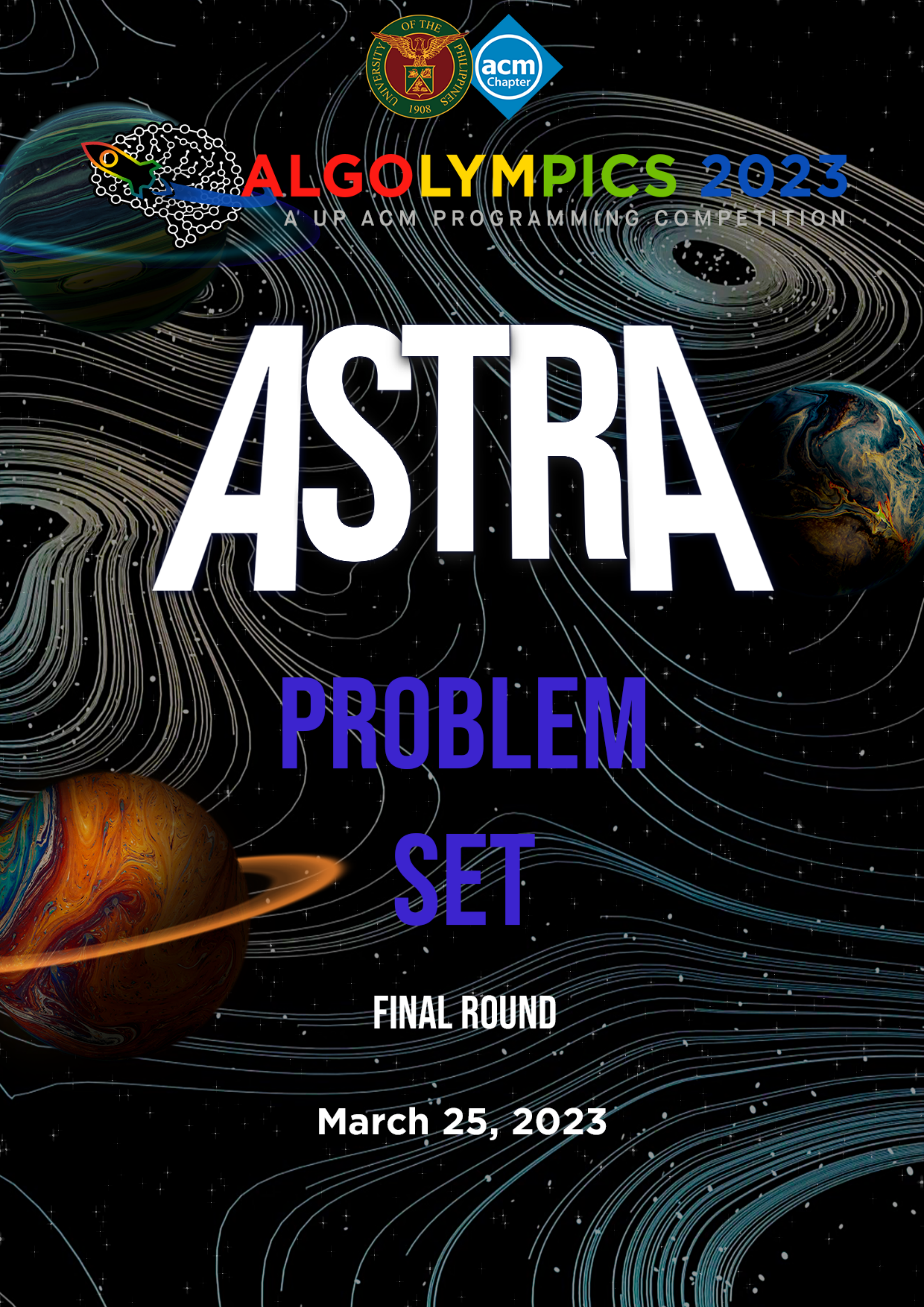


ALGOLYMPICS 2023

A UP ACM PROGRAMMING COMPETITION

ASTRA

PROBLEM SET

FINAL ROUND

March 25, 2023

Contents

Notes

- Many problems have large input file sizes, so use fast I/O. For example:
  - In Java, use `BufferedReader` and `PrintWriter`.
  - In C/C++, use `scanf` and `printf`.
- All problems are solvable in C++, and Java, but the same is not guaranteed for Python due to its slowness.
- Good luck and enjoy the problems!

Very important! Read the following:

- Your solution will be checked by running it against several hidden test cases. You will not have access to these cases, but a correct solution is expected to handle them correctly.
- The output checker is usually strict, so follow these guidelines strictly:
  - It is *space sensitive*. Do not output extra leading or trailing spaces. Do not output extra blank lines unless explicitly stated.
  - It is *case sensitive*. So, for example, if the problem asks for the output in lowercase, follow it.
  - Do not print any tabs. (No tabs will be required in the output.)
  - Do not output anything else aside from what's asked for in the Output section. So, do not print things like "Please enter t".
  - Generally, each output line should be ended by a newline (`'\n'`) character, even if there are no succeeding output lines. For example, in C/C++ this could look like `printf("The only line\n")`. Note that for Python, using `print("The only line")` by default adds the newline at the end.

  Not following the output format strictly and exactly may result in a *Wrong Answer* verdict.
- Do not read from, or write to, a file. You must read from the standard input and write to the standard output.
- For Java, do not add a package line at the top of your code. Otherwise, your submission will be judged *Runtime Error*.
- Only include one file when submitting: the source code (.cpp, .java, .py, etc.) and nothing else.
- Only use letters, digits and underscores in your filename. Do not use spaces, or other special symbols.
- Many problems have large input file sizes, so use fast I/O. For example:
  - In Java, use `BufferedReader` and `PrintWriter`.
  - In C/C++, use `scanf` and `printf`.

  We recommend learning and using these functions during the Elimination Round and/or the Practice Session.
- All problems are solvable in C++ and Java, but the same is not guaranteed for Python due to its slowness.
- Good luck and enjoy the contest!

Problem A
Alien Gordon Ramsey
Time Limit: 1.5 seconds

Consider the following problem posed by Alien Gordon Ramsey:

> I have n restaurants in Alien Las Vegans, labeled 1 to n. Alien Las Vegans has $n-1$ roads, each one connecting two different restaurants, and the roads are set up such that it is possible to travel from any restaurant to any other restaurant, just by taking these roads.
>
> Now, I want each restaurant to have a **theme** (e.g. Alien Vegan Burgers, Alien Vegan Steakhouse, Alien Vegan Hell's Kitchen, etc.). I could give each restaurant a different theme, but that's a lot of work that I don't actually want to do. Instead, I want to see if I can get away with recycling themes *within Alien Las Vegans* and seeing if no one will notice.
>
> - No two restaurants directly connected by a road must have the same theme.
> - No two different *roads* exist such that the two restaurants at their endpoints have the same pair of themes.
>
> I need you to compute the *minimum* number of different restaurant themes that I need to develop in order to satisfy these conditions. And not just that—I need you to assign a theme to each restaurant such that these conditions hold. Do this, and I will grant you the title of Alien Master Chef.

Unfortunately, Alien Pal Erdős is here to ruin the fun. "I really don't think this problem is solvable with the current techniques of modern mathematics," he says.

Alien Gordon Ramsey answers, "Well [illegible] you, you [illegible]. Alright fine, what if I add the following condition: **It is possible to get from any restaurant to any other restaurant by taking B roads or fewer.**"

Alien Pal Erdős says: "Suppose the humans threaten to obliterate us if we can't produce a program that solves this problem for $B = 5$ within a year. Then we should marshal together the universe's smartest computer scientists, and we could probably pull it off. If the humans ask us to solve the problem for $B = 6$, then I propose that we just kill all the humans."

Alien Gordon Ramsey thought for a moment, and then finally responded, "I can't wait a year. If $B = 4$, then do you think a team of three or so undergrad-

Old.St LABS
This problem is sponsored by Old.St Labs

uates could solve the problem within five hours?"

Input

The first line of input contains a single integer T, the number of test cases.

Each test case begins with a line containing a single integer n.

Then, $n-1$ lines follow, each containing two space-separated integers u and v, meaning that a road exists directly connecting restaurants u and v.

Output

For each test case, output two lines.

First, output a line containing a single integer c, the number of different restaurant themes.

Let's represent these c themes by the positive integers $1, 2, 3, \ldots, c$. Next, output a line of n space-separated integers, each from 1 to c, such that the ith of these integers corresponds to the theme assigned to restaurant i.

