

icpc International Collegiate Programming Contest



The 2025 ICPC North America Rocky Mountain Regional Contest

Official Problem Set



icpc titanium
multi-regional sponsor





NOVEMBER 8, 2025

Contest Problems

A : Adrift At Sea
B : Buying Jerseys
C : Conveyor Belt Sushi
D : Corporate Retreat
E : Draft Lottery
F : Energy Usage
G : Genies
H : Hexadecimal Confusion
I : Poetic Tournament
J : Spies vs. Spies
K : Team Practice
L : Video Games
M: What Time Is It Mr. Fox?

This contest contains 13 problems over 38 pages. Good luck.

For problems that state “*Your answer must have a relative error of less than 10^{-9}* ”, your answer, x , will be compared to the correct answer, y . If $\frac{|x-y|}{|y|} < 10^{-9}$, then your answer will be considered correct.

For problems that state “*Your answer must have an absolute error of less than 10^{-9}* ”, your answer, x , will be compared to the correct answer, y . If $|x - y| < 10^{-9}$, 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 $s_1s_2\cdots s_n$ is lexicographically smaller than $t_1t_2\cdots t_\ell$ 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 ($A, B, \dots, Z$).
 - Lowercase letters are the lowercase English letters ($a, b, \dots, z$).
-

This page is intentionally left blank.

Problem A

Adrift At Sea

Time limit: 1 second

You are developing a navigation system for sailing boats, but unfortunately the normal navigation systems do not work for a couple of reasons. First, instructions like “turn left” are meaningless because there are no roads. Such instructions must be followed by the amount to turn (in degrees). In addition, “left” and “right” are not used in naval navigation. Instead, “port” and “starboard” are used for “left” (counterclockwise) and “right” (clockwise), respectively.

Given the current heading and the desired heading of the sailing boat, your task is to provide an instruction to turn to the desired heading. All headings are given by the 8 compass directions: N, NE, E, SE, S, SW, W, NW. Each direction is 45 degrees clockwise from the previous one in this list. In addition to turning, it is also possible for the instruction to be “Keep going straight” or “U-turn” (go in the opposite direction). You may assume turns can be made instantaneously.

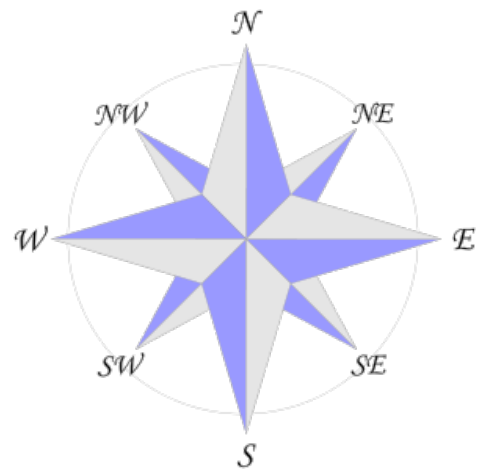


Image by [ElifQin](#). CC BY-SA 3.0

Input

The input consists of two lines. The first line is the current heading, and the second line is the desired heading. All headings are one of the 8 compass directions listed above.

Output

Output a single line in one of three forms:

- Keep going straight
- U-turn
- Turn d degrees dir

In the last case, d should be an integer between 1 and 179, and dir should be either the word port or starboard.

Sample Input 1

N	U-turn
S	

Sample Output 1

**Sample Input 2**

NW NW	Keep going straight
----------	---------------------

Sample Output 2**Sample Input 3**

N SW	Turn 135 degrees port
---------	-----------------------

Sample Output 3**Sample Input 4**

N E	Turn 90 degrees starboard
--------	---------------------------

Sample Output 4

Problem B

Buying Jerseys

Time limit: 4 seconds

You have found a job managing a soccer league in rural Saskatchewan. This league is located several hundred kilometers from any major city, and so has capacity for only two teams. Because there is little else to do in rural Saskatchewan, many players have developed bitter personal rivalries with other players. These players will refuse to be put on a team with any of their rivals. Last season, your predecessor found a way to assign players to teams that respects their preferences. Since it is a new season with new sponsors, you must purchase new jerseys for the league. After looking into the cost of buying jerseys, you have realized that last season's teams may be very expensive to support. Thus, you have decided to re-assign teams in order to minimize costs.

- Players on one team require jerseys, which cost one dollar per player.
- Players on the other team play without jerseys, at no cost.
- You may also choose to remove some players from the league. Removing a player costs the league two dollars each because their entry fee has to be refunded.

Despite your best efforts, it may be the case that last season's team assignment may already have been as cheap as possible. Regardless, your job is to assign each player to either the jersey team, the no-jersey team, or remove them from the league, so that no pair of rivals ends up on the same team, and the total cost is minimized.

