FAU Winter Contest 2015

January 24th

acm International Collegiate Programming Contest

FRIEDRICH-ALEXANDER UNIVERSITÄT ERLANGEN-NÜRNBERG

Problems

A Beacon
B Chronovisor *(easy)*
C Derivatives
D Hiking Trips *(easy)*
E Hollow earth
F Hurry!
G Jump3D
H Pyramids *(very easy)*
I Raffle
J Replace
K Same expression

This page is intentionally left (almost) blank.

Problem A: Beacon
Time limit: 8 seconds

To make navigation easier on the Sea of Tranquility, some radio navigation beacons have been installed at fixed positions. Each beacon transmits a radio signal on a specific frequency, and its signal can be received by ships anywhere on the sea.
You are the navigator of HMS Apollo and your task is to devise a route between two points. For the sake of safety, the captain insisted that the ship should always stay within a triangle defined by beacons operating at the same frequency and the ship's radio should be tuned to that frequency. You are allowed to change the frequency of the radio, but that will take some time during which the ship must be stopped.
Stopping and starting a ship is a complicated and time-consuming procedure, so your route should minimize the number of such manoeuvres.
Note that if more than three beacons operate at the same frequency then the ship can be in a triangle defined by any three of them, and that the edges and vertices of a triangle count as part of the triangle.

Input

The first line contains the x, y coordinates of two points s and t, in metres.
The second line contains a single integer k, $1 \leq k \leq 50$, the number of different frequencies.
Next, each frequency band is described as follows. First a line containing a single integer n_i, $3 \leq n_i$, the number of beacons that operate at that frequency. Then n_i lines follow with the x, y coordinates of each beacon.
Coordinates are integers with absolute value at most 10^6. There are at most $100\,000$ beacons in total. All beacons are in different locations. Not all beacons of the same frequency lie on the same line.

Output

The minimum number of stops needed to go from point s to point t, or the word "`impossible`" if there is no route satisfying the requirements.

Sample Input 1

```
0 -50 150 -50
2
4
100 0
0 100
-100 0
0 -100
4
250 0
150 100
50 0
150 -100
```

Sample Output 1

```
1
```

Sample Input 2

```
0 -50 250 -50
2
4
100 0
0 100
-100 0
0 -100
4
350 0
250 100
150 0
250 -100
```

Sample Output 2

```
impossible
```

Problem B: Chronovisor
Time limit: 1 second

Photos by Leo Reynolds, cc by-sa

With the help of the chronovisor (invented in 1972 by the priest François Brune), one is able to view past events by decoding the electromagnetic radiation from those events. Almost nobody believed that the chronovisor works, even after François Brune managed to take a photograph of Jesus Christ's crucifixion.

The chronovisor was forgotten for a while, but now a group of scientists tries to check if the time viewer may have worked, or even still works. They found a bunch of photographs that were taken with the help of the chronovisor. Each of these photographs lists the date of the shown event and something that looks like a picture of a clock. However the clock hands are of equal length, so we cannot distinguish the small from the large hand. Also the clock may be rotated. We measure the angle between the hands; your job is then to convert the angle to a human readable time in `HH:MM` format. You may assume that the clock hands move continuously and that the photograph was taken on a full minute.

Input

The input is given as one integer value A, the angle between the clock hands in degrees ($0 \leq A \leq 180$).

Output

Print the time of the event in 24 hour format "`HH:MM`". If there is more than one possible time, any will be accepted. You may safely assume that there is at least one such time.

Sample Input 1	Sample Output 1
`76`	`04:08`

Sample Input 2	Sample Output 2
`2`	`15:16`

Sample Input 3	Sample Output 3
`99`	`23:42`

This page is intentionally left (almost) blank.