Your answer will be considered correct if c is minimal, and the provided assignment of themes is valid. If there are multiple possible answers, any will be accepted.

Constraints

- $1 \leq T \leq 10^4$
- $1 \leq n \leq 3 \times 10^5$
- The sum of n across all test cases is $\leq 3.5 \times 10^5$.
- For each road, $u \neq v$ and there exists at most one road connecting any two restaurants.
- It is possible to travel from any restaurant to any other restaurant using 4 **roads or fewer**.

Sample Input

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

Sample Output

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

Problem B
Cult of Wah!
Time Limit: 2 seconds

By the order of the emperor, Blanc Enyelu, the Star Seeker, has infiltrated the notorious Order of the Crooked Tentacle, known as the Tentacult, within the imperial court! Her goal was to ascertain whether they are involved with NELOC. While infiltrating the cult, she was able to retrieve a document from their headquarters in Star System R136a1, in large part thanks to his footsteps being drowned out by the cultists repeated chants of "Wah!"

Sadly however, the document contains a message s that is encrypted. Luckily, one of Blanc's associates is an expert in cryptography, and she determines that the cult most likely uses *Caesar's cipher* for their messages. This cipher works by substituting each letter in the message with the letter k steps ahead of it in the alphabet. So for example, with $k = 3$, a becomes d, b becomes e, and so on. In case the shift takes you beyond the letter z, it will wrap around back to a. So, with $k = 3$, x becomes a, y becomes b and z becomes c. Anything that isn't a letter—such as a space—is ignored and kept as is.

Blanc's associate also tells her that the Tentacult only ever uses **positive integer values of** k for encrypting and decrypting their cipher, in order to minimize confusion with its non-programming members.

In order to decode this message, Blanc must identify what value of k was used by the cult. Luckily, she has a list of words which she is *sure* will show up in the decoded message. This list may or may not include the cultists creepy high-pitched chanting of "Wah!", so Blanc calls it the **Wah-List**.

She must use this information to identify the smallest possible positive value of k such that the decoded message contains **all** the words from this list. Of course, if no such value of k can be found, then Blanc needs to know so as well, so she can complain about the faulty information she was given.

Input

The input begins with an integer T denoting the number of test cases. T test cases follow.

Each test case is described by four lines.

1. The first line contains a single integer, the number of words in the Wah-List.
2. The second line contains the words in the Wah-List, separated by single spaces.

3. The third line contains a single integer, the number of words in the encrypted message.
4. The fourth line contains the encrypted message proper, with each of its words separated by single spaces.

Output

For each test case, output a line containing the smallest positive integer value of k such that all known words are present in the decoded message. In the case that no such k can be found, output -1 instead.

Constraints

- $1 \leq T \leq 10$
- The Wah-List consists of at least 1 and at most 20 words, where each of its words consists of at least 1 and at most 50 lowercase English letters.
- The total length of the encrypted message (including spaces) is at least 1 and at most 1500. Each word in the encrypted message consists only of lowercase English letters.
- The encrypted message doesn't begin or end with a space, and neither does it contain two consecutive spaces anywhere in the text.

Sample Input

```
2
2
wah umu
3
bhv zdk xpx
1
aaa
3
bbb ccc ddd
```

Sample Output

```
3
1
```

Explanation

For the first test case, if the message were encrypted using $k = 3$, then decoding it would reveal it to be `yes wah umu` which contains all the words in the Wah-List.

As for the second test case, the decoded message would contain the word `aaa` (the only word in the Wah-List) if it were encrypted with $k = 1$ or $k = 2$ or $k = 3$. Since we only want the smallest possible value of k, the answer is 1.

Problem C
Dethrone Antares Now
Time Limit: 2 seconds

`Nice Dan: Hello, General Ptolemy. I'm here with the rebellion. We've managed to free your family and secure them to a safehouse. I know it was hard to keep your cover, but you don't need to follow any of this cruel dictatorship's rules any more. You can do anything you like and say whatever you want and those tyrants won't be able to punish you.`

`I'm sorry, but just so we're sure that you're no longer working for them, you must help me with a task that we know that Antares Intergalactica wouldn't approve of.`

The star system is composed of n planets, labeled 1 to n. There also exist m **teleporter-pairs** that allow for free two-way transport between the two planets they connect. Due to the patrol ships of Antares Intergalactica (AI), these teleporters are the only way to move between planets.

Our forces have a commander on k different planets. Our goal is to get all our commanders to congregate at the same planet *at the same time*. At the start of each day, our commanders **must** use a teleporter to transport themselves to another planet. This is not optional; if they linger for too long on one planet, they are caught. After this, they will be unable to use another teleporter until the start of the next day. It is allowed for multiple commanders to be on the same planet together, even before the grand meeting.

We will consider your task successful if you can devise a plan that ends with all commanders on the same planet (*any planet*) at the same time, at day's end. We only ask that you make this plan as efficient as possible (i.e., if it takes d teleporter-uses, then d should be minimized).

`General Ptolemy, the Awesome: Thank you for freeing me! Okay, this sounds important, and I think it's doable. I'll tell you if it's impossible, too. The first step of the solution is to`

Input

The first line of input contains three space-separated integers n, m, and k.

Then, m lines of input follow, each describing a teleporter-pair. Each line contains two space-separated integers u and v, meaning that this teleporter-pair connects those two planets.

Finally, this is followed by a line containing k space-separated integers, describing the commanders where the ith integer is the planet that the ith commander is currently on.

Output

Output a line containing the string `DAN` if the task is possible, or the string `DAN'T` if the task is impossible.

In addition, if the task is possible, output $k+1$ more lines describing the plan.

- First, output a line containing an integer d, the number of times each commander uses a teleporter.
- Then, output k lines, where the ith of these lines contains the route for the ith commander (same order as they were given in the input). Each line should contain d space-separated integers, the sequence of planets that each commander should visit. The first planet should have a teleporter that directly connects it with the commander's starting planet, and every succeeding planet should have a teleporter that directly connects it with the previous one.

Any answer will be accepted so long as d is minimized, and all commanders end on the same planet (any planet).

Constraints

- $2 \le k \le 100$ and $k \le n$
- $2 \le n \le 9000$
- $0 \le m \le 6 \times 10^5$
- For all teleporter pairs, $u \ne v$, and there exists at most one teleporter pair. directly connecting two planets
- No two commanders start on the same planet.

Sample Input 1

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

Sample Output 1

```
DAN
2
2 3
4 3
8 3
```

Sample Input 2

```
2 1 2
1 2
1 2
```

Sample Output 2

```
DAN'T
```

Sample Input 3

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

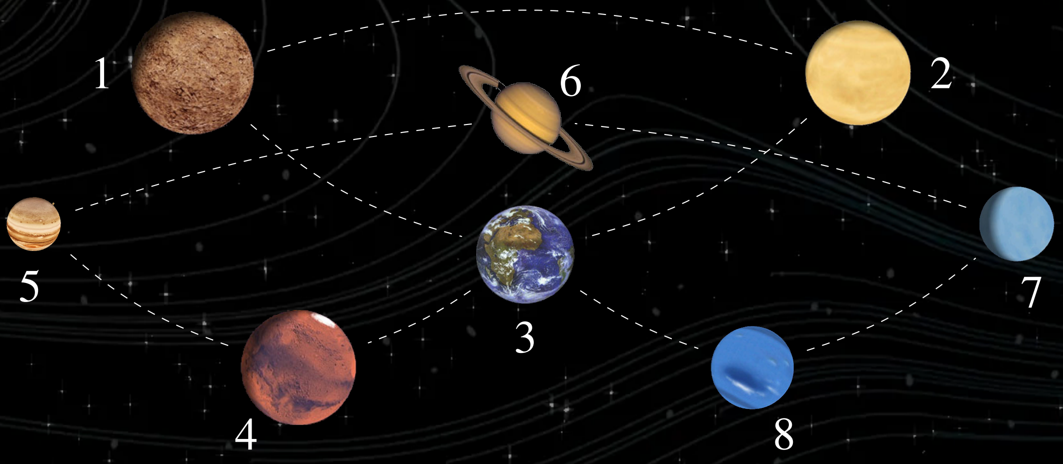
Sample Output 3

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

Explanation

In Sample Input 1, the star system has $n = 8$ planets, and there are $k = 3$ commanders, initially on planets 1, 5 and 7, respectively.

