ICPC SOUTH PACIFIC MOCK REGIONALS

OCTOBER 17, 2020

Read Me First

- The real Regional Contest will have two different problem sets, one for division 1 and one for division 2. This contest only has one problem set, however some problems are marked as division 1 difficulty, and others are marked as division 2 difficulty. Questions that are not marked as either division 1 or division 2 are at a difficulty level that could show up in either contest.
- This contest should be attempted by a team of 3 competitors using a single computer. If it is not possible to share a single computer (due to COVID-19 restrictions), then only one computer should be used at a time. No Internet access other than to the DOMJudge server and to programming language API documentation should be used. This will be policed during the real Regional Contest.
- This problem set contains 12 problems labeled from A to L.
- If you are new, start with problem A, which is designed to help new contestants learn.
- Some of the problems contain hints that should be helpful for newer contestants.

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:

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

Definition 5

A positive number is a number that is strictly greater than 0, while a negative number is strictly less than 0. This means that zero is neither negative nor positive. A number is nonpositive if it is not positive and a number is nonnegative if it is not negative.

Problem A
Question Number +1 (start with this one!)
Time limit: 2 seconds

Note that this question should be friendly for division 2 competitors.

The TL;DR of this question is to take a positive integer as input, add one to it, and output the resulting number.

If this is your first contest, then there a few things that you should know. First, to solve this question, you need to write a program in C++, Java, or Python3. The program needs to take input from the standard input stream, and produce output on the standard output stream. If you do not know what these are, feel free to look up your language's documentation. If you are still stuck, consider asking your coach. Note that during the real Regional Contest, you will *not be able to do this*. Because this is a mock contest, it is acceptable for this first question. However, it *not* OK to ask for help for any other questions in this problem set (until after the contest is over)! Remember that people are taking this contest seriously as practice, and the scoreboard should be an accurate reflection of performance.

You can submit your solution program to DOMjudge using the green "Submit" button at the top. After you submit, your program will get run on several test cases. These include the samples provided below, and also some secret test cases. There are a few verdicts you are likely to see. These include the following.

- CORRECT: Which means that you solved the question and got a point!
- WRONG-ANSWER: There was at least one test case where your program produced the wrong output
- TIMELIMIT: Your program took too long to run on some test cases. Usually there is a limit of a few seconds per test case. Usually this will happen if you use an algorithm with the wrong complexity. For example, using an $O(N^2)$ algorithm when $O(N \log N)$ is required. Make sure you read the bounds in the Input section carefully, and do not expect an slow algorithm to pass a question with large bounds!
- RUN-ERROR: Your program had a runtime error. Typically an exception was thrown, or there was a segmentation fault (invalid memory access).
- MEMORY-LIMIT: Your program ran out of memory. We allow at most 1GB of memory usage.
- NO-OUTPUT: Your program produced no output.
- COMPILER-ERROR: Your program failed to compile. Make sure you try to compile it on your end!

If you get a wrong answer verdict for this question, take a careful look at the Input section. The input number can be quite large. It can be so large that the result does not fit in a 32-bit integer! You might need to use a larger integer type (like `long long` in C++ or `long` in Java). If you are still stuck, you can use the "request clarification" button on DOMjudge. This button should only be used if you think there is something that is wrong during that contest. It is *not* for getting hints if you are stuck. For example, you should use it if you think a problem statement contains something ambiguous or contradictory, but you should not use it if you are stuck while trying to solve the problem. The only exception that that rule is this question, which we want to make sure everyone solves!

We suggest doing problem E after this problem. It contains more useful tips for new competitors.

Input

The input contains a single integer A ($1 \leq A \leq 4294967295$).

Output

Display $A + 1$.

Sample Input 1	Sample Output 1
1	2

Sample Input 2	Sample Output 2
999	1000

Problem B
Banking
Time limit: 2 seconds

Internet banking sites have a variety of methods to authenticate their users. The methods usually involve passwords or Personal Identification Numbers (PINs) together with a mechanism to verify that a person is attempting to authenticate rather than a computer program.