Input

The first line of input contains two integers N ($2 \leq N \leq 2000$) and M ($1 \leq M \leq 5000$) representing the number of players and number of rivalries. Each of the following M lines contains two integers u and v ($1 \leq u < v \leq N$) indicating that players u and v are rivals. It is guaranteed that each unique rivalry is listed only once. The final line contains a binary string S of length N representing the previous team assignment. That is, $S_i = 1$ if player i was on the jersey team, and $S_i = 0$ if player i was on the no-jersey team. It is guaranteed that this assignment of players to teams does not put any pair of rivals on the same team.

Output

Output a single integer representing the minimum total cost required to assign players to teams or remove them from the league.

Sample Input 1

```
4 1
1 2
0101
```

Sample Output 1

```
1
```

**Sample Input 2**

```
4 4
1 2
3 4
1 4
2 3
0101
```

Sample Output 2

```
2
```

Sample Input 3

```
8 7
1 2
1 3
1 4
1 5
2 6
2 7
2 8
01111000
```

Sample Output 3

```
3
```

Problem C

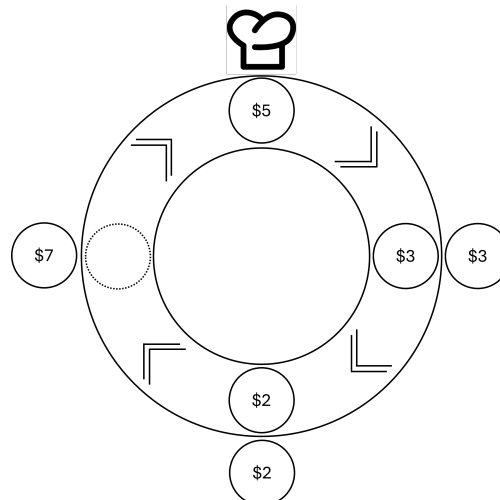
Conveyor Belt Sushi

Time limit: 2 seconds

After a successful contest, all the team members from your university have gathered at your local Sushi Shop to celebrate!

At the Sushi Shop, you and your teammates sit around a long, circular conveyor belt. The chef also has a designated location on the conveyor belt. When they see you enter (at time $t = 0$), the chef will begin putting sushi plates on the conveyor belt. Each second, the chef will attempt to place a new plate onto the belt in front of them. At the same time, a subset of the team members can take a sushi plate off of the belt in front of them. After that, the conveyor belt will advance all plates clockwise one seat. If a plate passes everyone without being taken, it returns to the chef the next second and makes another pass around the circle. The number of seats around the belt is one greater than the number of team members, as the chef has their own seat.

The conveyor belt is initially empty, but at time $t = 0$ seconds, the chef will place the first menu plate onto the belt. At time $t = 1$ second, the chef will place the second menu plate onto the belt. Each second, the chef will place the next menu plate if there is not already a plate in front of them. Once they have placed all menu plates onto the belt, they will start back at the first menu plate. If at any point there is already a plate in front of them, the chef will wait until the next opening on the belt to place the current plate.



Sample Input 1 after person 3 takes the \$7 sushi plate at time $t = 6$.

The Sushi Shop has developed a smart billing system that is meant to automatically track who takes which sushi and bill them properly. However, their system is down! Can you help them figure out how to split the bill?

Input

The first line of input contains an integer, M ($2 \leq M \leq 100$), the number of plates on the menu. The next line contains M integers, each detailing the cost, c_i ($1 \leq c_i \leq 100$), of the i -th plate in dollars.

The next line contains two integers, N ($1 \leq N \leq 100$) and E ($1 \leq E \leq 10\,000$), the number of team members and the number of events to process.

Each of the next E lines describes an event. Each event is described with two integers, t_j ($1 \leq t_j \leq 10\,000$) and p_j ($1 \leq p_j \leq N$), indicating that the person p_j seats clockwise from the chef takes the sushi plate in front of them off of the belt at time $t = t_j$ seconds. It is guaranteed that there is a sushi plate in front of person p_j .

Events are given in non-decreasing chronological order.

Output

Output N lines, detailing the bill for each team member. The first line should be the person one seat clockwise from the chef. From there, work clockwise around the conveyor belt to each team member.

Explanation for Sample Input 1

At time $t = 0$, the chef places a \$2 plate in front of them.

At time $t = 1$, the \$2 plate has rotated to be in front of person 1. The chef then places the \$3 plate.

At time $t = 2$, persons 1 and 2 take the \$3 and \$2 plates from in front of them. At the same time, the chef places a \$5 plate.

At time $t = 3$, the chef places the \$7 plate. The \$5 plate is now in front of person 1.

At time $t = 4$, the chef places a second \$2 plate. Persons 1 and 2 have \$7 and \$5 plates in front of them, respectively.

At time $t = 5$, the chef places a second \$3 plate. Persons 1, 2, and 3 have \$2, \$7, and \$5 plates in front of them.

At time $t = 6$, the \$5 plate has traveled fully around the conveyor belt. Because of this, the chef will wait for an open spot to place the next \$5 plate. However, person 3 will take the \$7 plate in front of them.

