of these lines contains a single string w_i, representing the i^{th} word in the sentence. The word contains only lowercase letters and consists of at least 1 and at most 10^6 characters.

The total length of all the words in the sentence is guaranteed to be at most 10^6.

Output

Display if there is an aligned typesetting of the sentence with the given length.

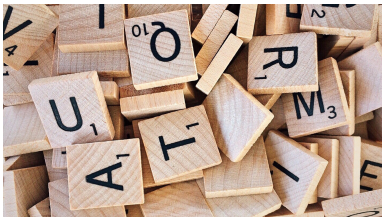
Sample Input 1	Sample Output 1
3 8 harry j p	No

Sample Input 2	Sample Output 2
1 5 harry	Yes

Sample Input 3	Sample Output 3
3 22 harry ron hermione	Yes

This page is intentionally left (almost) blank.

Problem B
Bad Keming
Time limit: 2 seconds



Source: Pexels

You and your fellow teammates have just founded a startup to sell nameplates to programmers. After having carefully researched your target market, you have determined that it is best to use a monospace font.

Your very first client has asked you to print the source string S onto a nameplate. Unfortunately, you have set up your laser printer incorrectly and have accidentally printed a space before the first character, after the last character and between every character of the string.

Your client is expecting this nameplate very soon (in 5 hours, to be precise), so you do not have time to reconfigure your printer or to find a new plate. To salvage your current plate, you have decided to use the printer to fill in each space with a single character, such that the longest possible prefix of the source string S appears as a substring of your plate. A substring is a contiguous subsequence of characters.

For example, say you intended to print `ENDED`. Instead, your printer printed `_E_N_D_E_D_` (using `_` to indicate a space), which you could fill as `JEJNJDJENDE`. This string contains `ENDE`, which is a prefix of S with length 4. This is the best you can do.

As a second example, consider the string `ERROR`. This would be printed as `_E_R_R_O_R_`, which you could fill as `JEJRERRORRJ`. In this case, the entire source string `ERROR` is a substring of the final plate.

What is the length of the longest prefix of the source string that you can print on the nameplate?

Input

The input consists of a single line containing the string S. The string contains only uppercase letters and consists of at least 1 and at most 2 000 000 characters.

Output

Display the length of the longest prefix of S that you can print on the nameplate.

Sample Input 1	Sample Output 1
ENDED	4

Sample Input 2	Sample Output 2
ERROR	5

This page is intentionally left (almost) blank.

Problem C
Collecting Stars
Time limit: 2 seconds

You are planning to complete the latest Super Mario game as fast as you can. In this game, there are n stars that can be collected and your objective is to collect any k of them. Each star takes a certain amount of time to get. After collecting a star, Mario will reappear at the start of the game, so the time taken to collect any sequence of stars is the same regardless of the order that they are collected in. However, some stars do not become available for collection until a certain quantity of stars has already been collected.

Given a description of the stars, determine the fastest time in which you could collect k of them or determine that it is impossible to do so.

Source: Pexels

Input

The input starts with a line containing two integers n ($1 \leq n \leq 200\,000$), which is the number of stars, and k ($1 \leq k \leq n$), which is the number of stars you must collect.

The following n lines describe the stars. Each of these lines contains two integers t ($1 \leq t \leq 10^9$), which is the amount of time it will take to collect the star, and d ($0 \leq d < n$), which is the number of stars that must be collected before the star is available.

Output

Display the minimum amount of time to collect k stars. If you cannot collect k stars, display `IMPOSSIBLE`.

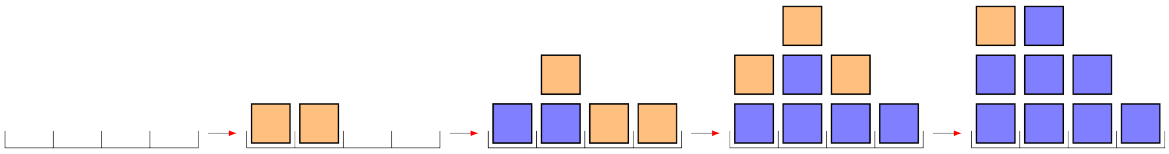
Sample Input 1	Sample Output 1
5 4 1 0 2 1 3 1 2 3 4 0	8

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

This page is intentionally left (almost) blank.