The following is an illustration of the system:



Sample Output 1 describes a plan where each commander uses a teleporter $d = 2$ times:

1. At the start of day 1, the commander on planet 1 teleports to planet 2, the one on planet 5 teleports to planet 4, and the one on planet 7 teleports to planet 8.

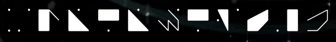
2. At the start of day 2, each of the commanders teleports to planet 3.

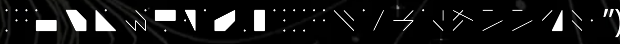
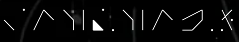
Since all commanders ended up on planet 3, the plan is valid. Also, it can be shown that 2 is the minimum possible d, so the answer is accepted.

Problem D
Eliens Slurs
“ ”

Time Limit: 4 seconds

Even 10,000 years since its advent, Social Media is still a big thing on the inter(galactic)net. One of the leading platforms is Tatooitter, enabling cosmic entities to broadcast their thoughts throughout the Universe Wide Web.

The language of the Eliens race, or **Eliensese**, is the common and most used language in Tatooitter. The language itself is rich and contains 300 unique characters in its alphabet. For example, “” is a standard way of expressing affection in Eliensese. Figure 1 shows the full Eliensese alphabet, should you wish to see it.

Some however use Tatooitter to insult others using **Eliens slurs** (or in Eliensese, “”). Some examples of Eliens slurs are “ ”, “” and “ ”, the latter two of which are commonly used by those in high-stress work-environments, such as chefs. Some even find ways to incorporate a given slur or slur-like phrase multiple times in a single tweet. This is causing problems with Tatooitter and their moderation principles. Now they are devising a way to catch all violators that use a particular slur and slur-like phrase.

A slur S is a given sequence of s characters. A string L is **slur-like** if it's of length s and each character L_i is off by at most one from S_i. More formally, let $\mathrm{p}(c)$ be the position of the character c within the Eliensese alphabet. Then L is slur-like if the length of L is s, and $\mathrm{p}(S_i)-1 \le \mathrm{p}(L_i) \le \mathrm{p}(S_i)+1$ for all $1 \le i \le s$. Note that the alphabet doesn't wrap around, i.e., there is no 301st character or 0th character in the alphabet.

Now, given a tweet and a slur, help Tatooitter find the number of slur-like substrings of the tweet. For easier parsing in programming languages that cannot encode/decode Eliensese, the tweet and slur are each given as a sequence of alphabet positions of each character.

Input

The input consists of three lines describing a single test case.

- The first line contains two space-separated integers t and s, the length of the tweet T and the length of the slur S respectively.
- The second line contains t space-separated integers $\mathrm{p}(T_1), \mathrm{p}(T_2), \ldots, \mathrm{p}(T_t)$ describing the tweet.
- The third line contains s space-separated integers $\mathrm{p}(S_1), \mathrm{p}(S_2), \ldots \mathrm{p}(S_s)$ describing the slur.

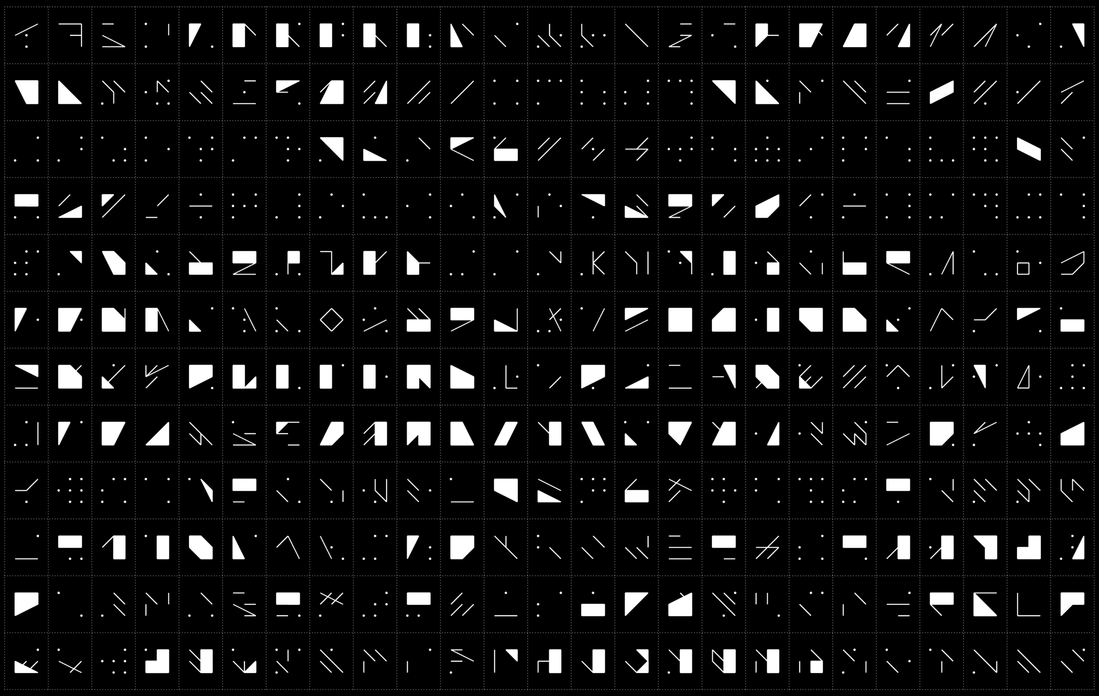


Figure 1: The full Eliensese alphabet in alphabetical order from left to right, then top to bottom.

Output

Output two lines.

- On the first line, output an integer c, the number of slur-like substrings in the tweet.
- On the second line, output c space-separated integers $i_1, i_2, \ldots, i_c$ denoting the locations of the slur-like substrings in increasing order. Formally, i_k is the one-based index of the first character of the kth slur-like substring of T from the left.

Constraints

- $1 \leq t \leq 9 \times 10^5$
- $1 \leq s \leq 1.25 \times 10^5$
- $s \leq t$
- $1 \leq \mathrm{p}(T_i), \mathrm{p}(S_i) \leq 300$

Sample Input 1	Sample Output 1

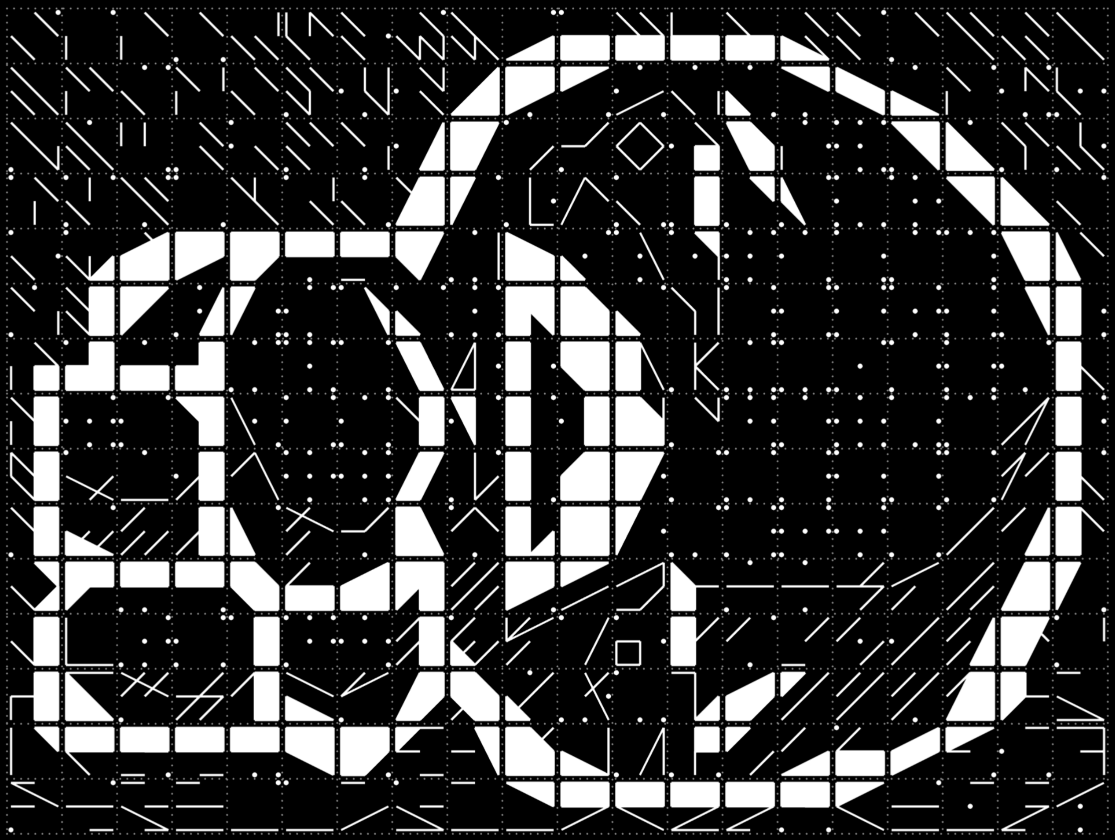
14 3
1 3 2 2 1 5 1 2 1 6 1 2 2 1
1 2 1
	5
	1 3 7 11 12

