ACM ICPC SOUTH PACIFIC DIVISIONALS

OCTOBER 14, 2017

Contest Problems

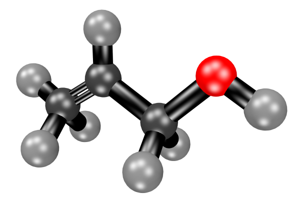
This contest contains twelve problems. Twelve teams will advance to the Regional Finals.

For problems that state "*Your answer should have an absolute or relative error of less than* 10^{-6}", your answer, a, will be compared to the correct answer, b. If $|a - b| < 10^{-6}$ or $\frac{|a-b|}{|b|} < 10^{-6}$, then your answer will be considered correct.

Judging team:
Daniel Anderson, Darcy Best, Mike Cameron-Jones, Malcolm Corney, Zac Forman, Tim French, Andrew Gozzard, Walter Guttmann, Andrew Haigh, Richard Lobb, Evgeni Sergeev, Kevin Tran, Max Ward-Graham, Peter Whalan

Problem A
Atomic Mass
Time limit: 1 second

Elements in chemistry are represented by their symbol, which is either one uppercase letter or an uppercase letter followed by a lowercase letter (for example, H is hydrogen and He is helium). A compound is a combination of several elements. Numbers appearing after an element symbol indicate that an element is used multiple times. For example, H_2O is made up of two hydrogen (H) and one oxygen (O), while CH_4 is one carbon (C) and four hydrogen (H). Note that a symbol can appear in multiple locations of a compound. For example, acetic acid is CH_3COOH.



Source: Pixabay

The atomic mass of an element is the mass of one atom of that element. The molecular mass of a compound is the sum of the atomic masses of the elements in the compound.

For example, the atomic mass of hydrogen (H) is 1.01 and the atomic mass of oxygen (O) is 16.00, which means that the molecular mass of water (H_2O) is $2 \cdot 1.01 + 16.00 = 18.02$.

What is the molecular mass of the given compound?

Input

The input starts with a line containing a single integer n ($1 \leq n \leq 20$), which is the number of elements.

The next n lines describe the elements. Each line contains the element's symbol followed by a real number m ($0.01 \leq m \leq 500.00$), which is the mass of this element. The symbol will be either a single uppercase letter or an uppercase letter followed by a lowercase letter. No two elements will have the same symbol. All masses will be specified to exactly two decimal places.

The last line describes the compound in question. The compound will be a string of element symbols and integers. Each integer will appear directly after the corresponding element symbol, and each integer will be between 2 and 100 inclusive. The compound consists of between 1 and 25 characters inclusive. All elements that appear in the compound are one of the n elements listed above. The molecule will not be a complex molecule such as $Al_2(SO_4)_3$.

Output

Display the molecular mass of the compound. Your answer should have an absolute or relative error of less than 10^{-6}.

Sample Input 1	Sample Output 1
2 H 1.01 O 16.00 H2O	18.02

Sample Input 2	Sample Output 2
2 H 1.01 C 12.01 CH4	16.05

Sample Input 3

```
2
Cl 35.45
Na 22.99
NaCl
```

Sample Output 3

```
58.44
```

Sample Input 4

```
13
Al 26.98
C 12.01
Cd 112.41
I 126.90
F 19.00
N 14.00
O 16.00
Pa 231.04
S 32.07
Si 28.08
Th 232.04
U 238.03
V 50.94
SOUThPaCIFICdIVISiONAlS
```

Sample Output 4

```
1568.27
```

Problem B
Bake Off
Time limit: 8 seconds

Source: Pexels

Davy decided to start a weekend market stall where he sells his famous cakes. For the first market stall, Davy decided to bake n cakes. Each cake is described by its deliciousness and the flavours it contains, which is a (possibly empty) subset of the flavours {caramel, cherry, chocolate, cinnamon, coconut, cookies}.