Problem D
Dropping Blocks
Time limit: 2 seconds

Daniel likes playing a game with blocks. The game starts with N empty piles of blocks in a line. While playing the game, Daniel does the following operation: he chooses a pile k and puts a block in every pile either to the left or to the right of pile k (including pile k). A valid game state is reached by applying only this operation zero or more times.



For example, in the above image, Daniel played a game with four piles and performed four operations. Firstly, he put one block in each pile to the left of pile 2 (including pile 2), then he put one block in each pile to the right of pile 2 (including pile 2), then he put one block in each pile to the left of pile 3 (including pile 3), and finally, he put one block in each pile to the left of pile 1 (including pile 1).

Given the number of blocks in each pile, determine if it is a valid game state.

Input

The first line contains a single integer N ($1 \leq N \leq 100\,000$), which is the number of piles.

The second line describes the piles. The line contains N integers, each of which is the number of blocks in a pile. The piles are listed from left to right and each number is at least 0 and at most 100 000.

Output

Display if the input describes a valid game state.

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

Sample Input 2	Sample Output 2
3 1 2 1	YES

Sample Input 3	Sample Output 3
5 1 2 1 2 1	NO

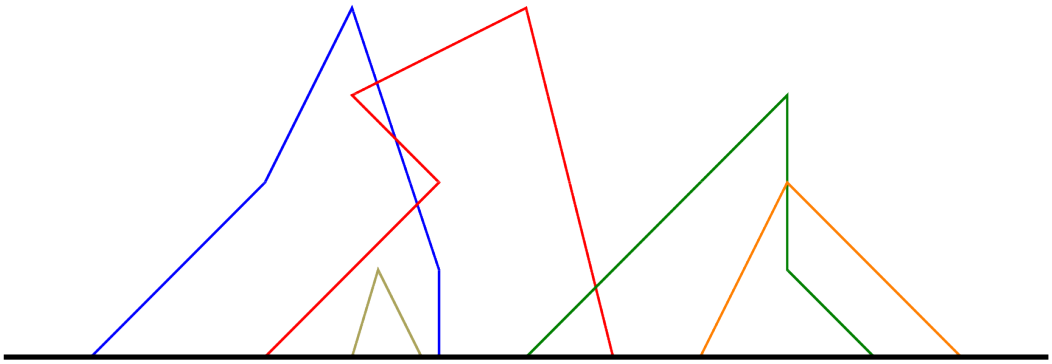
Sample Input 4	Sample Output 4
5 1 2 3 2 1	NO

This page is intentionally left (almost) blank.

Problem E
Explosive Wiring
Time limit: 5 seconds

You are trying to design the wiring for a new kind of computer chip. Unfortunately, the wires are made of a strange material that will explode under the wrong conditions. You have a set of wires that you can install on the chip. Each wire has an associated usefulness value.

To simplify things, you may assume that the chip is on the x-axis. Each wire connects two different points on the x-axis and is described by a 2D polyline connecting those two points. A polyline is a sequence of points connected by straight line segments. All wires are above the x-axis except where they touch the chip at their first and last points. In addition, all the x-coordinates of each wire polyline stay within the interval on the x-axis formed by that polyline's first and last points.



Two wires *interfere* if their polylines touch or intersect at one or more points (a wire does not interfere with itself). A set of the wires is *safe* if each wire in the set interferes with exactly one other wire in the set. A set's *utility* is the sum of the wires' usefulness values. Given a set of wires from which to choose, what is the largest utility over all safe subsets of those wires?

Input

The first line of input contains a single integer N ($1 \leq N \leq 150$), which is the number of wires.