Sample Input 2	Sample Output 2

5 2
1 300 300 1 299
1 300
	2
	1 4

Note

Actually, the writing direction of Eliensese is different from English. Eliensese is usually written from bottom to top then from right to left. For example, here's how the Eliensese alphabet is typically written by natives:



Problem E

Euclidean Travel with Parallel Universes

Time Limit: 2.5 seconds

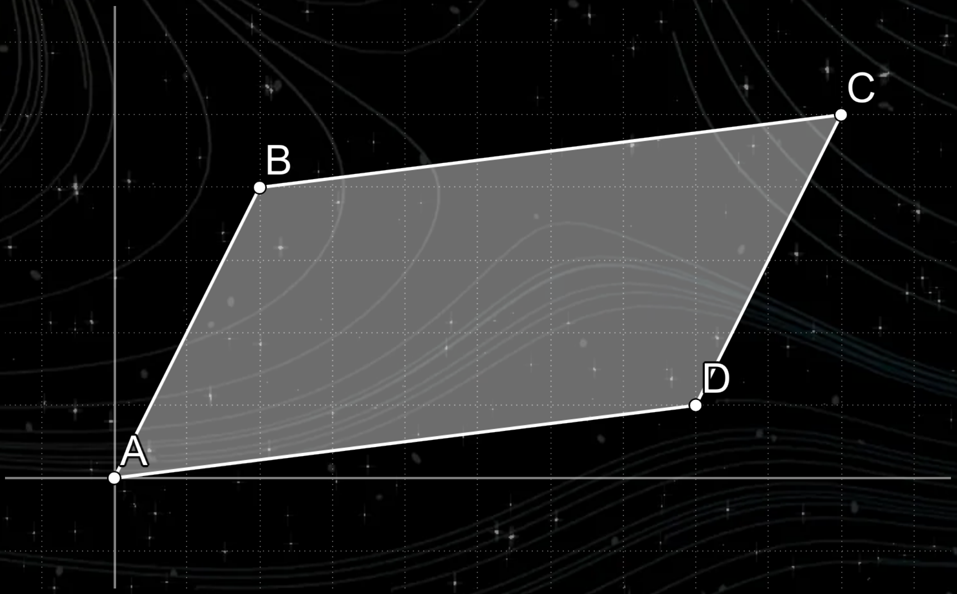
Welcome to today's episode of Conquistadora the Explorer! Conquistadora is thinking of expanding her nation's empire through violent conquest across the galaxy, but in order to properly organize the logistics of her army, she needs *your* help in understanding the geometry of the known universe!

First let's establish some inviolable truths. Recall that our universe is shaped like a *parallelogram*. Let's label its four vertices A, B, C, and D. For ease of discussion, we assign (x,y) coordinates to each of its coordinates as follows:

- Point A is at $(0,0)$
- Point B is at (x_1,y_1)
- Point D is at (x_2,y_2)
- Point C is at (x_1+x_2,y_1+y_2)

We note that the four values x_1, y_1, x_2, and y_2 are sufficient to completely describe the parallelogram. These values need not be positive.

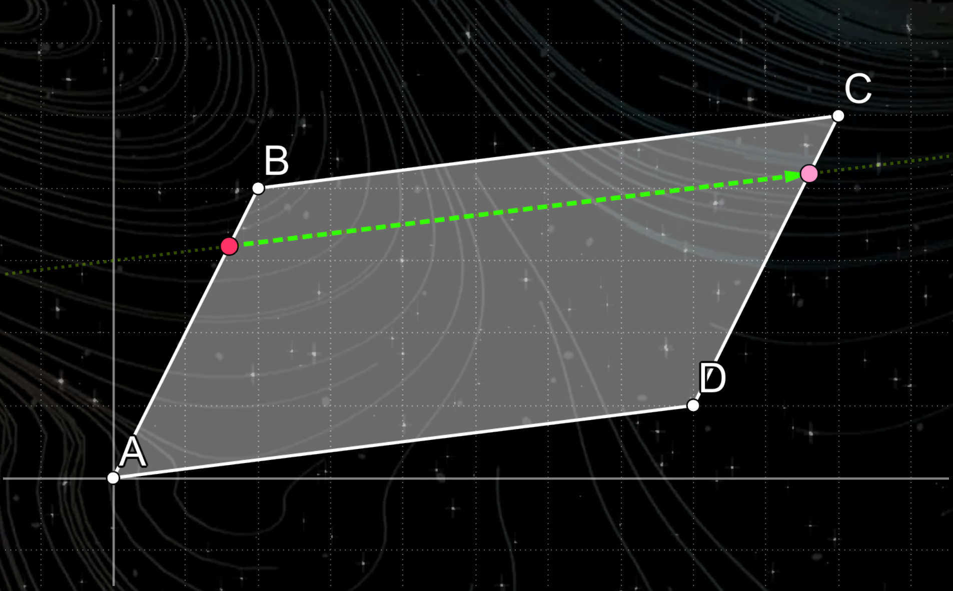
As an example, if $x_1=2$, $y_1=4$, $x_2=8$, and $y_2=1$, we get this parallelogram.



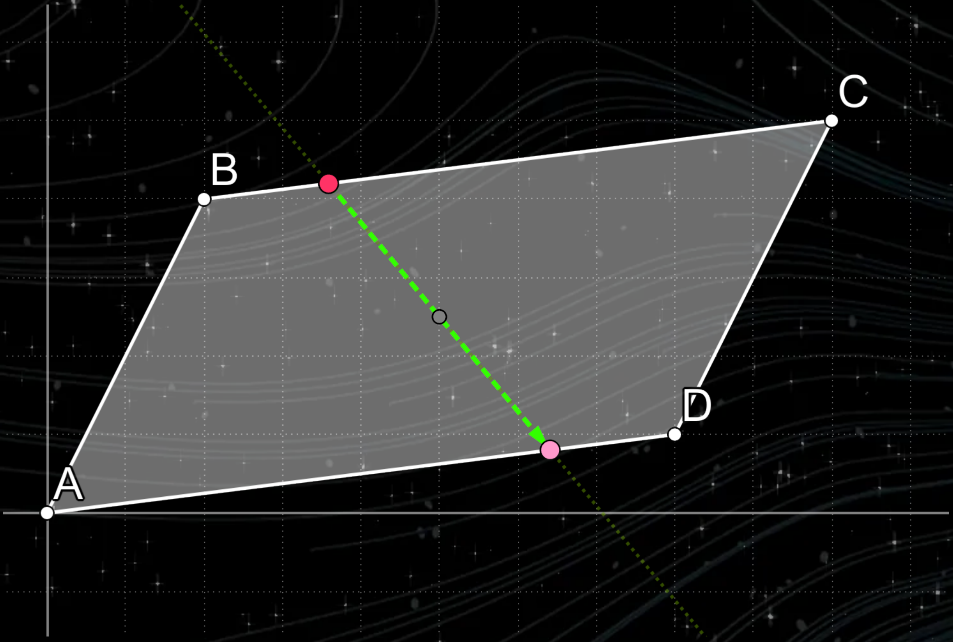
There are many ways to travel through our universe! First, there is the straightforward way. Suppose points (a,b) and (c,d) are both in the parallelogram. Then, you can directly walk from (a,b) to (c,d) (like a peasant) at a rate of 1 unit/s. Thus, this travel option takes $\sqrt{(a-c)^2+(b-d)^2}$ seconds.

We can also abuse the wonky geometry of the universe and use the wormholes along the edges of the world.

Next, sides AB and DC are teleportically connected by wormholes in a "straight" fashion. Formally, consider any line that is parallel to both BC and AD, and consider the point where it intersects with AB and the point where it intersects with DC. Anyone at the point on AB can teleport to the corresponding point on DC, and vice versa, taking 0 seconds.