Sample Input 1

```
4
2 3 5 7
3 3
2 2
2 1
6 3
```

Sample Output 1

```
3
2
7
```



Sample Input 2

```
4
2 3 5 7
4 6
3 2
4 1
6 4
8 3
8 2
15 1
```

Sample Output 2

```
9
6
2
5
```

This page is intentionally left blank.

Problem D

Corporate Retreat

Time limit: 7 seconds

Your company is going on a one-day retreat to help your employees build new connections. Three activities will be run simultaneously at the retreat so each employee will participate in exactly one activity.

Some pairs of employees already *work closely* in their day-to-day operations. To maximize the likelihood of employees making new connections, you declare that no two employees that already work closely should be in the same activity. Finally, some of your employees are *executives*. Note, due to company policy every employee that is not an executive works closely with at least one executive.

Your job is to determine if it is possible to assign each employee to an activity so that no two employees that work closely are assigned to the same activity.

Input

The first line of input contains three integers: N ($1 \leq N \leq 1000$), K ($1 \leq K \leq \min(7, N)$), and M ($N - K \leq M \leq 5000$). Here, N is the number of employees, K is the number of executives, and M is the number of pairs of employees that work closely. The employees are numbered from 1 to N , and the executives are employees 1 through K .

Then M lines follow, each containing two integers i, j ($1 \leq i < j \leq N$) meaning employees i and j work closely together. No pair of employees that work closely will be listed more than once and every employee that is not an executive will work closely with at least one executive.

Output

In the first line, output `possible` or `impossible`, indicating whether or not it is possible to assign each employee to an activity while ensuring any two employees that work closely are assigned to different activities.

If it is possible, output a second line containing N integers, each being 1, 2, or 3. The i -th integer on this line indicates the activity that the i -th employee should attend. If there are multiple valid solutions, you may output any.

Sample Input 1

```
5 2 7
1 2
1 3
2 3
2 4
2 5
3 4
4 5
```

Sample Output 1

```
possible
2 1 3 2 3
```

Sample Input 2

```
5 2 8
1 2
1 3
2 3
2 4
2 5
3 4
3 5
4 5
```

Sample Output 2

```
impossible
```

Sample Input 3

```
3 2 2
1 3
2 3
```

Sample Output 3

```
possible
1 1 2
```

Problem E

Draft Lottery

Time limit: 2 seconds

In professional sports, a draft is typically held each year after the end of the season. The draft allows teams to select new players from among the available prospects. The order of selection is often determined using a lottery. A team's odds of getting a draft pick are based on how the team finished during the regular season. The worse a team's record in the regular season, the higher their chance of landing a top pick in the lottery (and strengthening their team before the next season).

Here we describe a novel draft lottery process. Each of the bottom N teams will have their team's logo placed on a certain number of lottery balls (each ball has exactly one team's logo); the i -th worst-ranked team receives b_i ping pong balls in the lottery. If team i had a worse record than team j , then $b_i \geq b_j$.

All ping pong balls are placed in a mixing container and balls are drawn sequentially. The team whose logo is on the first ping pong ball drawn gets the first draft pick. The ball drawn is discarded. A second ball is drawn and the team whose logo is on that ball gets the second draft pick, and so forth. Whenever a ball is drawn for a team that has already received a pick, it is ignored and another ball is drawn in its place.

The first 3 picks are assigned by this lottery drawing. After the top 3 picks have been assigned, the *remaining* teams (those that did not receive a top-3 pick from the drawing) are assigned picks in order from worst record to best.

You've been hired as a consultant by one team's Vice President of Analytics and Strategy to determine the probabilities of each team getting certain picks in order to help the team assess the values of potential trading situations.

Given the counts of ping pong balls that each team will receive in the draft, you must calculate the answers to queries of the form (t, k) , meaning, what is the probability that team t receives one of the first k draft picks?

Input

The first line contains an integer N ($3 \leq N \leq 50$), indicating the number of teams in the draft lottery. The next N lines provide the number of lottery balls that each team is allocated. Specifically, the i -th of these N lines contains two values: an integer b_i ($1 \leq b_i \leq 50$) and a string t_i . The team name t_i consists of one to ten uppercase letters. The number of lottery balls for t_i is indicated by b_i . Team names are unique and teams are listed strictly in *worst-to-best* order; for team t_i , listed immediately before team t_{i+1} , it is guaranteed that $b_i \geq b_{i+1}$.



Image generated by Gemini 2.5 Flash. CC BY-SA 3.0

The next line contains an integer Q ($1 \leq Q \leq 200$), the number of probability queries. Next, Q lines follow, each containing one query. Each query line consists of a team name t_j and an integer k_j ($1 \leq k_j \leq N$). The team name in each query is guaranteed to be present in the earlier allocation list.

Output

Output Q lines. Each line should contain the answer to the respective query (in the same order they appear in the input). If the j -th query is t_j and k_j , then the j -th output line should provide the probability of team t_j receiving one of the first k_j draft picks. Each probability must have a *relative error* of less than 10^{-6} .