The Actuarial Commerce Merchant bank has a scheme where, when you login, you are provided with a "pattern word", containing only upper and lower case letters. You must use this pattern word to extract and sum digits from your PIN as follows.

Letters in the pattern word are to be interpreted as numbers, with a (or A) = 1, b (or B) = 2, ... z (or Z) = 26. A lower case letter specifies a count of digits to extract from the PIN while an upper case letter specifies a counts of digits to be skipped. The letters in the pattern word are processed from left to right resulting in a sequence of extracted digits, which are added together to yield a number. You then enter that number into a field on the web page form to authenticate yourself. For example, if your PIN was `1093373`, and the pattern provided to you was `aBcA` you would extract one digit (namely 1) skip two digits (09), extract 3 digits (337) and then skip 1 digit (3), before totalling the extracted digits (1337) and entering 14 into the field on the web page form.

The bank allows you to have a PIN containing up to 256 digits and they intend to provide a pattern word in which the letters, when interpreted as numbers, sum to the length of the PIN. However, sometimes they get this wrong!

Write a program that reads a PIN and a pattern word and outputs the sum of the digits extracted from the PIN if the pattern is valid or outputs `non sequitur` if the length of the PIN and the length indicated by the pattern are different.

Input

The input contains a single test case.

The first line of input will contain an n-digit PIN, $6 \leq n \leq 256$. The second line will contain an m-digit pattern word containing only upper and lower case letters, $1 \leq m \leq 256$.

Output

The test case will produce one line of output being either the sum of the extracted digits from the PIN if the pattern word is valid or the text `non sequitur` if the pattern is invalid.

Sample Input 1	Sample Output 1
103373 aAcA	14

Sample Input 2	Sample Output 2
092384907653 bGc	23

Sample Input 3	Sample Output 3
092384907653 bGb	non sequitur

This page is intentionally left (almost) blank.

Problem C
Amazing Sushi
Time limit: 2 seconds

Mary and Marty were playing with their Marvelous Marble Machine and have become hungry. So, they decided to order some sushi. There are several types of sushi. The sushi platter comes with pieces of various types (possibly many of each type).

Both Mary and Marty know a range of the total number of pieces of sushi they can eat. They would like to know if there is a way to distribute their sushi such that neither of them eats too little or too much sushi and no sushi goes uneaten. To be fair, Mary and Marty each want to eat half of the pieces of each type. If there is an odd number of pieces for a given type, either one of them can eat the extra piece.

Does there exist a way for Mary and Marty to properly distribute their sushi?

Input

The first line contains a single integer n ($1 \leq n \leq 100$), which is the number of types of sushi.

The second line describes the number of pieces of sushi Mary can eat. This line contains two integers, x_1 ($0 \leq x_1 \leq 100\,000$) and y_1 ($x_1 \leq y_1 \leq 100\,000$). Mary must eat at least x_1 pieces and at most y_1 pieces.

The third line describes the number of pieces of sushi Marty can eat. This line contains two integers, x_2 ($0 \leq x_2 \leq 100\,000$) and y_2 ($x_2 \leq y_2 \leq 100\,000$). Marty must eat at least x_2 pieces and at most y_2 pieces.

The next n lines describe the n different types of sushi in Mary and Marty's platter. Each line contains a single integer m ($1 \leq m \leq 1\,000$), which is the number of pieces of sushi of this type.

Output

If there exists a way for Mary and Marty to properly distribute their sushi, display `Yes`. Otherwise, display `No`.

Sample Input 1	Sample Output 1
3 1 10 7 20 5 3 1	No

Sample Input 2	Sample Output 2
3 1 10 3 20 5 3 14	Yes

Sample Input 3	Sample Output 3
3 1 10 3 20 5 3 16	No

Problem D
Shelob's Lair
Time limit: 10 seconds

Note that this question is recommended only for division 1.

Sam Gamgee and Frodo Baggins are trapped in Shelob's lair. Shelob is a gigantic spider who lives in the caves at the edge of Mordor. Sam and Frodo are Hobbits, which means that they are little people with hairy feet.