The next N lines describe the wires. Each of these lines starts with two integers k ($0 \leq k \leq 100\,000$), which is the usefulness of the wire, and p ($3 \leq p \leq 10$), which is the number of points in the polyline. Following this are p pairs of integers $x_1, y_1, x_2, y_2, \ldots, x_p, y_p$ ($0 \leq x_i, y_i \leq 100\,000$), which denote that this wire's polyline is defined by $(x_1, y_1) - (x_2, y_2) - \cdots - (x_p, y_p)$. It is guaranteed that $x_1 < x_p$ and $x_1 \leq x_i \leq x_p$ for $2 \leq i < p$. Furthermore, $y_1 = y_p = 0$ and $y_i > 0$ for $2 \leq i < p$.

All points in the input are distinct.

Output

Display the largest utility over all safe subsets of the given wires.

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

Sample Input 2

```
5
1 3 0 0 50 50 100 0
1 3 45 0 50 40 110 0
1 3 10 0 20 10 30 0
1 3 20 0 30 10 40 0
1 3 35 0 45 10 50 0
```

Sample Output 2

```
4
```

Sample Input 3

```
5
1 3 0 0 50 50 100 0
1 3 45 0 50 40 110 0
1 3 10 0 20 10 30 0
1 3 20 0 30 10 40 0
10 3 35 0 45 10 50 0
```

Sample Output 3

```
11
```

Problem F
Far, Far Away
Time limit: 8 seconds

As you know, the purpose of a journey is not to reach the destination but to make the trip. You want to make a trip with as many legs as possible. However, each leg of the trip costs some money and your budget is limited. Find the longest path!

Source: Wikimedia Commons

Input

The first line contains two integers n ($1 \leq n \leq 100$), which is the number of locations, and m ($1 \leq m \leq 10^9$), which is the amount of available money.

The next n lines describe the legs. Each of these lines contains n integers, where the j^{th} integer of the i^{th} line c_{ij} is the cost of a leg from location i to location j ($1 \leq c_{ij} \leq 10^9$ for each $1 \leq i, j \leq n$). Your trip must start at location 1 and may end at any location.

Output

Display the maximum number of legs a trip from location 1 can have such that the sum of the costs of its legs is at most m. A trip may repeatedly visit the same location (including start and destination) and repeatedly use the same leg (paying its cost multiple times).

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

Sample Input 2	Sample Output 2
3 9 2 5 4 1 2 1 1 1 2	6

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

Sample Input 4	Sample Output 4
2 3 1 10 10 10	3

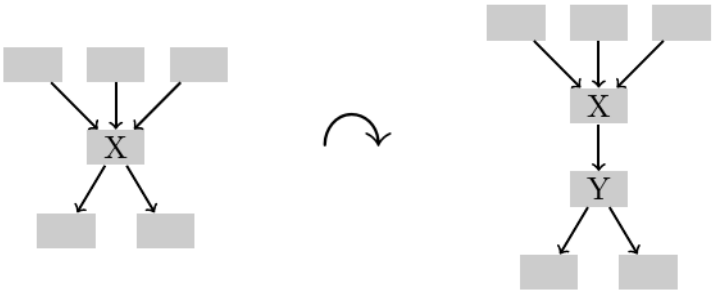
This page is intentionally left (almost) blank.

Problem G
Good Cable Management
Time limit: 5 seconds

Clara has worked herself into a bit of a cable mess. It all started when she got a great deal on a bulk order for cables and switchboxes. Each switchbox has incoming and outgoing connectors. A cable connects an outgoing connector of one switchbox with an incoming connector of another switchbox. The cable allows communication between two switchboxes. Note that communication is one-way: from one switchbox, out an outgoing connector through the cable to the other switchbox through its incoming connector.

Initially, Clara's network was simply a single switchbox. But over time, she extended her network by inserting new cables and switchboxes as needed using the following two types of upgrades.

- Length Upgrade on switchbox x:
  She picks an existing switchbox x, unplugs all the outgoing cables from x and replugs them into the outgoing connectors of a new switchbox y. Afterwards she adds a new cable from an outgoing connector of x to an incoming connector of y.



- Parallel Upgrade on switchbox x:
  She takes a new switchbox y and *duplicates* an existing switchbox x. That is, if there was a connection with switchbox x (either incoming or outgoing), then she connects that switchbox with y. Note that Clara does not add a new cable between x and y.