Sample Input 1

```
3
4 SUNS
2 NUGGETS
1 JAZZ
9
SUNS 1
SUNS 2
SUNS 3
NUGGETS 1
NUGGETS 2
NUGGETS 3
JAZZ 1
JAZZ 2
JAZZ 3
```

Sample Output 1

```
0.571428571
0.895238095
1.0
0.285714286
0.714285714
1.0
0.142857143
0.39047619
1.0
```

Sample Input 2

```
5
13 MAMMOTH
11 KNIGHTS
7 AVALANCHE
5 FLAMES
3 OILERS
9
MAMMOTH 1
MAMMOTH 2
MAMMOTH 3
MAMMOTH 4
MAMMOTH 5
KNIGHTS 1
AVALANCHE 2
FLAMES 3
OILERS 4
```

Sample Output 2

```
0.333333333
0.613999767
0.825377659
1.0
1.0
0.282051282
0.381096028
0.476970929
0.306898614
```

Problem F

Energy Usage

Time limit: 3 seconds

The price of electricity energy fluctuates depending on supply and demand. As the manager of a 24-hour factory producing widgets, Homer wants to minimize the energy cost to operate the factory. Homer has no ability to control the price, nor can he change the amount of energy required to produce widgets. However, the factory has some solar panels installed which convert sunlight into energy. Homer is considering purchasing a battery of capacity B Megawatt-hour (MWh) so he can store energy for use later. He would like to know if such a purchase makes financial sense.

Using past data, Homer predicts the following information for the next N hours ($1 \leq i \leq N$ below):

- p_i : the price of energy (in dollars/MWh) for the i -th hour
- r_i : the amount of energy (in MWh) required to operate the factory for the i -th hour
- s_i : the amount of energy (in MWh) generated by solar panels for the i -th hour

Furthermore, Homer assumes the battery is initially empty.

For each hour, Homer can decide to do one or more of the following actions to satisfy the energy requirement for that hour:

- use some amount of solar energy generated
- use some amount of battery energy stored
- purchase energy from the power company, paying p_i per MWh purchased

After fulfilling the energy requirements for the hour, Homer can also choose to do one or more of the following actions:

- store some amount of the excess energy (combined from excess solar energy, excess purchased energy, and stored battery energy) into the battery, up to the battery capacity.
- sell some amount of the excess energy. Each hour, the power company has M energy buyback plans. For the j -th buyback plan, Homer can sell exactly exactly $d_{i,j}$ MWh for $c_{i,j}$. Up to one plan can be chosen per hour.
- waste some amount of the excess energy.

Help Homer determine the minimum cost to operate the factory for the next N hours. Any energy left in the battery at the end of the N -th hour is considered wasted. A negative cost means that Homer can make a profit.

Input

The first line contains three integers N ($1 \leq N \leq 1\,000$), M ($1 \leq M \leq 10$), and B ($1 \leq B \leq 20$), where N is the number of hours, M is the number of buyback plans available for each hour, and B is capacity of the battery.

Each of the next N lines contains three integers, p_i , r_i , and s_i ($1 \leq p_i, r_i, s_i \leq 100$), where p_i is the price of energy for the i -th hour, r_i is the energy required to operate the factory for the i -th hour, and s_i is the amount of solar energy generated for the i -th hour.

Each of the next N lines contains M integers, $1 \leq c_{i,j} \leq 10\,000$, where $c_{i,j}$ is the price for the j -th buyback plan for the i -th hour.

Each of the next N lines contains M integers, $1 \leq d_{i,j} \leq 100$, where $d_{i,j}$ is the amount of energy (in MWh) that can be sold for the j -th buyback plan for the i -th hour.

Output

Output the minimum cost required to operate the factory for the next N hours.

Sample Input 1

```
2 1 10
4 3 3
1 5 3
4
9
2
3
```

Sample Output 1

```
-4
```

Sample Input 2

```
3 4 5
1 5 3
9 4 3
10 6 3
1 3 4 9
1 3 6 9
1 3 6 30
1 2 3 4
1 2 3 4
1 2 3 4
```

Sample Output 2

```
1
```



Sample Input 3

```
3 2 2
1 3 6
4 6 3
10 6 3
4 10
2 9
10 20
1 10
2 4
1 3
```

Sample Output 3

```
18
```

This page is intentionally left blank.

Problem G

Genies

Time limit: 2 seconds

Aladdin has just found a magical lamp, and after rubbing it a few times, a genie appears before him. Aladdin told the genie, “I have heard about genies. I can make 3 wishes but cannot wish for more wishes. Is that correct?” The genie responds with, “but there is a loophole for stronger genies like me.” It turns out that there is a hierarchy of genies. A genie of rank K is allowed to grant K wishes, and all K wishes must be used. Each genie can grant at most one wish to conjure another genie who can then grant more wishes.

However, a genie of rank K can only conjure another genie of a lesser rank $K' \leq K - C$. For example, if $C = 2$, then you can ask a genie of rank 5 to conjure another genie of rank 1, 2, or 3. So Aladdin can make a wish of the form “I wish to conjure another genie of rank K' ” for some $K' \leq K - C$. Of course, there are no genies of rank lower than 1.