Because of Davy's skill in baking, he has a line of m customers when he opens his stall who wish to buy a cake. Davy will serve them in the order they are lined up. Each customer has a required subset of flavours they would like in their cake, but are happy to receive additional flavours in their cake. Davy will give each customer the most delicious cake left that contains at least the flavours that the customer has asked for. You should help Davy determine which cake to sell to each customer (or if there is no cake that satisfies that customer's requirements, in which case, they buy nothing).

Input

The first line contains two integers n ($1 \leq n \leq 300\,000$), which is the number of cakes, and m ($1 \leq m \leq 100\,000$), which is the number of customers.

The next 6 lines describe the flavours contained in the cakes. The first of these lines contains a string of length n, which describes if caramel is in each cake. This string will contain a `1` in the ith position if cake i contains caramel and `0` otherwise. The second through sixth of these lines will describe cherry, chocolate, cinnamon, coconut and cookies, respectively, in the same format. The cakes are numbered from left to right, starting with cake 1 on the left. No two cakes have the same deliciousness and are sorted by their deliciousness, with cake 1 being the least delicious and cake n being the most delicious.

The next 6 lines describe the flavours requested by the customers. The first of these lines contains a string of length m, which describes if each customer has requested caramel in their cake. This string will contain a `1` in the ith position if customer i requested caramel and `0` otherwise. The second through sixth of these lines will describe cherry, chocolate, cinnamon, coconut and cookies, respectively, in the same format.

Output

Display the number of the cake purchased by each customer in the order that they are requested. If a customer does not purchase a cake, display `-1` for them instead.

Sample Input 1

```
4 2
0001
1111
0001
1111
0001
1111
01
11
01
11
01
11
```

Sample Output 1

```
4 -1
```

Sample Input 2

```
3 4
000
000
000
010
101
110
0000
0000
0000
0010
1000
0100
```

Sample Output 2

```
3 2 -1 1
```

Problem C
Clever Title
Time limit: 5 seconds


Source: Pexels

One day, I sat down in my chair and wondered, *"Which theorem is the best?"* After an hour on my favourite search engine (Ask Jeeves), I stumbled across the BEST Theorem. (The BEST Theorem gives a formula for the number of Eulerian circuits in directed graphs—though, that is not important for this problem.) The BEST Theorem is named after the authors of the paper: de **B**ruijn, van Aardenne-**E**hrenfest, **S**mith and **T**utte.

A title is *clever* if can be formed by selecting one uppercase letter from each author's name (in any order) to get the title. Note that this implies that the number of letters in the title must be the same as the number of authors.

Last year, the problem setters of the South Pacific Region proved a theorem about number of k-th powers modulo n, and we would like to give it a clever title. The problem setters' names are Anderson, Best, Cameron-Jones, Corney, French, Guttmann, Haigh, Lobb, Sergeev, Tran, Ward-Graham and Whalan. Unfortunately, while the "ABCCFGHLSTWW Theorem" does include one letter from each author's name, ABCCFGHLSTWW is not a valid word (in the real world, a valid word is an English word—in this problem, the set of valid words will be given). Moreover, no matter how you permute the names, there are no valid words that can be formed by using the uppercase letters from the problem setters' names. However, if just Anderson, French, Sergeev and Tran were to discover a theorem, they could call their theorem the FAST Theorem (**F**rench, **A**nderson, **S**ergeev and **T**ran). Interestingly, this is the only permutation of those names that can be used for the FAST Theorem. In contrast, imagine the authors were Merlin, Oberon and Othello, and these authors wished to call their theorem the MOO Theorem. There are two permutations (**M**erlin, **O**beron and **O**thello, as well as **M**erlin, **O**thello and **O**beron) of their names that can make the MOO Theorem clever.

Usually the ordering of names on the publication for a theorem is important. However, since I already have tenure, all I care about is counting the number of permutations that can make a title clever. Given the authors of a theorem, and a list of valid titles for that theorem, count the number of permutations of author names that can be used to make each title clever.

Input

The input starts with a line containing two integers n ($1 \le n \le 10$), which is the number of authors, and k ($1 \le k \le 1\,000$), which is the number of valid words.