Unfortunately, her network of switchboxes and cables has grown to the point where it becomes difficult to tell which switchboxes can communicate with one another. Two switchboxes can communicate if there is a directed path from one switchbox to the other (in either direction). Can you help her?

Input

The input starts with a line containing two integers N ($1 \leq N \leq 200\,000$), which is the number of upgrades Clara has done, and Q ($1 \leq Q \leq 200\,000$), which is the number of queries.

The next N lines describe the upgrades. Each of these lines starts with either the character `L`, for a length upgrade, or `P`, for a parallel upgrade. This is followed by x ($1 \leq x \leq N$), which is the index of the switchbox that Clara upgrades. The newly added switchbox is labelled with y, which is the smallest positive integer not already used. It is guaranteed that x was in the network prior to this upgrade ($x < y$). The initial switchbox is labelled with 1.

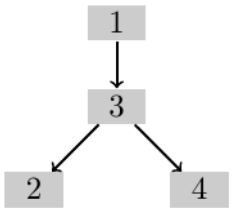
The next Q lines describe the queries. Each of these lines contains two integers u and v ($1 \leq u < v \leq N+1$), which are the two switchboxes for this query.

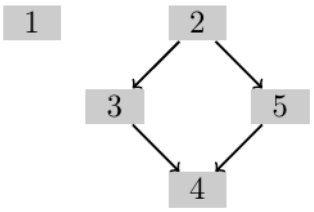
Output

Display the number of queries for which the two switchboxes can communicate.

Figure of Sample Inputs

The left subfigure shows the resulting network after applying the upgrades in Sample Input 1. The right subfigure similarly shows Sample Input 2.





Sample Input 1

```
3 2
L 1
L 1
P 2
1 4
2 3
```

Sample Output 1

```
2
```

Sample Input 2

```
4 2
P 1
L 2
L 3
P 3
1 5
3 5
```

Sample Output 2

```
0
```

Problem H
Hazardous Driving
Time limit: 6 seconds

When driving a hire car in the UK in winter, it has sometimes struck me that the navigation system's option of avoiding major roads is almost the opposite of what I want. Major roads tend to be less hazardous, being more likely to be cleared of snow in cold winters and less likely to be flooded in warm winters.

I need to get to Hazel's house for afternoon tea. Given my emphasis on safety, each road will have a hazard rating and a length. I want a route that minimises the maximum hazard rating encountered on the route. Out of all the routes that minimise the maximum hazard rating encountered, I want one that minimises the total length of the route. Each road is two-way. There is at least one route from my house to Hazel's house. What is an optimal route to get from my house to Hazel's house?

Source: Pexels

Input

The first line contains 4 integers N ($2 \leq N \leq 200\,000$), which is the number of locations, M ($1 \leq M \leq 200\,000$), which is the number of roads, S ($1 \leq S \leq N$), which is the location of my house, and E ($1 \leq E \leq N$), which is the location of Hazel's house (and is not equal to S).

The next M lines describe the roads. Each of these lines contains 4 integers A ($1 \leq A \leq N$), which is one endpoint of the road, B ($1 \leq B \leq N, A \neq B$), which is the other endpoint of the road, H ($1 \leq H \leq 10^8$), which is the hazard rating of the road, and L ($1 \leq L \leq 10^8$), which is the length of the road.

Output

Display the maximum hazard rating of an optimal route and its total length.

Sample Input 1

```
4 5 1 4
1 2 1 5
2 4 2 10
1 3 2 5
3 4 2 5
1 4 5 4
```

Sample Output 1

```
2
10
```

Sample Input 2

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

Sample Output 2

```
1
4
```

Sample Input 3

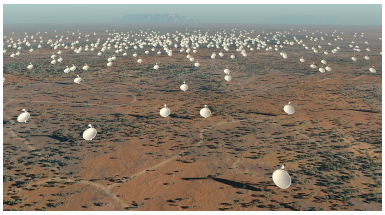
```
2 2 1 2
1 2 3 4
2 1 2 6
```

Sample Output 3

```
2
6
```

This page is intentionally left (almost) blank.