A “real wish” is any wish that does not conjure a new genie. For example, Aladdin may tell the genie “I wish for all my computer programs to be bug-free.”

Can Aladdin make exactly W real wishes?

Input

The first line contains three positive integers W ($1 \leq W \leq 500$), K ($1 \leq K \leq 200$), and C ($1 \leq C \leq 200$). Here, W is the number of real wishes Aladdin wants to make, K is the rank of the initial genie, and C dictates the rank of the genies Aladdin can wish for.

Output

If Aladdin can make exactly W real wishes, output `yes`. Otherwise, output `no`.

Sample Input 1

3 3 2

Sample Output 1

yes

Sample Input 2

16 12 2

Sample Output 2

yes

Sample Input 3

98 40 10

Sample Output 3

no

This page is intentionally left blank.

Problem H

Hexadecimal Confusion

Time limit: 1 second

Addison Hex is an aspiring athlete training for the next mental Olympics. The event Addison wants to compete in is Fast Hexadecimal Addition, but only the top 16 people will make it to the finals. That means a lot of practice from now on!

In Fast Hexadecimal Addition, contestants are shown very quickly a sequence of hexadecimal numbers (up to 4 digits each) and they must correctly give the sum of these numbers at the end in hexadecimal. No numbers will have leading zeros, and the integer 0 will be represented by exactly one 0.

Addison does not have any sponsorship deals, so she is stuck with an old LCD display that will flash the numbers quickly for her to add. Unfortunately, in the standard 7-segment displays, the digits 8 and B look the same. Also, the digits 0 and D look the same. Thankfully all other digits are unambiguous. Instead of finding the exact sum, she will practice by finding both the maximum possible sum and the minimum possible sum based on what she sees. While Addison may see numbers with leading zeros as she tries to record the numbers as quickly as she can, she also knows that the original numbers have no leading zeros and will take that into account in her calculations.

As a reminder, the digits ‘A’, ‘B’, ..., ‘F’ in hexadecimal have equivalent decimal values of 10, 11, ..., 15, respectively.



Image edited from an original generated by ChatGPT
(©OpenAI, used with permission)

Input

The first line contains a positive integer N ($1 \leq N \leq 1\,000$), which is the number of integers that Addison sees. Each of the next N lines contains a nonnegative integer in hexadecimal, with up to 4 digits. Each digit is one of the decimal digits or an uppercase letter ‘A’, ‘B’, ‘C’, ‘D’, ‘E’, ‘F’. Each number is recorded based on what Addison sees.

Output

On the first line, output the maximum possible sum in hexadecimal. On the second line, output the minimum possible sum in hexadecimal. There should be no leading zeros in the output, and the integer 0 should be represented by exactly one 0.

Sample Input 1

2	C908
ABCD	B8F8
1038	

Sample Output 1
Sample Input 2

10	2ACEE
1111	2ACEE
2222	
3333	
4444	
5555	
6666	
7777	
9999	
1234	
2345	

Sample Output 2
Sample Input 3

10	450B7
1234	3B6EB
8008	
6348	
2008	
ABCD	
EFAB	
2147	
3728	
30DB	
BDBD	

Sample Output 3
Sample Input 4

4	265FE
0012	248E4
D012	
ABCD	
0	

Sample Output 4

Problem I

Poetic Tournament

Time limit: 4 seconds

This is an interactive problem. This means that the input to your program will change based on what your program outputs.

Väinö has been selected as the new chairman for the Iconic Character Poetry Contest (ICPC), a televised battle of wits and verse boasting the nation's best and brightest. Unfortunately, ICPC's popularity has exploded in recent years, and Väinö is beset with hoards of new applicants wishing to test their mettle.

The program has a limited number of slots to accept applicants, and Väinö is tasked with ensuring that only the best applicants get said slots. He has decided to host a qualifying tournament to decide who advances. In the tournament, he will invite two applicants at a time to compete with each other to assess which has the greater skill at poetry, or *runous*. He is certain that no two applicants will have the same skill level.

However, his time is limited and he wants to limit how many matches he must hold in order to determine the top applicants that will advance. Can you help him choose which matches to hold?

Interaction

Interaction begins by reading two integers, N and K , representing the total number of applicants and the number that will advance. You are given that $1 \leq N \leq 15\,000$ and $1 \leq K \leq \min(N, 30)$. The N applicants are numbered from 1 to N .

You may then make at most $12N + 1\,000$ queries of the form `? a b`, where $1 \leq a, b \leq N$ and $a \neq b$. This represents holding a match between applicants a and b , and the interactor will respond with either a or b , representing which applicant won the match and has higher skill.

Remember to flush your output after making a query.

```
cout.flush(); // C++
System.out.flush(); // Java
print(..., flush=True) # Python
```

Your submission will receive a Wrong Answer verdict if you make more than $12N + 1\,000$ queries.