Problem C: Derivatives
Time limit: 2 seconds

Jeff recently joined an algebra working group that investigates algebraic rings of integers modulo N. An integer in this group is characterized by its remainder, which is left over after division by N.
Jeff has a special problem, where he needs to evaluate the derivatives of polynomials at the value 0 in the previously described ring. Derivatives are defined in the standard way, i.e., the first derivative of the term x^n is $n \cdot x^{n-1}$, the second derivative is $n(n-1) \cdot x^{n-2}$ and so on. A special property of the algebraic structure leads to the fact that the exponents of the terms are relatively large with respect to N. Each exponent is a non-negative integer greater or equal to $N - 1000$. Jeff is not very good in programming and therefore he asks you to write a program to automatize his calculations. Please determine all non zero values of the derivatives of his polynomial for the input value 0.

Input

The first line contains two integers N and M where N determines the size of the ring ($1 \leq N \leq 2 \cdot 10^9$) and M represents the number of terms of Jeff's polynomial ($1 \leq M \leq 1000$). The following M lines contain two integers m_i and n_i ($0 < m_i < N$, $\max(0, N - 1000) \leq n_i \leq 2 \cdot 10^9$). Each pair m_i and n_i describes a term of Jeff's polynomial. The i-th term is defined by $m_i \cdot x^{n_i}$. The exponents n_i are given in strictly increasing order.

Output

Print several lines of output. The first line should contain an integer K, the number of non zero values if the derivatives of Jeff's polynomial are evaluated at 0. Each of the following K lines should contain two integers d_i and f_i ($0 < f_i < N$). Each pair d_i and f_i should represent a non zero value of the derivative of Jeff's polynomial, i.e., the value of the d_i-th derivative evaluated with zero is f_i. The values d_i should be printed in strictly increasing order.

Explanation of the first sample case

Jeff's polynomial is $f(x) = 7 \cdot x^0 + 5 \cdot x^3$.
The 0-th derivative is equal to f and $f(0) = 7$. The first derivative $(15 \cdot x^2)$ and the second derivative $(30 \cdot x^1)$ evaluate to 0. The third derivative is equal to $30 \cdot x^0$ and its evaluation at 0 yields $30 \equiv 8 \mod 11$. The 4-th and all further derivatives are equal to zero and are therefore always evaluated to 0.

Sample Input 1

```
11 2
7 0
5 3
```

Sample Output 1

```
2
0 7
3 8
```

Sample Input 2

```
1999999973 4
3 1999999960
2 1999999965
1 1999999970
5 1999999975
```

Sample Output 2

```
3
1999999960 1272169057
1999999965 1367460299
1999999970 999999986
```

This page is intentionally left (almost) blank.

Problem D: Hiking Trips
Time limit: 3 seconds

Marty and Erica go hiking every weekend. Each of them has a knapsack. As experienced hikers they share items like a map to reduce weight. Furthermore, they distribute all items over the two knapsacks in a greedy manner: The items are in an initial order, which will not be changed. For each item they weigh the two knapsacks and put the item in the lighter knapsack. Obviously, this approach will not necessarily result in an optimal distribution of the items, i.e. where the knapsack weights are as equal as possible. Marty is a real gentleman and therefore always picks the heavier knapsack.

Someday Marty thinks about the process and reckons that sorting the items from heaviest to lightest might result in a smaller difference in weight between the two knapsacks. Erica is not delighted by Marty's idea and plans to convince him to go back to the greedy approach. Her plan is to present Marty a series of item weights a_i in an order far from descending, where the greedy strategy leads to equal weights.

Erica is not very successful in generating those sequences, so she needs your help. Create a sequence consisting of unique positive integers such that the greedy approach leads to equal knapsack weights. Erica wants the *smallest* sequence, because she thinks this series is farthest from descending order. A series $a_0, a_1, \ldots, a_N$ is considered smaller than a series $b_0, b_1, \ldots, b_N$, if there is an integer p such that $a_i = b_i$ for all $i < p$ and $a_p < b_p$.