Finally, sides BC and AD, meanwhile, are teleportically connected by wormholes in an "opposite" fashion. Formally, consider any line that passes through the center of the parallelogram, and consider the point where it intersects with BC and the point where it intersects with AD. Anyone at the point on BC can teleport to the corresponding point on AD, and vice versa, taking 0 seconds.



Note that the vertices A, B, C, D themselves belong to two different sides, so wormholes of both kinds are available from those positions. We also stress that *you are not permitted to leave the parallelogram at any time* (not that you would ever want to, anyway).

Now, given the shape of the parallelogram, as well as two points (x_s, y_s) and (x_t, y_t) that are in it, find the minimum possible time it takes to travel from the former point to the latter.

Input

The first line of input contains an integer T, the number of test cases. T test cases follow.

Each test case is described by a single line with the eight space-separated integers x_1, y_1, x_2, y_2, x_s, y_s, x_t, and y_t.

Output

For each test case, output a single line containing a single non-negative real number, which is your answer for that test case. Your answer will be accepted if it has an absolute or relative error of at most 10^{-8} from the judge's answer. In symbols, if a is your answer and b is the judge's answer, then your answer will be accepted if and only if

$$\frac{|a-b|}{\max(1,|a|,|b|)} \le 10^{-8}.$$

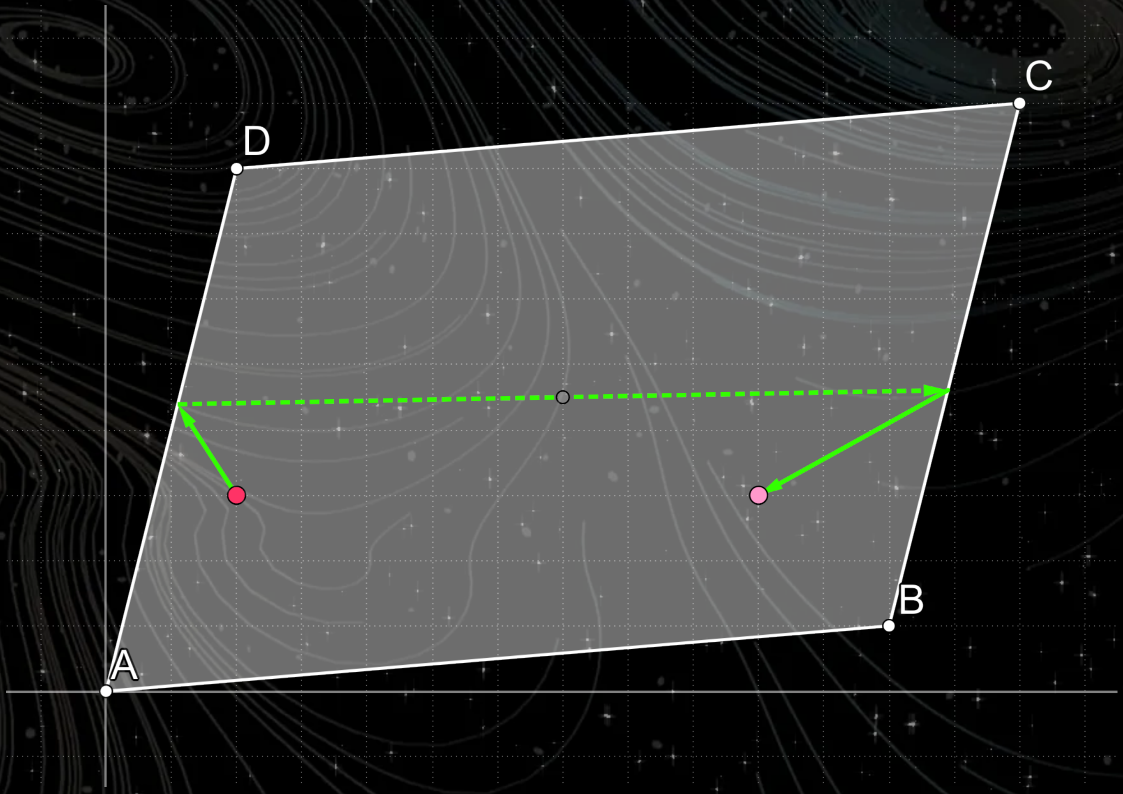
Constraints

- $1 \le T \le 10^5$
- The absolute values of all coordinates are $\le 10^5$.
- All four of A, B, C and D are distinct.
- No three of A, B, C, D are collinear.
- (x_s, y_s) and (x_t, y_t) are both inside or along the boundary of the parallelogram.

Sample Input	Sample Output
1 12 1 2 8 2 3 10 3	4.976414963

Explanation

The following illustrates one way to travel from $(x_s, y_s) = (2,3)$ to $(x_t, y_t) = (10,3)$ with the minimum possible time of ≈ 4.976414963 seconds:



Problem F

Flow Maximal

Time Limit: 2 seconds

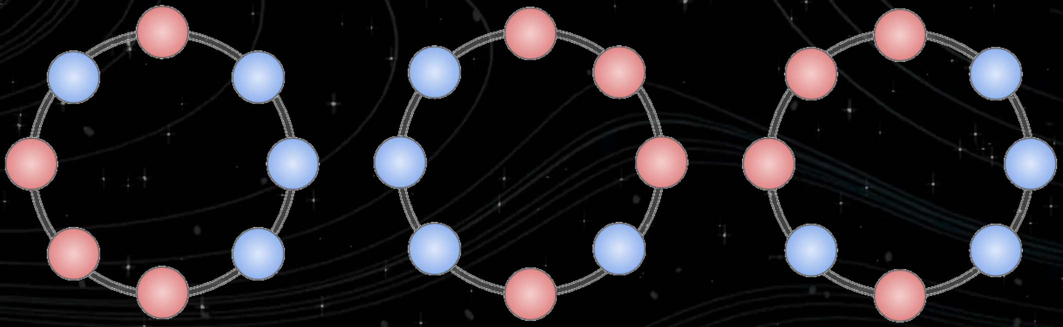
We normally think of galaxies as gigantic, given that they are composed of billions of stars. However, in this wild and fascinating world, it's actually possible for a galaxy to be tiny—so tiny that it could fit on a bead in a charm on the collar of a cat.

You have some more galaxies, so let's create more galactic jewelry!

Each of our **beads** contains a galaxy, so we call them **Cosmic**. Cosmic beads look exactly the same from any direction, regardless of how it has been rotated or flipped upside-down. Cosmic beads that contain spiral galaxies are called **Andromedal**. An Andromedal bead is clearly distinguishable from a non-Andromedal bead, and vice versa.

We can take several Cosmic beads and connect them together in a **chain**. A chain is formed out of c Cosmic beads that are arranged in a circle, connected by c separate links (where each link connects two adjacent beads in the circle). We use the term "(a,c)**-chain**" to denote a chain built out of c Cosmic beads, such that exactly a of those beads are Andromedal.

Two chains are said to be **equivalent** if it is possible to make one look like the other by rotating the chain and/or flipping it over. Here are some examples of $(4,8)$-chains that are actually all equivalent. We color the Andromedal beads in red, and the non-Andromedal beads in blue.



A link is said to have **flow** if it connects an Andromedal bead and a non-Andromedal bead. An (a,c)-chain is called **flow-maximal** if, among all possible (a,c)-chains, it has the maximum number of links that are in flow. The chain shown above has 4 links that are in flow, and we can show that it is **not** flow-maximal.

Now, let's suppose that $f(a,c)$ *counts* the number of different flow-maximal (a,c)-chains. We could just ask you to evaluate $f(a,c)$... but we can do better than that!

Given some non-negative integer b, evaluate the *sum* of $f(a,c)$ over all pairs of non-negative integers a and c such that $0 \leq a \leq c$ and $a^2 + b^2 = c^2$. The answer can be quite huge, so compute it modulo $998244353 = 7 \times 17 \times 2^{23} + 1$, a prime number. If the sum does not have a finite value, then you should report so as well.

Input

Input consists of a single line with a single non-negative integer b.

Output

Output a single line containing the answer.

- If the desired sum has a finite value, then output that value, taken modulo 998244353.
- If not, then output the string `INFINITE` instead.

Constraints

- $0 \leq b \leq 5 \times 10^5$

Sample Input	Sample Output
9	14

Explanation

For $b = 9$, there are three pairs (a,c) such that $0 \leq a \leq c$ and $a^2 + b^2 = c^2$, namely $(0,9)$, $(12,15)$ and $(40,41)$. The corresponding $f(a,c)$ are:

- $f(0,9) = 1$,
- $f(12,15) = 12$,
- $f(40,41) = 1$.