Once you have identified the K applicants with highest skill, print them out in the form `! x1 x2 x3 ... xK`. The top K applicants may be output in any order. Your program should then exit.

The interactor is not adaptive, meaning that the skill level of each applicant is predetermined at the start of the interaction and does not change during the interaction.

Beyond correctly identifying the top K applicants, your queries must also show that your answer is the unique set of K applicants that could be the top K —Väinö's wrath will befall you if there is any possibility of you missing a budding talent! For example, simply outputting `! 5 3 4` for the sample will receive Wrong Answer since your queries have not proven that $\{5, 3, 4\}$ are the top 3 applicants.

If the interactor receives any invalid input, the interactor will output `-1` and then immediately terminate.

Your program should cleanly exit in order to receive a Wrong Answer verdict, otherwise the verdict that it receives may be an arbitrary verdict indicating that your submission is incorrect.

You are provided with a command-line tool for local testing. The tool has comments at the top to explain its use.

Read	Sample Interaction 1	Write
6 3		
	? 1 3	
3		
	? 1 2	
1		
	? 3 5	
5		
	? 3 6	
3		
	? 5 4	
5		
	? 3 4	
4		
	! 5 3 4	

Problem J

Spies vs. Spies

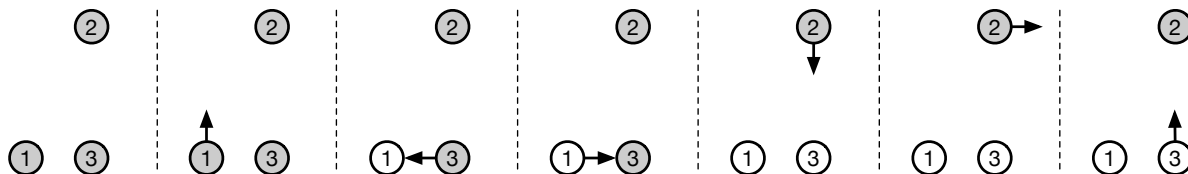
Time limit: 7 seconds

Spies vs. Spies is a large group game. Every team consists of exactly two players: a *spy* and a *handler*. The spies are placed at integer points (x, y) in the playing field which we view as a Cartesian plane.

A turn consists of a team's handler telling their team's spy to look in one of the four cardinal directions: North (N), South (S), East (E), or West (W). Here, North is in the direction of the positive y -axis.

That team's spy will look carefully in that direction from their location. All spies they see in that direction are eliminated. To be precise, when a spy i at location (x, y) looks in one of the four directions, they only see other spies that lie exactly on the straight line originating from (x, y) and drawn in the direction that spy i is looking. For example, if a spy at $(5, 7)$ looks North (N) they only see other spies with coordinates $(5, y)$ where $y > 7$. The spies are eliminated in order from closest to farthest from spy i 's position.

You are reviewing the transcript of instructions that were given over a series of turns. Determine which spies were eliminated in each turn.



A depiction of the initial layout of the three spies and the results of each instruction. A grey circle indicates the spy has not yet been eliminated, a white circle indicates the spy has been on that turn or earlier.

Input

The first line consists of two integers N ($1 \leq N \leq 300\,000$) indicating the number of teams and M ($1 \leq M \leq 300\,000$) indicating the number of turns. Teams are numbered from 1 to N .

Then N lines follow, the i -th such line contains two integers x_i, y_i ($-10^9 \leq x_i, y_i \leq 10^9$) indicating the position of the i -th team's spy. No two spies will be at the same location.

Then M lines follow. The i -th line contains one integer t_i ($1 \leq t_i \leq N$) and a direction N, S, E, or W. This indicates the handler of team t_i instructed their spy to look in that direction.

Output

Output a single line for each instruction. For the i -th line, if the spy for team t_i was eliminated in a previous turn, output `ignore`. Otherwise, output the number of spies that were eliminated followed by the list of the team numbers of all spies that were eliminated, in the order their spies were eliminated.

Sample Input 1

```
3 6
0 0
1 2
1 0
1 N
3 W
1 E
2 S
2 E
3 N
```

Sample Output 1

```
0
1 1
ignore
1 3
0
ignore
```

Sample Input 2

```
5 4
0 0
1 0
2 0
3 0
4 0
5 N
4 E
3 W
4 W
```

Sample Output 2

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

Problem K

Team Practice

Time limit: 6 seconds

Dr. Mitko Dimitrov is planning team practices to prepare his students for the next regional contest. He wants all of his students to get experience practicing together so that they can share knowledge and avoid the situation where the whole team graduates at once and leaves no one for the next year.

To facilitate this practice, he has booked a bifurcated room where students can practice in two separate teams (team A and team B). This allows him to monitor both teams simultaneously and to direct students to switch teams so that they can get experience working with different people. The practice will be held in hour-long sessions, and he can direct a small number of students to switch teams between sessions. However, he wants everyone to be together in team A for both the first and last sessions to explain the practice and to debrief at the end.