Input

The first line contains two integers N and Q. N represents the number of items in the sequence you have to generate ($3 \leq N \leq 2^{60}$) and Q the number of positions Erica wants to know ($1 \leq Q \leq 100\,000$). The following Q lines contain a single integer i each ($0 \leq i < N$), the 0-based index of the position Erica wants to know. No two items may have the same weight and all weights have to be positive integers.

Output

Print Q lines of output. Each line should contain the weight of the item at the requested position.

Sample Input 1

```
7 2
0
5
```

Sample Output 1

```
1
9
```

Sample Input 2

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

Sample Output 2

```
1
2
4
3
4
```

Sample Input 3

```
987654321123456789 1
987654321123
```

Sample Output 3

```
987654321124
```

This page is intentionally left (almost) blank.

Problem E: Hollow earth
Time limit: 2 seconds

Some people already suspected it, and it's true: we are living on the outside of a hollow earth! Some even claim that we're living on the inside, but who would listen to such absurd conspiracy theories… In fact, the inside of the hollow earth is inhabited by people like you and me — they call themselves the "insiders". Lately, more and more people have become aware of the inner world, and consequently, the insiders started to fear that their part of the world could be invaded by greedy people from the outside who only want to exploit their abundant resources of He^3.

Thus, the insiders decided to start a gigantic media campaign on the outside by publishing articles depicting people who believe in the inner world as weirdos. They have already created a detailed plan where to publish specific articles by bribing some newspapers.
Shortly before starting their campaign, they noticed an important fact: Newspapers are organized in a hierarchy, where each newspaper has at most one superior newspaper (of course, there are no cycles). Also, each newspaper publishes two types of news: Local and supra-regional news. To save time and money, all newspapers copy all of the supra-regional news of their superior newspaper. They may still add new supra-regional news to their newspaper, though. Local news are only published in one specific newspaper and never copied to others.
Due to this hierarchy, the insiders are able to publish an article in an entire set of newspapers by bribing just one newspaper! While it might be cheaper to change the original plan and publish articles in superior newspapers, the insiders worried that this might draw too much attention. So they decided to stick to their original plan.
Given a newspaper hierarchy and a set of articles with the newspaper they have to be published in, find out the minimum number of newspapers to bribe.

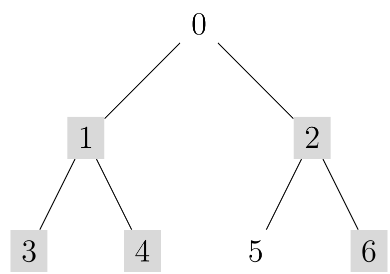


Figure E.1: Illustration of the first sample input (first article). The insiders want to publish the article in the newspapers $1, 2, 3, 4, 6$. They can do so by bribing the newspaper 1 to publish the article in the supra-regional section and by bribing newspapers 2 and 6 to publish the article in the local section. Thus, they have to bribe 3 newspapers.

Input

The first line of the input contains an integer N ($1 \leq N \leq 2\,000$), the number of newspapers. The next line contains N integers. The ith integer $S_i (0 \leq S_i < N)$ is the ID of the newspaper the ith newspaper copies all supra-regional articles from, or -1 if there is no such newspaper. Then follows a single line containing A ($1 \leq A \leq 300$), the number of articles. For each article, a line follows giving L ($0 \leq L \leq N$), the number of newspapers this article is to be published in, followed by L numbers, the IDs of the newspapers P_i ($0 \leq P_i < N$).

Output

One line per article, containing the minimum number of newspapers to bribe.

Sample Input 1

```
7
-1 0 0 1 1 2 2
2
5 1 2 3 4 6
7 6 5 4 3 2 1 0
```

Sample Output 1

```
3
1
```

Problem F: Hurry!