The next n lines describe the authors. Each of these lines contains a string, which is the name of the author. The author's name uses only lowercase and uppercase letters, and consists of between 1 and 20 characters, inclusive. There is at least one uppercase letter in the author's name.

The next k lines describe the valid words. Each of these lines contains a string, which is the valid word. Each valid word uses only uppercase letters and consists of between 1 and 20 characters, inclusive.

Output

Display k integers, the number of valid orderings of author names for each title respectively.

Sample Input 1

```
4 1
deBruijn
vanAardenneEhrenfest
Smith
Tutte
BEST
```

Sample Output 1

```
1
```

Sample Input 2

```
10 3
Anderson
Best
CameronJones
Corney
French
Guttmann
Haigh
Lobb
Sergeev
Tran
DOESTHISWORK
TOOSHORT
THISISTOOLONG
```

Sample Output 2

```
0
0
0
```

Sample Input 3

```
4 2
Anderson
French
Sergeev
Tran
SLOW
FAST
```

Sample Output 3

```
0
1
```

Sample Input 4

```
3 1
AA
BB
AB
AAB
```

Sample Output 4

```
2
```

Problem D
Deck of Cards
Time limit: 5 seconds


Source: Pexels

While on a break from making the problems for Regional Finals, the South Pacific Programming Judges frequently like to play their favourite card game, Uno. In the current game, there are just two players left, Malcolm and Richard, who are trying to figure out the optimal strategy to win. In this version of Uno, each player has a hand consisting of cards, each of which has a colour: red (R), green (G), blue (B) or yellow (Y) and a numeric value $1 - 9$.

The players alternate taking turns to play one card from their hand. On any turn, the card played must either be the same colour as the previous card that was played or have the same numeric value, or both (there may be duplicates of the same card). The first player who does not have any cards that can be played (nothing in their hand has the same colour or numeric value as the previous card that was played) loses. Since the judges prefer games of skill to games of chance, they have decided to play the game with a rule such that every player can see every player's hand. It is currently Malcolms's turn to play. Given the current card on the table (the most recently played card), determine who will win the game if both Malcolm and Richard play optimally.

Input

The first line of input contains two integers m ($1 \leq m \leq 1\,000$), which is the number of cards in Malcolm's hand, and r ($1 \leq r \leq 1\,000$), which is the number of cards in Richard's hand.

The second line describes Malcolm's hand. This line contains m strings describing the cards in Malcolm's hand. Each card has a colour (either `R`, `G`, `B` or `Y`) and a numeric value (either `1`, $\ldots$, `8` or `9`).

The third line describes Richard's hand. This line contains r strings describing the cards in Richard's hand. Each card is described in the same way as above.

The fourth line of input contains a single string which describes the most recently played card in the same way as above.

Output

Display the winner of the game assuming both players play optimally.

Sample Input 1	Sample Output 1
3 2 B4 B5 Y2 B3 B7 B2	Malcolm

Sample Input 2	Sample Output 2
4 5 G1 Y3 R3 R9 B3 B9 G4 R8 Y1 R5	Richard

Sample Input 3

```
1 1
B1
G7
R5
```

Sample Output 3

```
Richard
```

Problem E
Extra Judicial Operation
Time limit: 8 seconds


Source: Pexels

The Suitably Protected Programming Contest (SPPC) is a multi-site contest in which contestants compete at specified sites, which are linked by the Notorious Broken Network (NBN) of two-way links between some pairs of individual sites. There is at most one link directly between any pair of sites and there is no link from a site to itself. Contestants submit their solutions to a judging server to get them tested. The event will use one or more judging servers. Each judging server is located at a contest site and may be accessed directly from that site or through a sequence of linked sites.

At any time during the contest, each contestant must have access to at least one judging server. The links between sites are set up in such a way that if all links work properly, it would suffice to operate one judging server. At the other extreme, having each site operate its own judging server would guarantee access even if all of the links failed. The organisers wish to minimise the number of judging servers that they operate, while still maintaining the integrity of the contest.