Therefore, the required sum is $1 + 12 + 1 = 14$ which modulo 998244353 is also 14.

Problem G
Irreversible Events
Time Limit: 2.5 seconds

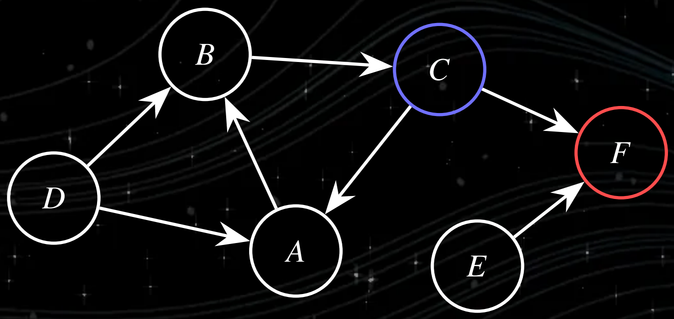
Time travel is now so common, and paradoxes are everywhere. At this rate, it will lead to the collapse of the universe! It is up to the Association for Continuity Mitigation (or ACM) to hunt down these paradoxes like some badass time police.

The core of these problems lies in the transition of events. Here, an *event* is a point in space-time. There are also *transitions* from one event to another, i.e., an event A can reach an event B directly if there is an event transition (A,B). Note that transitions are one-way, that is, a transition (A,B) does not imply that B can reach A directly.

We say that an event B is **reachable** from event A if there is a sequence of event transitions that leads from A to B. And if an event B is reachable from event A, but not the other way around, then we say that B is **irreversible** from A, and any path from A to B is called an **irreversible path**.

It's possible that there are events A and B wherein A can reach B and conversely B can reach A. This is called a time loop, and is none of our concern. What *is* our concern is that paradoxes occur whenever any two irreversible paths to an event X *diverge*. To be more precise, we define an event to be **divergent** if there exists any two irreversible paths towards X wherein the two paths do not share a common event transition. Otherwise the event X is **in continuity**.

Here's an example, where circles are events and arrows are event transitions.



In this example, event C is in continuity because the only two irreversible paths ($D \to B \to C$ and $D \to A \to B \to C$) share a common event transition ($B \to C$). In contrast, F is divergent because there are two irreversible paths ($D \to B \to C \to F$ and $E \to F$) that do not share a common event transition.

ACM is given the ability to prohibit some event transitions, effectively removing them, so that all events are left to be in continuity. What is the minimum number of event transitions that they need to prohibit?

Input

The input begins with a single integer T, the number of test cases. T test cases follow.

The first line of each test case contains two integers n and m, the number of events and event transitions, respectively. The events are labeled from 1 to n.

Then m lines follow, the ith of which contains two integers a_i and b_i, denoting an event transition from a_i to b_i.

Output

For each test case, output a line containing the minimum number of event transitions that ACM needs to prohibit, so that all events are in continuity.

Constraints

- $1 \leq T \leq 3000$
- $1 \leq n \leq 3 \times 10^5$
- $0 \leq m \leq 4 \times 10^5$
- The sum of all n across all t cases will not exceed 3×10^5.
- The sum of all m across all t cases will not exceed 4×10^5.
- There may be multiple transitions from one event to another.
- There are no transitions from an event to itself.

Sample Input

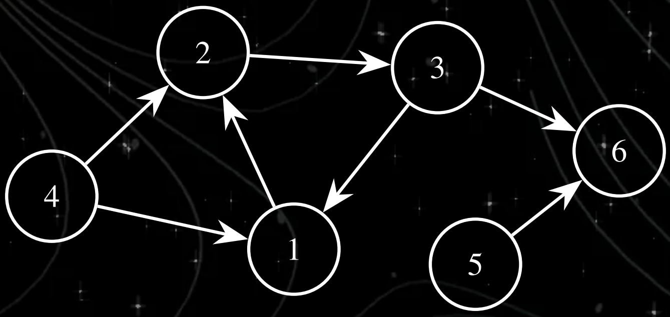
```
3
3 2
1 2
2 3
4 3
1 2
2 3
1 3
6 7
4 2
4 1
1 2
2 3
3 1
5 6
3 6
```

Sample Output

```
0
1
2
```

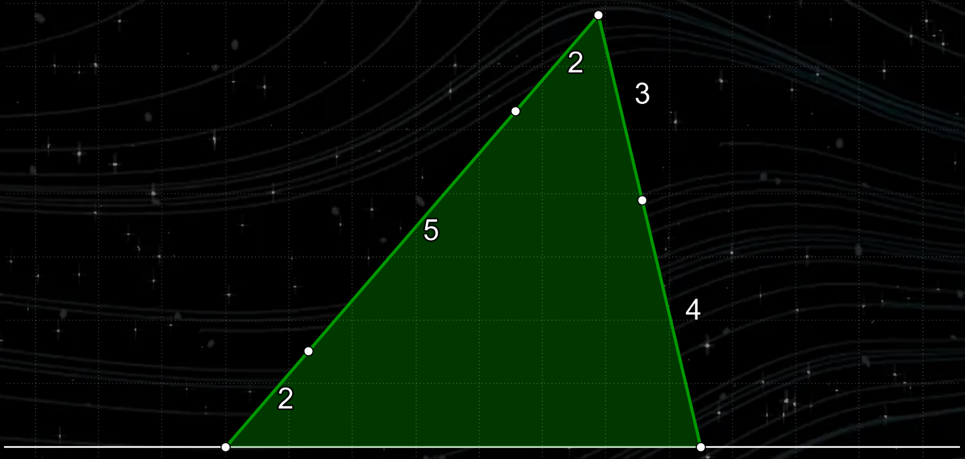
Explanation

Here is an illustration of the third test case:



Problem H
Not Just an NP-Hard Problem
Time Limit: 1.5 seconds

Antares Intergalactica is hiring new software developers! To be considered for the position, you need to pass a simple coding interview. Here's an item taken from their archives:

> Suppose that you have n sticks that we model as line segments whose lengths are $x_1, x_2, \ldots, x_n$. A **beam** is some collection of sticks attached end-to-end **parallel to each other** to form a longer line segment. We then create a shelter by the following method:
>
> - Create exactly two beams using only the n sticks. Note that if a stick is used in one beam, it cannot be used in the other beam. You also do not need to use all the sticks.
> - Take one endpoint from each beam and attach them to each other, joining the two beams together at some angle of your choosing (strictly between 0° and 180°).
> - Rest this structure on the ground, such that each beam's other endpoint is touching the ground.
>
> We model the ground as an infinitely long *flat* line. The **spaciousness** of this shelter is equal to the area of the region enclosed by the two beams and the ground.
>
> Consider the following example, where our sticks have lengths 2, 2, 3, 4, and 5.
>
> 
>
>
> We can assemble two beams of lengths $2+5+2=9$ and $3+4=7$. If we attach them together at the angle shown in the diagram, then rest the shelter on the floor, we can use Heron's Formula to

> show that the enclosed region has an area of approximately 25.54 square units. This is not the maximum possible spaciousness, though—if we use our sticks more optimally, we can construct a shelter that encloses an even greater area.
>
> Given the lengths of the sticks, the tribes' original challenge to one another was to determine the maximum possible spaciousness of a shelter constructed by this method.

This is already quite a hard problem. In fact, it is an *NP-Hard* problem, which can be proven by the fact that there exists a polynomial-time reduction from the Degree-Constrained Spanning Tree problem to our Maximum-Spaciousness Shelter problem.

But Antares Intergalactica's participants grow stronger and stronger, and they need *even harder* coding questions in order to properly weed out the cream of the crop. They don't just want NP-Hard problems anymore! So they decide to make the problem even more complicated by adding the following extra twist:

> Before building the beams, you must first **break** one of the sticks into two smaller ones. The smaller sticks must *also* have integer lengths.

Formally, select an index i and two positive integers a and b such that $x_i = a+b$. We then delete stick i and insert two new sticks $n+1$ and $n+2$ such that $x_{n+1} = a$ and $x_{n+2} = b$. To minimize messiness, we do not re-label the indices of the remaining sticks and just let index i be missing.

Since its introduction, *no one* has successfully solved the Modified Maximum-Spaciousness Shelter Problem. Can you?

Input

The first line of input contains an integer T, the number of test cases. T test cases follow.

The first line of each test case begins with a line containing a single integer n.