Time limit: 4 seconds

I am sure you already heard of these Nazi UFOs, the so-called *Reichsflugscheibe*. Politicians try to deny the existence of these flying objects, but I am sure they still exist. Probably the Reichsflugscheiben are even responsible for the crop circles!
Recently, on the International Conspiracy Proof Congress I heard some rumors: their construction plans were taken to the Louvre in Paris and hidden behind the famous Mona Lisa drawing. Tomorrow, we will break into the Louvre and steal these documents to present proof to the world that the Reichsflugscheibe exist. We have planned everything: how to get into the museum, when the shift changeover of the guards will take place, how to outsmart the laser detectors and other technical security staff. But there is one problem left: we have to get out of Paris to a small airport as fast as possible.
That is where we need your help: you get a map with all relevant junctions and streets of Paris including the Louvre and the airport. Every street on the map has length 1. Furthermore, we have marked every police station for you. To reduce the risk of being caught by the police, our way to the airport should be the route that stays farthest away from any police station. If there is more than one such route, take the shortest. But in the worst case, we even use junctions with police stations. And now: hurry!

Input

The first line of the input contains two integers N and M ($2 \leq N \leq 300\,000$, $1 \leq M \leq 300\,000$), where N is the number of junctions and M is the number of streets. Then follow M lines, each giving one street as a pair of junctions a_i b_i ($0 \leq a_i, b_i < N$, $a \neq b$). We are in a hurry, so we ignore one way street signs if there are any. You may assume that the underlying graph is connected.
The next line contains an integer P ($0 \leq P \leq N$), which specifies the number of police stations on the map. The next P lines specify the junctions with police stations p_i ($0 \leq p_i < N$). The Louvre is at junction 0 and the airport is at junction at $N - 1$.

Output

Print the length of the shortest route from Louvre to the airport. Keep in mind to stay away as far as possible from police stations.

Sample Input 1

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

Sample Output 1

```
3
```

Sample Input 2

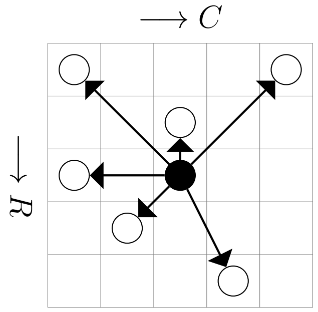
```
9 11
0 4
4 2
1 2
2 3
3 4
3 5
8 5
6 8
2 5
6 7
0 6
2
7
1
```

Sample Output 2

```
4
```

Problem G: Jump3D
Time limit: 2 seconds

Jump3D is a 3D board game consisting of a stack of boards with L levels. Each level is a board with $R \times C$ squares in a grid with R rows and C columns. Every player starts from the same square on one of the boards. The goal is to reach a specific target square as quickly as possible. However, the assigned target square is different for each player.
In each step, one of the following moves is possible:



Furthermore, it is possible to jump up *one* level or jump down *two* levels (increasing, resp. decreasing the level number) in one step. Jumping up and down does neither change row nor column. Finally, B squares are blocked, so it's not possible to move or jump on these. It is also not allowed to move/jump outside of a board.
Obviously, a random board configuration is unfair, as it may prefer some players. A configuration is considered fair if every player can reach its target square and the minimal number of steps from start to target is equal for every player. It is considered better when the minimal distance from start to targets is longer. What's the start square in the best fair configuration given the (already selected) target squares of all players?

Input

The first line of the input contains the integers L, R, and C. These describe the number of levels, rows, and columns ($0 < L * R * C \leq 50\,000$). The next line contains the integer P, the number of players ($0 < P \leq 10$). Then follow P lines, each containing three integers l r c, specifying the level, row and column of a player's target square ($0 \leq l < L$; $0 \leq b < B$; $0 \leq r < R$). The next line contains the integer B, the number of blocked squares ($0 \leq B \leq L * R * C - P$). Then follow B lines, each containing three integers l r c, specifying the level, row and column of a blocked square. You may safely assume that no square is blocked twice and no target square is blocked.

Output

If there is no fair board configuration, print "`UNFAIR`". Otherwise print two lines: the first specifying the start square as three integers (the level, row, and column) in the best board configuration. If there is more than one, print any. On the second line, print two integers: the minimal distance D from start to targets squares for this best board configuration and the number of possible start squares with distance $= D$.