The cave is a large rectangular cavern and Shelob has cast many great webs in the cave, and now Frodo and Sam (who are at the South wall of the cave) must reach the North wall to escape. If Sam and Frodo touch any of the web, they will become stuck and Shelob will come and eat them.

Sam has a magic sword, Sting, that can cut through Shelob's web. However, Frodo is poisoned and Sam is exhausted from their adventures, so Sam only has the strength to make one vertical slice through the web once. A well-chosen slice at a point will cut through all of the webs that pass through the point, allowing Sam and Frodo to pass. Sam and Frodo, being little people, can fit through an infinitely small slit.

Given the locations of all the webs in the cave, determine if it is possible for Sam and Frodo to escape.

We suppose that each web is a vertical sheet that runs from one point (given by Cartesian coordinates) to another point. The webs are fixed at the roof and the floor of the cave, and run in a straight line between the two points. Multiple webs can cross one another (Shelob is a skilled web spinner) and if Sam were to slice exactly where they cross he could slice all of the webs at once. No web touches the North or South wall of the cave. Sam is also able to cut at precisely the point one or more webs connect to the East or West walls of the cave. You may treat Frodo and Sam as a point, so they can fit through the vertical cut and can fit between two webs that do not intersect.

Input

The input contains a single test case.

The first line consists of three integers, w (the width of the cave), d (the depth of the cave), and n (the number of webs that are cast), where $1 \leq w, d \leq 1000$ and $1 \leq n \leq 500$.

Next, n lines follow where each line contains 4 integers, $x_1,\ y_1,\ x_2,\ y_2$, where $0 \leq x_1, x_2 \leq w$ and $0 < y_1, y_2 < d$. (x_1, y_1) is the Cartesian coordinates of one end of the web, and (x_2, y_2) is the Cartesian coordinate at the other end of the web. The coordinates are arranged so that the South-West corner of the cave is the point $(0, 0)$, and the North-East corner is the point (w, d).

Output

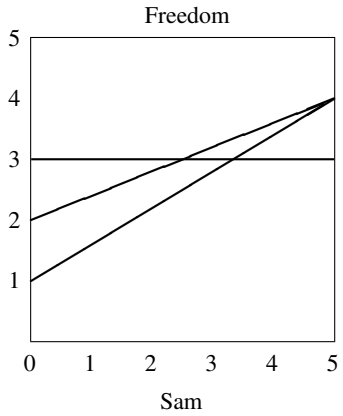
If it is possible for Frodo and Sam to reach the North wall, making at most one slice through the web, print the line:

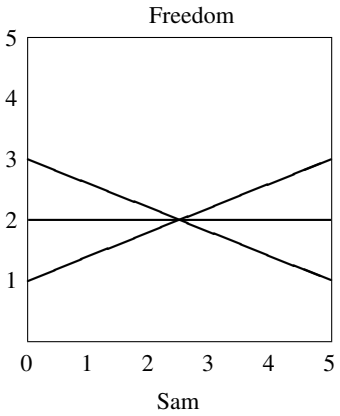
```
We can make it Mr Frodo!
```

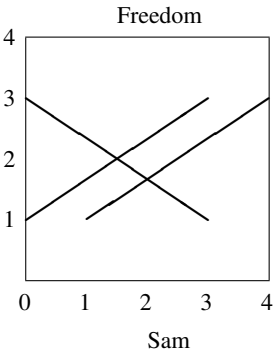
If it is impossible for Frodo and Sam to reach the North wall without making more than one slice through the web, print the line:

```
We're doomed Mr Frodo!
```

The following diagrams are images for the Sample Inputs:







Sample Input 1

```
5 5 3
0 3 5 3
0 1 5 4
0 2 5 4
```

Sample Output 1

```
We're doomed Mr Frodo!
```

Sample Input 2

```
5 5 3
0 1 5 3
0 3 5 1
0 2 5 2
```

Sample Output 2

```
We can make it Mr Frodo!
```

Sample Input 3

```
4 4 3
0 1 3 3
0 3 1 3
4 3 1 1
```

Sample Output 3

```
We can make it Mr Frodo!
```

Problem E
Easiest Problem
Time limit: 1 second

Note that this question should be friendly for division 2 competitors.