Examination of the NBN's performance over time reveals some good news. The NBN is reliably unreliable! At any given time, exactly one link is broken (that is, communications cannot travel in either direction on that link). Given this fact, the organisers want you to work out the minimum number of judging servers that must operate so that no matter which link is broken, every contestant still has access to at least one judging server.

As they know that they must operate at least one judging server, they ask you to tell them how many extra judging servers they must operate.

Input

The first line of the input contains a single integer S ($2 \leq S \leq 100\,000$), which is the number of sites. The sites are numbered from 0 to $S - 1$.

The next S lines describe the sites. Each of these lines starts with an integer ℓ ($0 \leq \ell < S$), which is the number of links that this site has to a site with a larger number. Then follow ℓ integers in ascending order, which are the sites to which those links go. Note that, although links are two-way, they are only mentioned in the lower numbered site's line. The total number of links is no more than $100\,000$.

Output

Display the minimum number of extra judging servers the organisers must operate.

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

This page is intentionally left (almost) blank.

Problem F
Front Nine
Time limit: 6 seconds

Andrew is building a 2D golf course called the Front Nine where each hole will be exactly n units long. In order to generate many (hopefully) unique holes, he will do it with a randomised algorithm. A single hole is defined by a function $y : [0, n] \to [0, h]$ which for each x-coordinate in the range $[0, n]$ gives the height of the hole's terrain (between 0 and h) at that point.

Andrew's randomised algorithm for generating a single hole is as follows:

Step 1 Set $y(0) = a$.

Step 2 For each $i = 1, 2, \ldots, n$, let

$$y(i) = \text{fix}(y(i-1) + r(i))$$

where $r(i)$ is a random integer chosen from the set $\{-1, 0, 1\}$ with probabilities P_{-1}, P_0 and P_1 percent, respectively ($P_{-1} + P_0 + P_1 = 100$). And fix is a function defined by

$$\text{fix}(y) = \begin{cases} 0 & \text{if } y < 0, \\ y & \text{if } 0 \le y \le h, \\ h & \text{if } h < y \end{cases}$$

which clamps (restricts) its output to the range $[0, h]$.

Step 3 Once we have $y(i)$ for each $i = 0, 1, ..., n$ we fill in each interval $(i, i+1)$ of the function with the straight line that joins the points $(i, y(i))$ and $(i+1, y(i+1))$.

Step 4 All of the area under y is filled in with dirt.

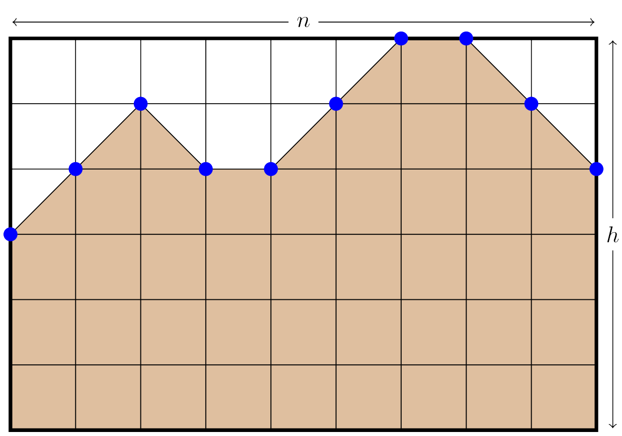


Figure F.1: An example with $n = 9$, $h = 6$, $a = 3$. The area of dirt is 42.5. One possible function r: $r(1) = 1$, $r(2) = 1$, $r(3) = -1$, $r(4) = 0$, $r(5) = 1$, $r(6) = 1$, $r(7) = 1$, $r(8) = -1$, $r(9) = -1$.

Since Andrew wants to build many holes, he needs to know how much dirt he will need. Help him by determining the expected area under the terrain for each hole.

Input

The input consists of a single line containing six integers n ($1 \le n \le 100\,000$), which is the length of the hole, h ($0 \le h \le 100$), which is the maximum height of the hole, a ($0 \le a \le h$), which is the height at $x = 0$,

P_{-1} ($0 \leq P_{-1} \leq 100$), P_0 ($0 \leq P_0 \leq 100$) and P_1 ($0 \leq P_1 \leq 100$), which are the probability of r being -1, 0 and 1, respectively. In addition, $P_{-1} + P_0 + P_1 = 100$.