Each student can be classified as either a problem solver or a problem coder depending on their strengths (but not both). To minimize foot traffic, Dr. Dimitrov will direct at most one student of each type to switch teams during a break between sessions.

Dr. Dimitrov has also prepared a small number of challenge problems that he plans to present to specific teams at various points throughout the practice. As the problems have various difficulties, he has predetermined the number of problem solvers and problem coders that are needed to solve the problem during a session. There might also be a combined minimum number of students needed to solve the problem that is greater than the sum of the required number of solvers and coders. Dr. Dimitrov wants to structure the teams so that all challenge problems are solved during the practice.

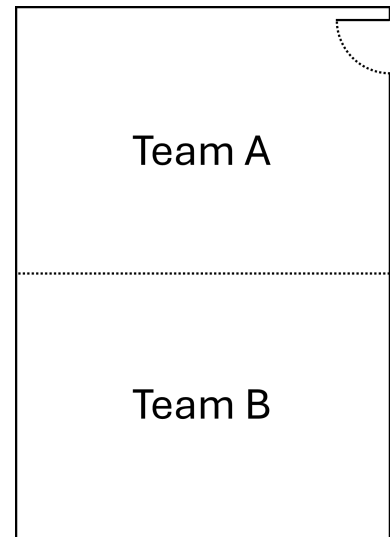
Note that a team may be given multiple challenge problems for the same session, and that they can solve all of them if the team composition meets the requirements for each problem. For example, note that a team with 6 solvers and 5 coders can solve the second and third problems in the first sample input in one session.

How many ways can Dr. Dimitrov direct the students to switch teams to ensure that the constraints are met? Output your answer modulo $10^9 + 7$. Two ways are considered distinct if there is some session in which a specific student is asked to switch teams in one way and not in the other.

Input

The first line of input contains 4 integers— N ($3 \leq N \leq 10^{18}$), the number of sessions in the practice, P ($1 \leq P \leq 5$), the number of challenge problems, S ($1 \leq S \leq 35$), the number of problem solvers, and C ($1 \leq C \leq 35$), the number of problem coders.

Each of the next P lines describes a challenge problem. The line begins with an integer t_i ($1 < t_i < N$)



Team Practice Room

and a character g_i ($g_i \in \{A, B\}$), indicating that the i -th challenge problem is presented to team g_i during the t_i -th session in the practice. Note that Dr. Dimitrov will not hand out challenge problems during the first and last practice sessions. The rest of the line contains three integers describing which students are needed to solve the challenge problem: s_i ($0 \leq s_i \leq S$), c_i ($0 \leq c_i \leq C$), and b_i ($\max(1, s_i + c_i) \leq b_i \leq S + C$), which indicate that the i -th challenge problem requires s_i problem solvers, c_i problem coders, and at least b_i solvers and coders in total.

The challenge problems are listed in non-decreasing order by session.

Output

Output the number of ways, modulo $10^9 + 7$, that Dr. Dimitrov can direct his students to switch teams within the constraints.

Explanation for sample 1

In this test case, there are 7 solvers and 5 coders. Everyone is together in team A during session 1. There is a challenge problem during session 2 for team B that needs one of each type of student. Dr. Dimitrov must pick one of the 7 solvers and one of the 5 coders to send to team B at that time. During session 3, the coder that went to team B must return to team A to help with the first of the two challenge problems there. The solver that went to team B does not need to return immediately, so they could return to team A for either session 3 or session 4. Everyone is together again during session 4.

The answer is $7 \cdot 5 \cdot 2 = 70$ (consider which solver and coder are selected to go to team B and at what time the solver comes back to team A).

Sample Input 1

```
4 3 7 5
2 B 1 1 2
3 A 0 5 6
3 A 6 2 8
```

Sample Output 1

```
70
```

Sample Input 2

```
8 2 5 3
6 A 4 2 7
7 B 2 1 3
```

Sample Output 2

```
0
```

Sample Input 3

```
10 2 3 4
3 A 1 2 4
6 B 2 1 3
```

Sample Output 3

```
289313785
```



Sample Input 4

10000000000000000000 2 4 4
50000000000000000000 A 2 1 4
50000000000000000000 B 1 2 4

Sample Output 4

759384011

This page is intentionally left blank.

Problem L

Video Games

Time limit: 1 second

Alice, Bob, and Charlie share a collection of video game cartridges. This way, only one of them needs to own a game in order for them all to enjoy it. The popular games right now are a fishing game, a golf game, and a hockey game.

Initially, Alice has the fishing game, Bob has the golf game, and Charlie has the hockey game. Over time, the games gets passed around between them and it can be hard to keep track of who currently has each game.

Given a sequence of requests indicating that a particular friend wants to play a particular game, determine who they have to borrow that game from.

Input

The first line of input contains a single integer N ($1 \leq N \leq 100$) indicating the number of requests. Then N lines follow describing the requests, the i -th line contains the text a_i wants to play b_i where a_i is one of alice, bob, or charlie and b_i is one of fishing, golf, or hockey.

After the i -th request, friend a_i is now in possession of game b_i .