When doing ICPC contests, one of the most important skills is correctly determining the difficulty of each problem quickly. To get the best placement in an ICPC contest, it is in a team's best interest to solve the easiest problem first and work on the harder problems later.

To ensure that they find the easiest problem, all three members of Team Merlin will read all of the problems and rank each one in terms of difficulty. Each ranking is an integer between 1 and 100, where 1 is the easiest and 100 is the hardest. A problem's *difficulty score* is the average of the three team members' rankings. The problem with the lowest difficulty score is deemed to be the *easiest problem*. In the case of a tie, the problem that received the lower ranking from the team captain is the easiest problem. It is guaranteed that the captain will not give two problems the same rank.

What is the easiest problem?

Hints for new contestants: Sometimes it can be difficult to figure out which questions are easy and which are hard. For example, a question might seem easy, but have subtle edge cases, or require a clever algorithm to avoid a timelimit verdict. One strategy you can use to help determine which questions are easier is to check the scoreboard on DOMjudge. A question that everyone is solving is likely to be easier. If a question has not been solved by many teams, or has a lot of red submissions, it's likely to be harder. Notice that the teams on the scoreboard are ranked by number of questions solved, then by time penalty. Time penalty is the sum of the submission times (in minutes) for solved problems, plus a penalty of 20 minutes for each incorrect submission.

Input

The first line of the input contains a single integer P ($1 \leq P \leq 15$), which is the number of problems in the contest.

The next P lines describe the problems. Each line starts with a string, which is the name of the problem. Then follow three integers a, b, c ($1 \leq a, b, c \leq 100$), which are the three rankings of the members of Team Merlin. The team captain's ranking is a. The problem's name uses only lowercase and uppercase letters and consists of between 1 and 50 characters inclusive.

The P problem names will be distinct. Problem names are case-sensitive.

Output

Display the name of the easiest problem.

Sample Input 1	Sample Output 1
2 AnEasyProblem 5 3 7 AHardProblem 92 81 97	AnEasyProblem

Sample Input 2	Sample Output 2
12 AustralianVsAmerican 10 10 10 BombsAhoy 20 20 20 CrypticClues 25 25 20 DubiousRecording 35 23 15 EasiestProblem 1 1 1 Flow 30 30 30 GodsNumber 70 75 80 Holiday 34 15 88 IslandOfLove 100 100 100 JuiceMachine 50 12 32 KrazyTaxi 23 30 30 LoveActually 99 99 99	EasiestProblem

Problem F
Protest
Time limit: 3 seconds

A group of militant IT students want to march from the Computer Science building to the university Vice Chancellor's office to protest the removal of BASIC as the first-year programming language. The students plan to march arm-in-arm in rows to present a unified front. Once a row locks arms, they will stay that way until their demands are met.

The protest planners have mapped out all the walkways on campus to figure out how many students wide each walkway is. All walkways on the campus allow foot traffic in both directions. The planners can see from the campus map that there is a path from the Vice Chancellor's office from the Computer Science building but they need your help with their plans.

As the protesters' leading algorithmist they need you to find a path from the Computer Science building to the Vice Chancellor's office that maximises the number of students who can walk arm-in-arm for the duration of the walk.

Input

The input contains a single test case.

The first line contains two integers p $(2 \leq p \leq 1\,000)$ and w $(1 \leq w \leq 50\,000)$ specifying the number of unique walkway end points and the number of walkways on the university campus.

The second line contains two integers c and v $(0 \leq c, v < p; c \neq v)$ being the walkway end points for the computer science building and the vice chancellor's office respectively.

The following w lines contain three integers w_{start}, w_{end} $(0 \leq w_{start}, w_{end} < p; w_{start} \neq w_{end})$ and w_{width} $(1 \leq w_{width} \leq 1\,000)$ being the start and end points for a walkway and the number of students that will be able to walk arm-in-arm along that particular walkway.

Output

On a single line output the maximum number of students who can walk arm-in-arm from the Computer Science building to the Vice Chancellor's office.