Output

Display the expected area under the terrain. Your answer should have an absolute or relative error of less than 10^{-6}.

Sample Input 1	Sample Output 1
`4 10 3 100 0 0`	`4.5000000000`

Sample Input 2	Sample Output 2
`2 10 5 50 0 50`	`10.0000000000`

Problem G
Greedy Generosity
Time limit: 2 seconds

Source: Pexels

Many currencies are designed such that, if you have a sufficient supply of coins of each denomination, a greedy approach will enable you to make up a specified amount of change using the fewest coins possible for that currency. The greedy approach repeatedly picks the largest coin type not exceeding the amount left. This problem involves such a suitably designed currency, one with 1c, 2c, 5c, 10c, 20c and 50c coins. If we were to make change for 77c, for example, the greedy approach would pick 50c, then 20c, then 5c and finally 2c.

In recent years, some shops have been installing self-service checkout machines into which customers can insert money, with the machine then giving them any change required. The shops wish to encourage customers to use these machines. Here we consider a possible way for the shops to encourage customers to use their machines, drawing from an idea in gambling machines into which people also put money and out of which they also tend to get less than they put in.

Every self-service machine has a stock of each type of coin and will start the day at its *standard level*. The machine will always make change using the greedy approach described previously. However, if it does not have enough of each coin type to do this, then it will reward the shopper with *greedy generosity*.

The generous aspect of the machine is as follows: if the machine cannot give coins in the greedy fashion, then the amount of change will be increased by as little as needed to allow it to be given by the greedy algorithm. For example, if a machine that only contains two 5c coins, one 20c coin and zero 10c coins is required to give 10c change, it will give the 20c coin as change. After such a payout, there will be buzzers and flashing lights (keeping with the gambling machine spirit), and the staff will immediately adjust the machine's coin stock back to its standard level.

Your task is to simulate such a machine and to work out how much extra money that it gives the customers. The machine will need to process several transactions in order. The machine will start stocked at its standard level. The money that the customer inserts is added to the stock of coins that it has (there is no limit for how many coins the machine can have) before attempting to give change. The coins paid out to customers are removed from the machine's stock. Each customer will pay an amount that is at least as much as the value of the sale, and if they pay more than this, then the machine must make change for the difference. It is guaranteed that the largest coin a customer will put in will exceed the value of the correct change.

Input

The first line of input contains six integers $C_1, C_2, C_5, C_{10}, C_{20}, C_{50}$ ($0 \le C_i \le 100$), which are the number of coins of each denomination when the machine is stocked at its standard level.

The second line of input contains a single integer T ($1 \le T \le 1\,000$), which is the number of transactions to process.

The next T lines describe the transactions. Each of these lines contains seven integers V ($1 \le V \le 99$), which is the value of the sale, d_1, d_2, d_5, d_{10}, d_{20}, d_{50} ($0 \le d_i \le 100$), which are the number of coins of each type that the customer puts into the machine.

Output

Display the total amount (in cents) extra that the machine pays over the course of the transactions.

Sample Input 1

```
0 0 2 0 1 0
1
60 0 0 0 0 1 1
```

Sample Output 1

```
10
```

Sample Input 2

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

Sample Output 2

```
5
```

Problem H
Hole in One
Time limit: 5 seconds


Source: Pexels

Everyone loves mini-golf! The aim of the game is to hit a ball from a starting location into a hole, possibly bouncing off objects on the way.

You would like to get some more practice in, so you are going to write a computer program that will help you. You need to figure out whether you can get the ball in the hole with just one shot. Since you are still an amateur, you are only comfortable taking a shot that will bounce off at most one object before landing in the hole. The course is a rectangle with N obstacles on it, each of which are straight line segments in the course. You may assume that the obstacles are infinitely thin and the ball and hole are points.