This is followed by a line containing n space-separated integers $x_1, x_2, \ldots, x_n$.

Output

For each test case, first output a line containing three space-separated positive integers i, a, and b, corresponding to the "break" operation described above. Note that $x_i = a+b$ must hold.

Then, output two lines, each containing some number of space-separated indices. The indices in the first of these lines corresponds to the sticks you wish to use in the first beam, and the indices in the second of these lines corresponds to the sticks you wish to use in the second beam.

Note that each beam must contain at least one stick. The sticks must have indices from 1 to $n+2$ only, **except** for index i (which you cannot use, because you broke that stick). You are reminded that $x_{n+1} = a$ and $x_{n+2} = b$, as dictated by your *break* operation. No index must be used in both beams. Again, you do not need to use all the sticks.

Finally, output a line containing a single real number, the spaciousness of the shelter that you wish to build.

Your answer will be accepted if and only if both of the following are true:

- It is actually possible to build a shelter with this spaciousness, using the two beams that you have constructed.
- Your answer has an absolute or relative error of at most 10^{-11} away from the mathematically optimal answer. In symbols, if a is your answer and b is the judge's answer, then your answer will be accepted if and only if
$$\frac{|a-b|}{\max(1,|a|,|b|)} \le 10^{-11}.$$

If there are multiple possible answers, any will be accepted.

Constraints

- $1 \le T \le 42$
- $2 \le n \le 35$
- $2 \le x_i \le 10^8$ for each i

Sample Input	Sample Output

1
3
6 7 6
	1 3 3
	2 4
	3 5
	45.000000

Explanation

The lengths of the sticks are

$$[x_1, x_2, x_3] = [6, 7, 6].$$

The solution chose to break stick 1 into pieces of lengths $a = 3$ and $b = 3$, yielding the sticks

$$[x_1, x_2, x_3, x_4, x_5] = [missing, 7, 6, 3, 3].$$

It then places sticks 2 and 4 in the first beam, and sticks 3 and 5 in the second beam. Finally, it claims that it can use these beams to create a shelter with a spaciousness of 45.000000.

Problem I
Ominous Acids
Time Limit: 10 seconds

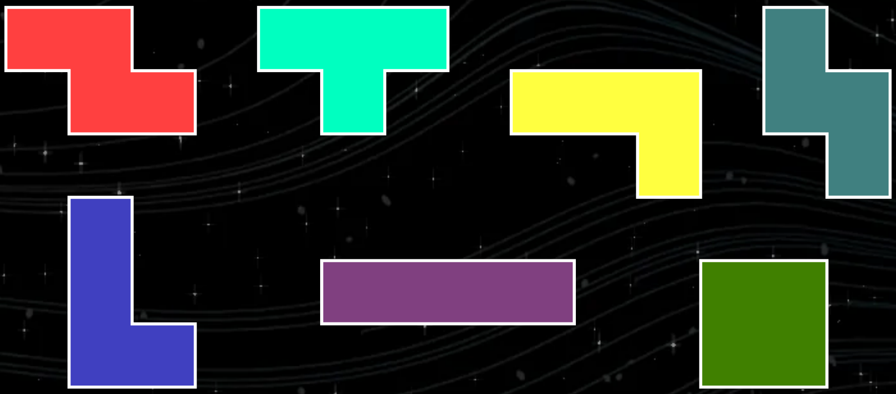
The Goldilocks Cantina is renowned across the universe for their signature *Primordial Soup*, a nutritious and delicious dish which tastes like it's *brimming* with life. The soup is famous for containing all the different kinds of ominous acids, the building blocks of all Alien Life.

However, they have noticed that aliens these days are *just too busy* to sit down and enjoy a nice bowl of soup. Therefore, they have hired you to create a new product for them that contains all the nutritious and life-bestowing ominous acids in a convenient and ready-to-go package: a **protein bar**!

Let's look at the fundamental structure of an ominous acid. An **ominous acid** is a geometric figure formed by joining together exactly k different unit squares, subject to the following constraints:

- There is no overlap of non-zero area between unit squares.
- If two unit squares touch, then they either share an *entire* edge, or only touch at corners.
- The entire ominous acid is *connected*. Formally, suppose that a prokaryote can hop between two squares on the ominous acid if and only if those squares share an edge; then, the prokaryote should be able to travel from any square to any other square by some sequence of hops.

As an example, here is a diagram showing the 7 different ominous acids if $k = 4$.



Note that we said "different": formally, two ominous acids are the same if it is possible to transform one into the other by some combination of rotations and shifting up-and-down. Notably, we consider mirror images to **not** necessarily be the same as each other.

Given k, create any rectangular protein bar that contains **all** different ominous acids at least once each, *with no extra or wasted space* (we will not compromise on nutrition or quality!). You are even free to choose whatever dimensions you want for the protein bar (within reason, of course). Of course, if the task is impossible, please do say so to your client.

Input

The input consists of a single line containing a single positive integer k.

Output

Output a line containing the word:

- `Ominous` if the task is possible, or
- `Ignominious` if it is impossible.

In addition, if the task is possible, you need to construct a valid protein bar.

First, output a line containing two space-separated positive integers r and c, the number of rows and the number of columns in your protein bar.

Then, output your protein bar as an ASCII grid. Output r lines, each containing c characters, encoding the grid. Each character represents a unit square, and should be an uppercase English letter. If a unit square is represented by the same letter as a square directly above or below it or to its left or right, then we assume that these two squares belong to the same ominous acid and are joined together.

Your answer will be considered correct if all of the following are satisfied:

- $1 \leq r \leq 2023$ and $1 \leq c \leq 2023$;
- Every square in the protein bar belongs to a figure that is an ominous acid;
- Every possible ominous acid appears at least once somewhere in the protein.

Remember that every ominous acid has exactly k unit squares.

We can show that if the task is possible, then a solution exists with the given bounds for r and c.

Caution: Check your spelling! It should be `Ignominious` or `Ominous`.

Constraints

- $1 \leq k \leq 15$

Sample Input

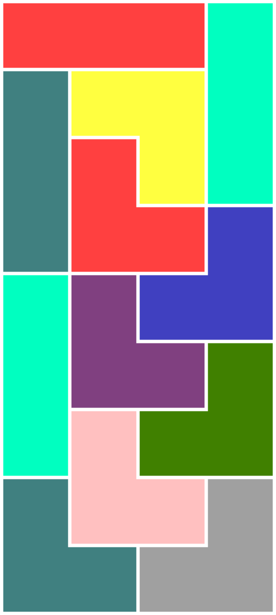
```
3
```

Sample Output

```
Ominous
9 4
OOOH
WAAH
WOAH
WOOT
HUTT
HUUY
HEYY
WEEB
WWBB
```

Explanation

The following illustrates the sample output:



Note that there are two different ominous acids for $k = 3$, and they both appear in this protein bar. There are many other solutions, some of which are smaller than this. They will all be accepted as long as all the different ominous acids appear in them.

Problem J
Sensor Logs

Time Limit: 1 second

The nuclear launch codes have been stolen! The commander of the Nebula Star Cruiser calls in all of the staff to the boardroom.

"Has the ship been sealed?" He asks one of the soldiers who nods in response. "We can't let those codes get to NELOC or else we could lose the war. No one leaves until I find the impostor!"

"Good thing we have someone from MI7 here to figure it out. So how is it going, Agent Blanc?"

The commander provides you with a blueprint of the star cruiser, made up of four rooms (labeled with letters A, B, C, and D) and three corridors (labeled with numbers 0, 1, and 2). Room A also contains the sleeping quarters for all n employees (whose IDs are numbered from 0 to $n-1$), and everyone starts there at the beginning of the day. A visualization of the room and corridor layout is depicted in Figure 2.