Sample Input 1	Sample Output 1
7 11 3 5 0 1 15 0 2 23 1 2 16 1 3 27 2 4 3 2 6 21 3 4 14 3 5 10 4 5 50 4 6 9 5 6 42	16

This page is intentionally left (almost) blank.

Problem G
Flow
Time limit: 2 seconds

Note that this question should be friendly for division 2 competitors.

Maximus Aquius, the Roman aqueduct themed supervillain, is trying to irrigate the streets of Rome. He plans to do this by building a network of aqueducts that direct water from the surrounding countryside to the city. There are a number of water sources around Rome, each producing a fixed volume of water each hour. Each source of water has some altitude, and water can only flow downhill. If a water source has a lower altitude than the city itself, it cannot be used to irrigate the city.

Maximus Aquius, having no faith in your ability to solve complex problems, has hired a more talented computer scientist to compute the set of additional aqueducts he needs to build. Having now decided on his plan, he needs to figure out the logistics necessary to build the network.

Your task is to determine how many labourers must be hired to construct the aqueducts to ensure they are completed before Caesar returns from Gaul. Each labourer can build exactly 1 cubit of aqueduct each day. Multiple labourers can work on the same aqueduct at the same time and multiple aqueducts can be worked on at the same time, but a worker can only work on a single aqueduct on any given day. Given the length of each of the aqueducts (in cubits), what is the minimum number of labourers that must be employed to build all aqueducts before Caesar returns?

Input

The first line of input contains two integers m ($1 \leq m \leq 100\,000$), which is the number of aqueducts to be built, and n ($1 \leq n \leq 100\,000$), which is the number of days until Caesar returns.

The next m lines describe the aqueducts. Each line contains a single integer ℓ ($1 \leq \ell \leq 1\,000$), which is the length of this aqueduct in cubits.

Output

Display the minimum number of labourers required.

Sample Input 1

```
5 10
2
2
2
2
2
```

Sample Output 1

```
1
```

Sample Input 2

```
6 16
18
3
5
4
1
1
```

Sample Output 2

```
2
```

This page is intentionally left (almost) blank.

Problem H
Jumping Impala
Time limit: 8 seconds

Note that this question is recommended only for division 1.

Vlad the impala is both excited and nervous, as in a few months he will be participating in the annual challenge event where his herd will choose its leader for the coming year. As the eldest, he will be the first to undertake the challenge and will become the new leader if he succeeds. If Vlad does not succeed, the second eldest will then undertake the challenge and so on.

The impalas want a leader who can lead them safely to grazing grounds. Along the way, there are many predators and the challenge event is designed to test skills needed to avoid predators. Past volcanic activity has created a crocodile infested circular lake with a lushly vegetated circular island located exactly on the lake's centre. The challenge is to go from the outer edge of the lake to the island, graze for a couple of hours, then come back to the outer edge and finally make a second trip out to the island and again return back to the outer edge. Impalas are so frightened by the crocodiles that if an impala touches the water, it freezes and awaits its fate. Fortunately, there are some (unit radius) circular stones in the lake that impalas may use. By using these, it is not necessary to go to the island in a single bound from the lake's outer edge. Unfortunately, not only can crocodiles grab an impala in the water, but they can also grab an impala from a stone (but not the island or outer edge of the lake). However, Vlad has noticed that the crocodiles, which are slow by comparison with a leaping impala, always position themselves to exploit the impala tendency to reuse a previous route, e.g. returning from grazing on the island via the same stones as used on the way to the island. Indeed, the crocodiles never grab an impala from a stone that it is visiting for the first time. Vlad aims to exploit this weakness in the crocodile strategy. (As the first to attempt the challenge, he's not in a position to become leader by the usual manner—being the lucky impala that happens to attempt the challenge when the crocodiles are too full of previous challenge participants to be interested in another!)

An impala can leap (up to) a certain distance any number of times and turn arbitrarily tightly, so Vlad is working on improving his maximum leap distance and wants to know what it needs to be for him to be able to go to and from the island twice in safety (i.e. without ever revisiting any stone or touching the water). As Vlad has failed to "get real", he wants to know the relevant integer value, e.g. if he needs to be able to leap at least 2.01, you'll have to tell him 3 to be on the safe side. You can assume that Vlad can leap from the exact edge of one object to the exact edge of another and that it is safe to leap over a previously visited stone without landing on it.