Problem I
Interesting World of Arrays
Time limit: 3 seconds



Source: Wikimedia Commons

Gwen is just about to finish her Ph.D. entitled *The Interesting World of Arrays*. She studies many different kinds of arrays, but her favourite type is *counting arrays*. A counting array simply counts the number of times each possible value appears in an array. Formally, $[c_0, c_1, c_2, \ldots, c_{n-1}]$ is the counting array of $A = [a_0, a_1, \ldots, a_{n-1}]$ if there are c_0 zeroes in A, c_1 ones in A, c_2 twos in A, and so on. For example, if $A = [4, 1, 2, 0, 2]$, then its counting is $[1, 1, 2, 0, 1]$. Counting arrays are not defined unless a_i is an integer and $0 \leq a_i < n$ for all $0 \leq i < n$.

The last chapter of Gwen's thesis is on *mod-m self-describing arrays*. Let $A = [a_0, a_1, \ldots, a_{n-1}]$ be an array with counting array $[c_0, c_1, \ldots, c_{n-1}]$. For a positive integer m, the array A is a mod-m self-describing array if $a_i \equiv c_i \pmod{m}$ for all $0 \leq i < n$. That is, a_i and c_i leave the same remainder when divided by m. For example, consider $A = [6, 6, 4, 6, 3, 5, 3]$ and its counting array $[0, 0, 0, 2, 1, 1, 3]$. Since they are the same modulo 2 (both become $[0, 0, 0, 0, 1, 1, 1]$), the array A is said to be a mod-2 self-describing array.

The only thing left to do before Gwen submits her thesis is to compute the number of mod-m self-describing arrays for various values of n and m. Please help compute these numbers.

Input

The input consists of a single line containing two integers n ($1 \leq n \leq 12$), which is the length of the array, and m ($2 \leq m \leq 10^9$), which is the modulus.

Output

Display the number of mod-m self-describing arrays of length n.

Sample Input 1	Sample Output 1
4 3	6

Sample Input 2	Sample Output 2
5 3	20

Sample Input 3	Sample Output 3
7 4	72

This page is intentionally left (almost) blank.

Problem J
Judging Divisionals
Time limit: 1 second

After the 2018 South Pacific Divisionals, twelve teams advance to the South Pacific Regionals (you are one of those 12 teams! Congratulations!). Each team competing in the Divisionals is part of either the Central, Eastern or Western division. The teams are ranked from lowest to highest (lower is better). The 12 teams are selected in two selection steps:

- Selection Step I: For each division, the best-ranked team from this division is selected (say that they are from University X), then the next best-ranked team from this division that is not from University X is selected. A total of 6 teams are selected this way.
- Selection Step II: Then 6 more teams are selected by repeatedly taking the the best-ranked team such that (a) they have not already been selected and (b) at most one other team from their university has already been selected.

Place	Team Name	University	Division	Qualified for Regional Finals?
1st	Team 1	University of A	Eastern	Yes — First place university in Eastern Division
2nd	Team 2	University of B	Central	Yes — First place university in Central Division
3rd	Team 3	University of C	Western	Yes — First place university in Western Division
4th	Team 4	University of B	Central	Yes — Selection Step II (1/6)
5th	Team 5	University of D	Western	Yes — Second place university in Western Division
6th	Team 6	University of E	Eastern	Yes — Second place university in Eastern Division
7th	Team 7	University of C	Western	Yes — Selection Step II (2/6)
8th	Team 8	University of F	Central	Yes — Second place university in Central Division
9th	Team 9	University of G	Central	Yes — Selection Step II (3/6)
10th	Team 10	University of H	Eastern	Yes — Selection Step II (4/6)
11th	Team 11	University of I	Western	Yes — Selection Step II (5/6)
12th	Team 12	University of C	Western	No — Two teams from University of C already advanced
13th	Team 13	University of H	Eastern	Yes — Selection Step II (6/6)
14th	Team 14	University of A	Eastern	No — Only 12 teams will advance via Selection Step I and II

Which teams are selected to advance to the Regionals?

Input

The first line of input contains a single integer n ($12 \leq n \leq 100$), which is the number of teams that competed in the Divisionals.