Output

For each request i , output a single line containing one of the following messages. If a_i was already in possession of game b_i then output the message “ a_i already has b_i ” (without quotes). If a_i was not already in possession of b_i then output the message “ a_i borrows b_i from c_i ” (again without quotes) where c_i is the name of the friend who was in possession of game b_i at this time.

Sample Input 1

```
4
alice wants to play golf
bob wants to play golf
charlie wants to play hockey
charlie wants to play fishing
```

Sample Output 1

```
alice borrows golf from bob
bob borrows golf from alice
charlie already has hockey
charlie borrows fishing from alice
```

This page is intentionally left blank.

Problem M

What Time Is It Mr. Fox?

Time limit: 8 seconds

The game *What Time Is It Mr. Fox?* is a popular children's game. One child is the fox, the rest are the sheep. The game is on a field that is much longer than it is wide, so we will just assume it is played on a line that is L micrometers long. Position 0 is the left side and position L is the right side.

The sheep are initially located at position 0 on the line and the fox is initially located at position $L/2$. The game proceeds as follows:

- The sheep call out "What time is it Mr. Fox?".
- The fox announces the hour, either X o'clock where $X \in \{1, 2, \dots, 11\}$ or *lunch time!*.
- If the fox announced X o'clock, then all sheep take X steps to the right and the game proceeds with another call.
- But if the fox announced *lunch time!*, the sheep run to the side of the field they are closest to. If they are exactly halfway across the field, they run to the right side.
- The fox chooses at most one of the sheep to chase. If they catch the sheep before the sheep reaches the side it is running to, that sheep is out for the rest of the game.
- All sheep who make it to the right side of the field are *safe* for the rest of the game and no longer play.
- All sheep who make it to the left side of the field are still in the game.
- After all sheep are done running, the fox returns to the center of the field and the game resumes but only with the sheep remaining on the left side.

Each sheep i has a different step size $s_{i,X}$ for each possible value of $X \in \{1, 2, \dots, 11\}$. So when the fox announces it is X o'clock then sheep i moves exactly $X \cdot s_{i,X}$ micrometers toward the right side of the field. If this brings them to the right side, they stop at position L and will be declared safe next time the fox calls *lunch time!*. These step distances might not be increasing for a sheep, for example a sheep may take two large steps at 2 o'clock but three tiny steps at 3 o'clock.

All sheep run at the same speed and the fox runs at exactly twice the speed of the sheep. So if the fox chooses to chase a sheep whose distance is D micrometers to the side they are running to, the fox will catch the sheep if $L < 4D$ and otherwise the sheep will reach the side without being caught.

Every *lunch time!*, the fox uses the following to decide which sheep to chase:

- If the fox can catch a sheep running to the right side, the fox will choose the one with the smallest index i .
- Otherwise, if the fox can catch a sheep running to the left side, the fox will choose the one with the smallest index i .

- Otherwise, the fox chooses to not chase any sheep so all sheep will successfully reach their side.

You will be given the sequence of calls made by the fox during the game. Your job is to determine which sheep reach the right side and which sheep is caught after each *lunch time!* call.

Input

The first line of input contains three integers N ($1 \leq N \leq 200\,000$), M ($1 \leq M \leq 200\,000$), and L ($2 \leq L \leq 10^8$) indicating the number of sheep, the number of instructions, and the length of the field. The sheep are numbered 1 to N , inclusive.

Each of the next N lines contains exactly 11 integers. The X -th value on the i -th line is $s_{i,X}$ ($1 \leq s_{i,X} \leq 10^8$), i.e. the step size sheep i takes forward whenever the fox calls X o'clock.

Finally, M lines follow where each line contains an integer from 1 to 12. The j -th such number indicates what the fox calls: either X o'clock if the number X is from 1 to 11 or *lunch time!* if $X = 12$.

You are also guaranteed that L is an even number, that there are at most 20 *lunch time!* calls, and that the last call is *lunch time!*

Output

For each *lunch time!* call, output two lines. The first line should contain a list of the numbers of the sheep that reached the right side after this call. These numbers should be given in ascending order. If no sheep reached the right side this call, output `None` instead. The second line should contain the number of the sheep that the fox caught this call. If the fox did not catch a sheep this call, output `None` instead.

Sample Input 1

```
2 6 16
1 1 1 1 1 1 1 1 1 1 1
1 2 1 2 1 2 1 2 1 2 1
4
1
12
6
6
12
```

Sample Output 1

```
None
2
1
None
```

**Sample Input 2**

```
2 4 10
1 1 1 1 1 1 1 1 1 1
1 2 1 2 1 2 1 2 1 2 1
2
4
10
12
```

Sample Output 2

```
1 2
None
```

Sample Input 3

```
3 5 20
1 1 1 1 1 1 1 1 1 1
1 2 1 2 1 2 1 2 1 2 1
3 3 3 3 3 3 3 3 3 3 3
10
12
10
10
12
```

Sample Output 3

```
2 3
1
None
None
```

This page is intentionally left blank.