Input

The input will contain a single test case.

The input starts with a line containing the radius of the lake, L ($4 \leq L \leq 10^9$), the radius of the island, R ($1 \leq R \leq L - 3$), and the number of stones in the lake, S ($4 \leq S \leq 1\,000$). The next S lines contain the information about the stones, one stone per line. Each line describing a stone contains two integers, the $x-$ and $y-$coordinates of its centre, regarding the lake centre as the origin (each coordinate will have absolute value no more than 10^9). These stones will be fully in the water. Stones may touch each other, the island, or the lake's outer edge, but there will be no overlaps between these items.

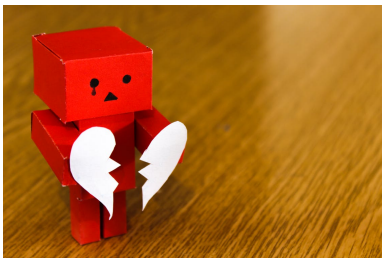
Output

Output the integer value Vlad wants to know. You can be assured that Vlad has to jump a positive distance in order to complete the challenge—otherwise, it wouldn't be a challenge!

Sample Input 1	Sample Output 1
10 2 5 -6 0 0 -6 0 6 3 0 9 0	4

Sample Input 2	Sample Output 2
10 2 5 -6 0 0 -6 0 6 3 1 9 0	5

Problem I
Island of Love
Time limit: 4 seconds



Island of Love is an exciting new game show. On this show, many of the contestants start as friends. The show is quite a high-pressure situation. It involves long bouts of relaxing by the pool, drinking lots of beer, gossiping, and strenuous arguing. Since it is such a stressful situation, people stop being friends as the season progresses. All friendships are bidirectional. That is, if person X is friends with person Y, then person Y is also friends with person X. Also, as the show goes on, people never become friends, they only ever stop being friends (yes, it is depressing).

Two people are on *speaking terms* if they are friends or if there is some sequence of friendships that connect the two people. For example, if person X is only friends with person Y and person Y is friends with person Z, then person X and person Z are on speaking terms. However, if person Y and person Z end their friendship, then person X and person Z are no longer on speaking terms since there is no sequence of friendships connecting them.

Your job is to process some friendship ending events, as well as determine if two people are on speaking terms at certain points throughout the season. All the events and queries will be in the order of the time that they occur. You can assume they all happen at distinct times. A friendship ending event results in two people no longer being friends.

Input

The first line of input contains three integers N ($1 \leq N \leq 10^5$), which is the number of people on the show, F ($0 \leq F \leq 10^5$), which is the number of distinct pairs of friends when the show begins, and Q ($0 \leq Q \leq 10^5$), which is the number of events and queries to process.

The next F lines describe the friendships. Each of these lines contains two integers X and Y ($1 \leq X < Y \leq N$), which denote that person X is friends with person Y. It is guaranteed that all pairs are distinct.

The next Q lines describe the friendship ending events and queries in the order they occurred. Each of these lines contains three items. The line starts with a single character t (t is either `E` or `S`). Then follow two integers X and Y ($1 \leq X < Y \leq N$). If t is `E`, then there was a friendship ending event between person X and person Y (it is guaranteed that they were friends before this event). If t is `S`, then you must determine if person X and Y are on speaking terms at this point in time.

Output

For each query in the given order, display if the two people are on speaking terms at that given time.

Sample Input 1	Sample Output 1
3 3 4 1 2 1 3 2 3 E 1 2 S 1 2 E 1 3 S 1 2	YES NO

Sample Input 2	Sample Output 2
4 4 5 1 2 1 3 2 3 3 4 E 1 2 S 1 4 E 2 3 S 1 4 S 2 4	YES YES NO

Problem J
Juice Machine
Time limit: 2 seconds

Constantina is trying to buy some juice from a vending machine in her home country of Ayeseepeesee. She has to pay for the juice using coins. In this country, they have a frustrating coin system. While most countries have coins which are a certain number of cents (that is, a fraction in the form $\frac{x}{100}$), Ayeseepeesee takes this further and uses fractions with a different denominator (for example, $\frac{1}{3}$).