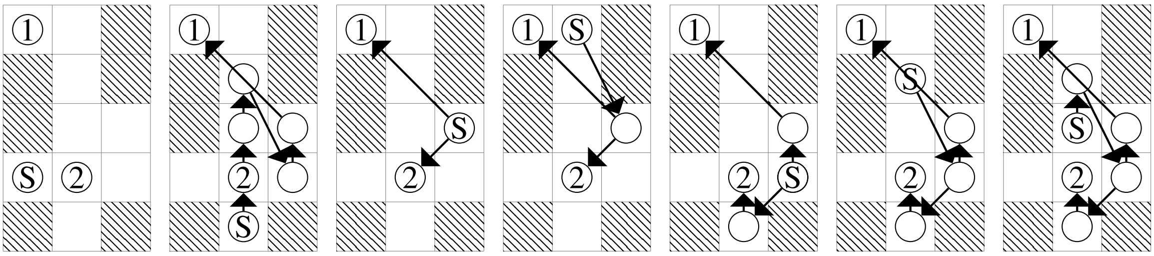


Figure G.1: Illustration of first sample input: shows the shortest move sequences from each possible start square (S) to the target squares (1 resp. 2).

Sample Input 1

```
1 5 3
2
0 0 0
0 3 1
6
0 1 0
0 2 0
0 4 0
0 0 2
0 1 2
0 4 2
```

Sample Output 1

```
0 2 1
4 1
```

Sample Input 2

```
1 5 3
2
0 0 0
0 3 1
8
0 1 0
0 2 0
0 4 0
0 0 2
0 1 2
0 4 2
0 2 2
0 3 2
```

Sample Output 2

```
UNFAIR
```

Sample Input 3

```
30 30 30
2
0 29 29
23 0 0
2
1 29 29
0 27 28
```

Sample Output 3

```
0 12 0
29 2
```

Problem H: Pyramids

Time limit: 1 second

The Illuminati have decided to finally leave the years of secrecy behind. They want to demonstrate their might by erecting pyramids at various points of interests all over the world. Local groups take care of the planning and the construction of these tokens of power, hence they have some liberties in their design. Naturally, there is but one global constraint for the pyramids: They must have a certain size to qualify as an expression of the enlightened group's might. Hence, all pyramid plans must be verified by Bob the Builder, the group's newly appointed lead pyramid construction manager.

For each pyramid, Bob has to review the side length a of the square base area and the height h (both in meters), as these values determine the volume of the pyramid. All pyramids with a volume of less than $23\,000m^3$ are deemed unworthy and must be rejected. As the number of planned pyramids grows into the hundreds, Bob asked you for help to automatically review the plans.

Input

One line with the integer numbers a and h with $42 \leq a, h \leq 1\,000$.

Output

Print one line of output containing "`worthy`" if the volume of the specified pyramid is at least $23\,000m^3$. Print "`rejected`" if the pyramid is too small and gets rejected.

Sample Input 1	Sample Output 1
`1000 1000`	`worthy`

This page is intentionally left (almost) blank.

Problem I: Raffle
Time limit: 1 second

You want to take part in an online raffle for an awesome prize. The drawing procedure is as follows: a limited number of persons, n, is allowed to register. After this number of people has registered, one of them is chosen as the winner. The organizers try to be fair and got a real random number generator. This generator generates a perfectly uniformly distributed integer w. But the range ($0 \leq w < r$) differs from the number of participants. As a workaround, the organizers decided that the i^{th} registrant (0-indexed) wins if $i \cdot r/n \leq w < (i+1) \cdot r/n$. As the random number is an integer and numbers are always rounded down, some participants have better chances to win than others.
The registration website also shows a counter c of already registered people. The next registrant will be the cth one (0-indexed). You want to observe this counter and register in a moment that optimizes your chances of winning. But you have also to consider latency. In half of the tries, you will get the participant number c, in the other half the number $c+1$. If you try to get the last number, you may even be too late and miss the chance to participate in the raffle.