The next n lines describe the teams (from lowest ranked first to highest ranked last). Each of these lines contains three strings which are the name, division, and university of a team, respectively. Each of these strings contains only lowercase and uppercase letters and consists of at least 1 and at most 100 characters. The division is one of `Central`, `Eastern` or `Western`.

It is guaranteed that it is possible to select the 12 teams given the above selection steps. All universities have distinct names and each university belongs to exactly one division. All teams have distinct names.

Output

Display the twelve teams that are selected, sorted from lowest to highest ranking.

Sample Input 1

```
14
TeamA Eastern UniversityA
TeamB Central UniversityB
TeamC Western UniversityC
TeamD Central UniversityB
TeamE Western UniversityD
TeamF Eastern UniversityE
TeamG Western UniversityC
TeamH Central UniversityF
TeamI Central UniversityG
TeamJ Eastern UniversityH
TeamK Western UniversityI
TeamL Western UniversityC
TeamM Eastern UniversityH
TeamN Eastern UniversityA
```

Sample Output 1

```
TeamA
TeamB
TeamC
TeamD
TeamE
TeamF
TeamG
TeamH
TeamI
TeamJ
TeamK
TeamM
```

Sample Input 2

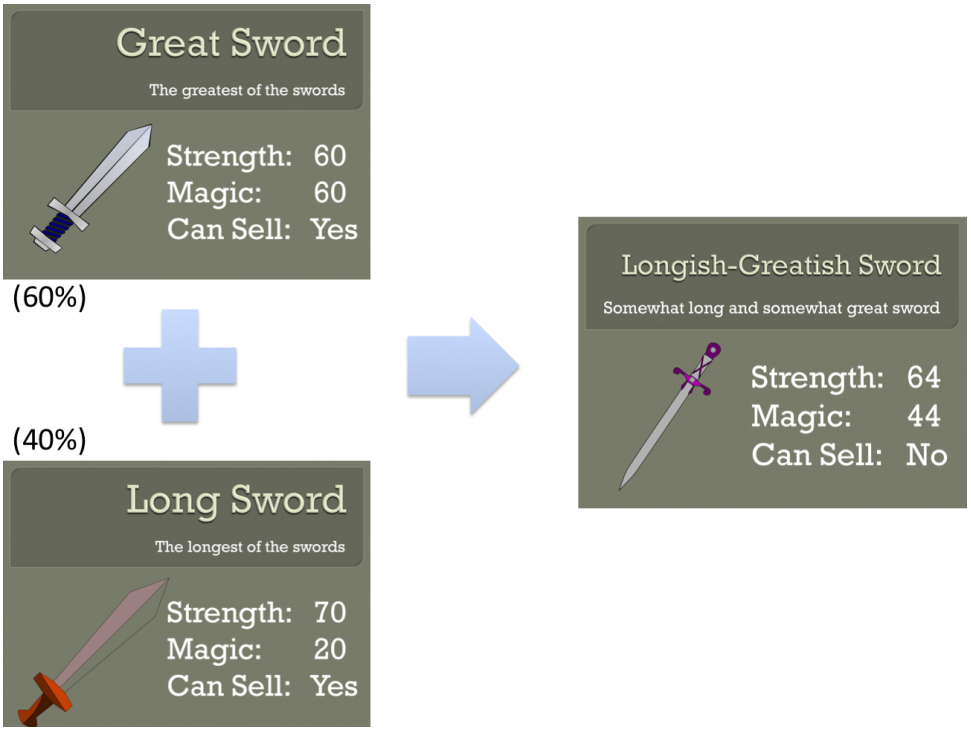
```
20
FJCVTZS Central TheUniversityofNewSouthWales
Ooneonn Western UniversityofMelbourne
MessyUniversity Eastern TheUniversityofAuckland
WhereIsAnand Western UniversityofMelbourne
MaxWard Western TheUniversityofWesternAustralia
stinkypete Western TheUniversityofAdelaide
excuseme Central TheUniversityofSydney
numonash Western MonashUniversity
Semicolon Central UniversityofWollongong
CodeBenders Central UniversityofTechnologySydney
CSEplusplus Central TheUniversityofNewSouthWales
averageatbest Central TheUniversityofNewSouthWales
maeth Central TheUniversityofNewSouthWales
TensorMellow Western TheUniversityofAdelaide
coolbeans Central TheUniversityofSydney
ThreeHeadsNoIQ Central TheUniversityofNewSouthWales
MonteCarloMethodists Eastern MasseyUniversity
YouSee Eastern UniversityofCanterbury
sublimatetk Central TheUniversityofNewSouthWales
Delta Western MonashUniversity
```