Assuming that Constantina has an unlimited supply of several different types of coins, what is the minimum number of coins she must use to pay for her juice? She must pay the exact amount for the juice. Note that we are interested in the minimum number of coins used, not the minimum number of coin types used.

Input

The input starts with a line containing two integers N ($1 \leq N \leq 50$), which is the cost of the juice, and C ($1 \leq C \leq 10$), which is the number of coin types Constantina has.

The next C lines describe the coin types she has. Each of these lines contains two integers n and d ($1 \leq n, d \leq 10$), which denote that Constantina has an unlimited supply of coins with value $\frac{n}{d}$.

Output

Display the minimum number of coins needed to pay the exact amount for the juice. If it is impossible to pay for the juice, display `-1`.

Sample Input 1	Sample Output 1
3 2 8 7 3 7	7

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

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

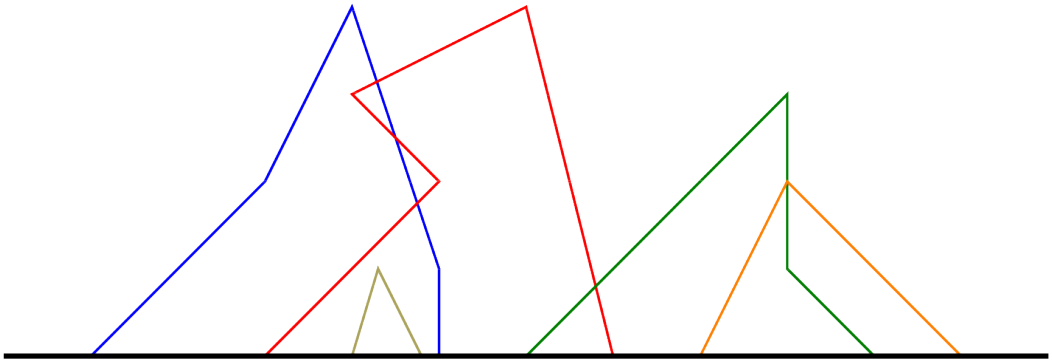
This page is intentionally left (almost) blank.

Problem K
Explosive Wiring
Time limit: 5 seconds

Note that this question is recommended only for division 1.

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 L
Almost an Anagram
Time limit: 2 seconds

Andy loves anagrams. For the uninitiated, an anagram is a word formed by rearranging the letters of another word, for example `rasp` can be rearranged to form `spar`. Andy is interested to know if two words are almost anagrams. A word is almost an anagram of another word if:

- one word is shorter than the other by one letter but otherwise contains the same letters in any order; or
- the two words are the same length and their character multisets differ by one character only e.g. "`aaa`" and "`aab`"

Your job is to help Andy to determine if two words are identical, anagrams, almost anagrams or nothing like each other.

Input

The input contains a single test case.

The input will be a single line of text containing a pair of words separated by a single space. The words will be in lower case and will contain alphabetic characters only. Words will contain between 1 and 1000 letters inclusive.

Output

Your program should produce one line of output as follows:

- If the words are identical, output: $\texttt{word}_a$ `is identical to` $\texttt{word}_b$
- If the words are anagrams, output: $\texttt{word}_a$ `is an anagram of` $\texttt{word}_b$
- If the words are almost anagrams, output: $\texttt{word}_a$ `is almost an anagram of` $\texttt{word}_b$
- Otherwise, output: $\texttt{word}_a$ `is nothing like` $\texttt{word}_b$

In all cases the first word in the output sentence must be the shorter word or if the words are the same length the first word must be the lexicographically least.

Sample Input 1	Sample Output 1
`spar rasp`	`rasp is an anagram of spar`

Sample Input 2	Sample Output 2
`table able`	`able is almost an anagram of table`

Sample Input 3	Sample Output 3
`table sable`	`sable is almost an anagram of table`

This page is intentionally left (almost) blank.