Given a description of the course, the location of the hole and the starting location of the ball, you need to determine whether or not it is possible to get the ball into the hole in one shot such that the ball bounces at most once. The ball can bounce off any of the N obstacles or the 4 walls. However, the ball may not touch the endpoint of any obstacle. Note that if the ball touches the intersection of two objects (either obstacles or walls), then this will count as hitting two objects.

Input

The first line of input contains three integers N $(0 \leq N \leq 1\,000)$, which is the number of obstacles on the course, W $(3 \leq W \leq 100)$, which is the width of the course, and H $(3 \leq H \leq 100)$, which is the height of the course.

The second line describes the ball. This line contains two integers x $(1 \leq x < W)$ and y $(1 \leq y < H)$, which are the coordinates of the ball.

The third line describes the hole. This line contains two integers x $(1 \leq x < W)$ and y $(1 \leq y < H)$, which are the coordinates of the hole.

The next N lines describe the obstacles. Each of these lines contains four integers x_1 $(0 \leq x_1 \leq W)$, y_1 $(0 \leq y_1 \leq H)$, x_2 $(0 \leq x_2 \leq W)$ and y_2 $(0 \leq y_2 \leq H)$, which indicates that there is an obstacle that is a straight line with endpoints (x_1, y_1) and (x_2, y_2). It is guaranteed that $(x_1, y_1) \neq (x_2, y_2)$.

The ball and the hole will be at distinct locations and will not be touching any object (wall or obstacle). It is possible that the objects (obstacles and walls) intersect each other. However, they will not overlap at more than one point.

Output

If it is possible to get the ball into the hole with at most one bounce, display `YES`. Otherwise, display `NO`.

Sample Input 1	Sample Output 1
0 10 10 1 1 5 5	YES

Sample Input 2	Sample Output 2
1 10 10 1 1 9 9 1 9 9 1	NO

Sample Input 3	Sample Output 3
1 10 10 1 5 9 5 5 6 5 8	YES

Sample Input 4	Sample Output 4
2 20 4 1 2 19 2 0 1 20 1 0 3 20 3	YES

Sample Input 5	Sample Output 5
2 10 10 1 1 9 9 4 4 6 6 4 6 6 4	NO

Problem I
Incomplete Book
Time limit: 1 second

Source: Pexels

Meorge Arr Arr Gartin, the pirate, is currently writing a series of amazing novels. Full of inspiration, his first novel only took him k days to write. However, as time went on, he started writing slower and slower. In particular, if it took him ℓ days to write the ith book in the series, then it will take him 2ℓ days to write the $(i+1)$th book.

Because of how slow he is writing the series, fans are worried that he will not be around long enough to finish the series before he dies. What is the maximum number of books that he can finish before he dies?

Input

The input consists of a single line containing two integers k ($1 \le k \le 365$), which is the number of days needed to write the first book, and d ($k \le d \le 10^9$), which is the number of days after he started writing the first book that he will die.

Output

Display the maximum number of books that he can finish.

Sample Input 1	Sample Output 1
`1 1`	`1`

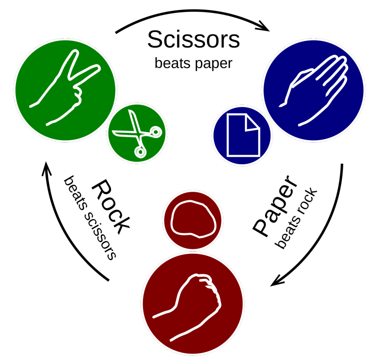
Sample Input 2	Sample Output 2
`1 2`	`1`

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

Sample Input 4	Sample Output 4
`117 1337`	`3`

This page is intentionally left (almost) blank.

Problem J
Jupiter Rock-Paper-Scissors
Time limit: 2 seconds



Source: Wikipedia

Next week is the Rock-Paper-Scissors World Championships between the top 2 contestants in the world: Alice and Bob! But this is not your typical Rock-Paper-Scissors tournament—this is *Jupiter Rock-Paper-Scissors*. The basics of the game are still the same: rock-beats-scissors, scissors-beats-paper, paper-beats-rock, but the luck of the game has been removed. Prior to the contest, each player selects a string of n moves, where each move is either `R`, `P` or `S`. We will call Alice's move string A and Bob's move string B.