Input

Two numbers are given per testcase. The number of participants n and the range of the random number generator r ($0 < n, r \leq 10\,000$).

Output

Output at which count you should try to register. If two counts lead to the same result, output the earlier one.

Sample Input 1	Sample Output 1
`100 100`	`0`

Sample Input 2	Sample Output 2
`100 99`	`1`

Sample Input 3	Sample Output 3
`100 101`	`98`

This page is intentionally left (almost) blank.

Problem J: Replace

Time limit: 8 seconds

You are working for a gaming company. The game designers decided that technical descriptions for loading bars like “Extracting gfx files” do not help the gamer and are especially no fun. So they decided to replace them by funny stuff like “Feeding the minions” and use them randomly for different actions. But now the gamers get confused that “Digging the dungeon” takes longer than “Polishing the armor”. So the plan is to reconsider the times that real and the fake action take. The relative time between actions do not need to be correct, but the overall order should be correct. I.e. a real action which takes more time, should be described by a fake action which takes more time. And when two real actions take the same time, the according fake actions should take the same time. Before implementing this the designers want to make sure that the variety for the gamers is big enough still to be fun. So given times of the different loading actions and the times of the fake actions, you should calculate the number of possible different assignments of real actions to fake actions. Each assignment must assign a different fake action to every real action.

Input

The input starts with a line containing two integers. The number of real actions a and the number of fake actions f. Followed by a line of a integers, the times of the real actions, in the order they are executed. The last line contains f numbers: the time the fake actions take. All numbers are positive and smaller than $5\,000$.

Output

Output the number of possible assignments modulo $2\,147\,483\,647$.

Sample Input 1

```
3 4
1 2 3
10 20 30 40
```

Sample Output 1

```
4
```

Sample Input 2

```
4 5
4 7 5 4
2 1 2 7 8
```

Sample Output 2

```
2
```

Sample Input 3

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

Sample Output 3

```
0
```

This page is intentionally left (almost) blank.

Problem K: Same expression
Time limit: 1 second

Jakob is teaching compiler construction again. This time, he has an exercise for his students, where they have to simplify expressions with constants. Sadly, the simplification code his students have written is so complicated that he doesn't understand it. Instead, he resorts to testing it. He creates several expressions and passes them through the student's simplification algorithm. Now he just has to check whether the resulting expression and the original expression are equivalent.
Expressions consist of integers, variables, additions and multiplications, nested in an arbitrary depth with braces.
Grammar:

```
EXPR ::= FACTOR | FACTOR '+' EXPR
FACTOR ::= ATOM | ATOM '*' FACTOR
ATOM ::= INT | 'x' | '(' EXPR ')'
```

An "`INT`" will match the regular expression `'[1-9][0-9]*'`, so there will be only positive integer literals. To further simplify things, there will be only one variable (denoted "`x`"), which will appear at most 12 times in the whole expression.

Input

Input contains two lines, with the original expression in the first line, and the modified version in the second line. Every integer in the expression will be smaller than 2^{10}, and the expression will be no longer than 127 characters. To make your task simpler, you may assume that there is no integer overflow in either one of the expressions, that is, for every x in the range $\{1, \ldots, 31\}$, the expression evaluates to a result less than 2^{40}.

Output

Print one line, containing either `EQUIVALENT` if the two expressions describe the same mathematical function or `NOT EQUIVALENT` if they don't.

Sample Input 1

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

Sample Output 1

```
NOT EQUIVALENT
```

Sample Input 2

```
1+x+3
4+x
```

Sample Output 2

```
EQUIVALENT
```

Sample Input 3

```
(x+2)*(x+2)*(2+x)
x*x*x+3*x*x*2+3*x*2*2+2*2*2
```

Sample Output 3

```
EQUIVALENT
```

This page is intentionally left (almost) blank.