Sample Output 2

```
FJCVTZS
Ooneonn
MessyUniversity
WhereIsAnand
MaxWard
stinkypete
excuseme
numonash
Semicolon
CodeBenders
CSEplusplus
MonteCarloMethodists
```

Problem K
Knights and Dragons
Time limit: 4 seconds

Max is playing Knights and Dragons. Throughout the game, Max collects a bunch of swords. Each sword has three attributes: its strength, its magic and its sellability (if it can be sold at a market). Later, Max acquires a spell called the *Sword Twister*. This spell takes two swords and makes a new sword (both old swords are still usable after the spell and their attributes do not change). To cast the spell, Max chooses a percentage and the Sword Twister makes a new sword whose strength and magic are just the weighted average of the original swords. The chosen percentage must be strictly between 0% and 100% (non-inclusive) and does *not* need to be an integer. The new sword is not sellable.



Since Max has this new spell, this makes some of his swords useless, so he may sell them at the market. For example, if he currently owns a sword with 64 strength and 44 magic, then he can sell that sword because he can use the Sword Twister on the Long Sword (40%) and the Great Sword (60%) to create the Longish-Greatish Sword, which has the same strength and magic. Max can use the Sword Twister as many times as he wishes. For example, if he also had a sword with 36 strength and 0 magic, then he could use the Sword Twister on this sword (50%) and the newly forged Longish-Greatish Sword (50%) to create a sword with 50 strength and 22 magic.

A sword is deemed replaceable if it is sellable and a sword with exactly the same strength and magic can be made via a sequence of Sword Twisters. Which of Max's swords are replaceable?

Input

The input starts with a line containing a single integer n ($1 \leq n \leq 200\,000$), which is the number of swords that Max has originally.

The next n lines describe the swords. Each of these lines contains two integers s ($0 \leq s \leq 10^9$), which is the strength of the sword, and m ($0 \leq m \leq 10^9$), which is the magic of the sword. All n swords are sellable and no two swords have the exact same strength and magic value.

Output

Display if each sword is replaceable or not in the same order as the input without any spaces and on a single line. Display `Y` if a sword is replaceable and `N` otherwise.

Sample Input 1	Sample Output 1
3 45 55 80 20 59 41	NNY

Sample Input 2	Sample Output 2
4 60 60 70 20 50 22 36 0	NNYN

Problem L
Laps
Time limit: 2 seconds

Source: John Jennings

Bethany is training for an athletics contest. Yesterday, she ran laps around an oval athletics track. She recorded her location on the track every minute. She has now forgotten how many laps she did yesterday.

Bethany has given you this list of locations on the track. Each location is the number of metres Bethany has run since the last time she passed the start point. Bethany always starts at the start point. Note that it is possible that Bethany does not move between two entries in the list. However, she will never run backwards. The athletics track is n metres long. This means that if Bethany ran n metres in total she would return to the start point.

Suppose you have a 300 metre long track and Bethany ran 200 metres from the start point. Bethany would record 200. Then, if she ran another 200 metres, she would record 100 having crossed the start point.

She wants to determine the minimum number of laps she could have completed. Can you help?

Input

The first line of input contains two integers n ($1 \leq n \leq 10^9$) and m ($1 \leq m \leq 10^5$), the length of the track and the number of locations Bethany recorded in her list. The next line contains m integers each of which is at least 0 and at most $n - 1$. These are the list of locations Bethany recorded. The list is in increasing order of time.

Output

Display the minimum number of laps Bethany could have completed.

Sample Input 1	Sample Output 1
1 1 0	0

Sample Input 2	Sample Output 2
4 2 2 1	1

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

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

This page is intentionally left (almost) blank.