The game is played in three setup phases, then one play phase. The phases happen in order: first Phase 1, then Phase 2, then Phase 3, followed by the Play Phase.

In Phase 1, Alice removes some moves from the beginning of A (possibly none) and some moves from the end of A (possibly none) so that she is left with exactly k moves. In Phase 2, Bob removes some moves from the beginning of B (possibly none) and some moves from the end of B (possibly none) so that he is left with exactly k moves.

In Phase 3, Alice selects ℓ consecutive moves from A and *morphs* them. Morphing a move means changing every rock-to-paper, paper-to-scissors and scissors-to-rock.

In the Play Phase, the players now play their k moves from left-to-right (using normal Rock-Paper-Scissors rules). The player that reaches m wins first is awarded two points. If neither player reaches m wins, then both players are awarded one point. Each player's goal is to maximise their point total.

For example, say that $n = 8$, $k = 4$, $\ell = 2$ and $m = 1$. Here is one possible play-through of this game.

	Alice	**Bob**
Original Moves	R P S S P R P R	S S P R S S R S
Phase 1	S S P R	S S P R S S R S
Alice removes 2 moves from the beginning of A and 2 moves from the end of A		
Phase 2	S S P R	S P R S
Bob removes 1 move from the beginning of B and 3 moves from the end of B		
Phase 3	R R P R	S P R S
Alice morphs the first two moves in A		

Then the Play Phase occurs. The first move is a win for Alice (rock-beats-scissors) and since $m = 1$, only one win is needed, so Alice is awarded two points and the game stops. Note that once Alice made her decision in Phase 1, there is no way for Bob to get two points (or even one point) since no matter what Bob decides to do in Phase 2, Alice always has a way to get the two points.

In all phases, both players have full knowledge of the other player's move string and all decisions made in previous phases. Assuming both players play optimally, who wins?

Input

The first line of input contains four integers n ($1 \leq n \leq 50$), which is the number of moves, k ($1 \leq k \leq n$), which is the value for Phase 1 and Phase 2, ℓ ($0 \leq \ell \leq k$), which is the value for Phase 3, and m ($1 \leq m \leq k$), which is the number of moves that a player must win in order to get two points.

The second line contains the string A, which is Alice's move string. The third line contains the string B, which is Bob's move string. Both move strings will have length n and consist of only `R`, `P` and `S`.

Output

Display the name of the winner that receives the most points assuming both players play optimally. If both players receive the same number of points, display `Draw` instead.

Sample Input 1

```
3 3 0 1
SRP
RSS
```

Sample Output 1

```
Bob
```

Sample Input 2

```
4 4 1 1
SRPR
RSSS
```

Sample Output 2

```
Alice
```

Sample Input 3

```
3 2 0 2
RRP
RRS
```

Sample Output 3

```
Draw
```

Sample Input 4

```
8 4 2 1
RPSSPRPR
SSPRSSRS
```

Sample Output 4

```
Alice
```

Problem K
Keeping Cool
Time limit: 2 seconds


Source: Pexels

Kevin has just gotten back to his car after a morning at the beach and is about to drive away when he realises that he has left his ball somewhere. Thankfully, he remembers exactly where it is! Unfortunately for Kevin, it is extremely hot outside and any sand that is exposed to direct sunlight is very hot. Kevin's pain tolerance allows him to only run for at most k seconds in the hot sand at one time. Kevin runs at exactly 1 metre per second on hot sand.

Scattered around the beach are umbrellas. Each umbrella is a perfect circle and keeps the sand underneath it cool. Each time Kevin reaches an umbrella, he will wait there until his feet cool down enough to run for another k seconds on the hot sand. Note that Kevin will not run more than k seconds in the hot sand at one time, so if two umbrellas are more than k metres apart, Kevin will not run between them.

Determine the minimum amount of time that Kevin must be in the sun in order to retrieve his ball and return back to the car.

Input

The first line of input contains four integers n ($0 \leq n \leq 100$), which is the number of umbrellas, k ($1 \leq k \leq 100$), which is the number of metres that Kevin can run on the hot sand, x ($-100 \leq x \leq 100$) and y ($-100 \leq y \leq 100$), which are the coordinates of the beach ball. Kevin starts at his car at $(0, 0)$. You may treat Kevin and the ball as single points.

The next n lines describe the umbrellas. Each of these lines contains three integers x ($-100 \leq x \leq 100$), y ($-100 \leq y \leq 100$) and r ($1 \leq r \leq 100$). The umbrella is a circle centred at (x, y) with radius r.

There may be multiple items (ball, umbrella(s) or Kevin) at a single location. All measurements are in metres.

Output

Display the minimum amount of time (in seconds) that Kevin must be in the sun. If it is impossible for Kevin to get to the ball and return back to the car, display `-1` instead. Your answer should have an absolute or relative error of less than 10^{-6}.

Sample Input 1	Sample Output 1
`0 1 0 0`	`0.0000000000`

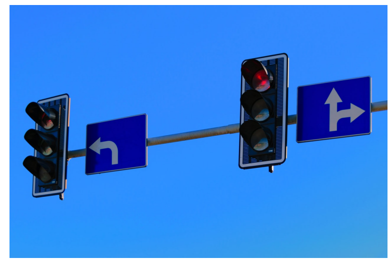
Sample Input 2	Sample Output 2
`0 20 1 2`	`4.4721359550`

Sample Input 3	Sample Output 3
`0 10 20 20`	`-1`

Sample Input 4	Sample Output 4
2 2 7 4 6 2 2 2 2 1	6.1289902045

Sample Input 5	Sample Output 5
1 2 3 3 0 3 2	4.0000000000

Problem L
Lights in the Morning
Time limit: 1 second


Source: Pexels

You woke up late this morning and are in a rush to get to work. You are the kind of person who gets mad if they get unexpectedly stuck at traffic lights, so you are going to expect the unexpected!

Your route to work is D kilometres long with N traffic lights along the way. Each traffic light has a red light and a green light. When the light is green, you may pass the traffic light, but when the light is red, you may not pass. When you leave your house, all of the traffic lights are red. For each traffic light, you know the first time that it will turn green. Once this happens, the traffic light will remain green for g minutes, then change to red for r minutes, then change back to green for g minutes, then red for r minutes and so on. These values may be different for each traffic light.

If your car arrives at a traffic light at the moment that it changes (either green-to-red or red-to-green), assume that you will make it through.

You travel at 1 kilometre per minute, so it takes D minutes to complete your journey. Determine whether or not you will get stopped at any of the traffic lights.

Input

The first line of input contains two integers N ($1 \leq N \leq 1\,000$), which is the number of traffic lights, and D ($2 \leq D \leq 10^9$), which is the length of the journey.

The next N lines describe the traffic lights. Each line contains four integers x ($1 \leq x < D$), which is the location of the traffic light in kilometres from your home, a ($1 \leq a \leq 10^9$), which is the number of minutes after leaving home that the traffic light first turns green, g ($1 \leq g \leq 10^9$), which is the number of minutes that the traffic light remains green on each cycle, and r ($1 \leq r \leq 10^9$), which is the number of minutes that the traffic light remains red on each cycle. The locations of the traffic lights are all distinct.

Output

If you will make it through all of the traffic lights without stopping, display `YES`. Otherwise, display `NO`.

Sample Input 1

```
1 10
5 3 3 3
```

Sample Output 1

```
YES
```

Sample Input 2

```
1 10
2 5 1 1
```

Sample Output 2

```
NO
```

Sample Input 3	Sample Output 3
3 20 5 4 2 3 10 1 2 2 15 10 2 2	YES

Sample Input 4	Sample Output 4
2 10 3 2 10 10 6 2 3 2	NO