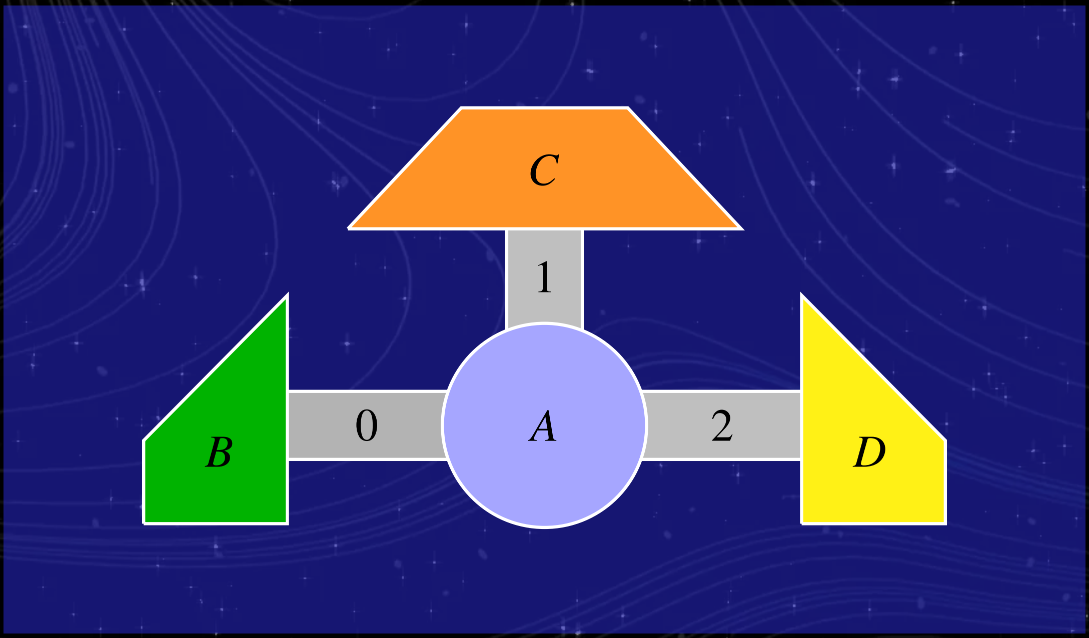


Figure 2: The base blueprints

You then take a look at the corridor access logs, which each automatically record a person's ID number whenever they pass through that corridor. Note that these sensors only trigger when a person exits the corridor into a room different from the one they entered it from.

A sample log is located below, which tells us that Employee ID 0 passed Corridor 0 into Room B, then Employee ID 5 passed Corridor 2 into Room D, and lastly Employee ID 0 passed Corridor 0 again to Room A:

```
0 0
5 2
0 0
```

From reading the blueprints earlier, you recall that a series of *vents* are built into the building, connecting all of the corridors to one another. This allows an intruder to enter a corridor, go through the vents to a different corridor, then trigger the access sensor in that different corridor.

You sift through the ventilation sensor logs and notice that an alarm was triggered that might signal movement in the vents. Unfortunately, these logs do not include the time and place the vents were entered from. In addition, one of the staff also tells you that rats in the vents may also trigger the alarm accidentally (meaning no actual person entered the vents).

A person is **suspicious** if their records on the sensor logs would only make sense if they are capable of using the vents; otherwise, we believe in innocent-until-proven-guilty. Identify who among the employees are suspicious (if any).

Input

The input consists of a single test case. The first line contains two space separated integers n and ℓ, the number of people and the number of lines in the access logs, respectively.

Then, ℓ lines follow, the ith of which contains two space-separated integers e_i and c_i, meaning that the person with Employee ID e_i passed through the corridor c_i. Note that employee IDs are unique per person and assigned in increasing order from 0 to $n-1$.

Output

Output a single line containing a string of length n, where the ith character (zero-indexed) is `1` if Employee ID i is suspicious, and `0` if they are presumed innocent.

Constraints

- $1 \leq n \leq 1000$
- $1 \leq \ell \leq 10^5$
- $0 \leq e_i \leq n-1$
- $c_i \in \{0, 1, 2\}$

Sample Input

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

Sample Output

```
010
```

Explanation

Based on the logs, Employee ID 0 moved from Room A to Room B, Room B to Room A, then finally Room A to Room D. This path is valid and can be conducted without passing a vent. Therefore, Employee ID 0 is presumed innocent.

For Employee ID 1, they first moved from Room A into Room C. Afterwards, they appeared to be in Corridor 2, which is not directly reachable from Room C. The only way this can be true is by using the vents to travel to Corridor 2. Therefore Employee ID 1 is suspicious and could be an impostor.

Lastly, we see no logs on the movement of Employee ID 2; they just stayed in Room A which makes them presumed innocent.

Problem K
Star Seeker's Socks
Time Limit: 4 seconds

Within the Star's Inquisition, socks are a symbol of prestige. Star Seeker Blanc loves to collect various kinds of socks. Short socks, long socks, brightly colored socks – she has collected almost every type of sock. She has only one rule in wearing them, and it's that her socks need to be *left-right neutral*, i.e., no difference of sock type between the left foot and the right foot.

Blanc's collection has grown to contain n types of socks, and she labels the types of socks with an integer from 1 to n. She has k_i sock pairs of type i; each pair is composed of 2 individual socks (obviously), for a total of $2k_i$ individual socks of type i. Socks of the same type are indistinguishable from one another.

Today is the Yellow Wedding, and only m sock types would be suitable for the event. These are given as types $t_1, t_2, \dots, t_m$. Blanc was about to get a suitable pair of socks from her walk-in sock closet when suddenly the only light in the room went out. Unfortunately, she was intoxicated from the pre-wedding festivities the previous night, and for some reason all her socks were heaped into a pile in the closet. The closet being in complete darkness, she can only get socks randomly from the heap.

Since she's in a hurry to get to the ceremony, she decides to just pick up a large clump of random individual socks, then just pick out a suitable pair from this clump on the way. Thinking quickly — what is the minimum number of individual socks she needs to grab from the closet to guarantee that she gets at least one suitable pair of socks?

Input

The input begins with a line containing an integer T, denoting the number of test cases. The descriptions of T test cases follow.

The first line of each test case contains two space separated integers n and m, the number of sock types and the number of sock types *suitable for the Yellow Wedding*.

The second line of each test case contains n space-separated integers $k_1, k_2, \dots, k_n$, where k_i is the number of pairs of the ith sock type.

The third and last line of each test case contains m space-separated integers $t_1, t_2, \dots, t_m$ denoting the labels of the suitable sock types.

Output

For each test case, output a single integer—the number of socks Blanc needs to grab from the closet to guarantee she gets at least one suitable pair of socks.

Constraints

- $1 \leq T \leq 100$
- $1 \leq n \leq 10^3$
- $1 \leq m \leq n$
- $1 \leq k_i \leq 10^5$
- $1 \leq t_j \leq n$
- All t_j are unique in a case.

Sample Input

```
3
1 1
1
1
2 2
6 4
1 2
5 3
3 5 7 11 13
1 2 4
```

Sample Output

```
2
3
44
```

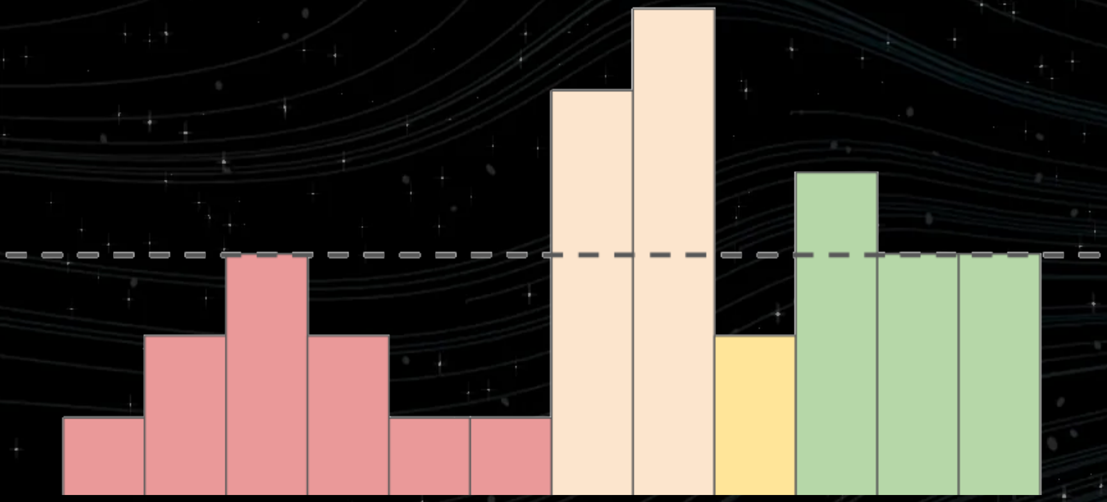
Problem L
Starquake!
Time Limit: 4.5 seconds

Many stars, like our sun, are massive balls of gas and plasma. However, there do exist stars that are dense enough to have solid surfaces. In fact, neutron stars are perhaps one of the densest objects *in the entire galaxy*. But a solid surface means we can have tectonic interactions. And that means we can have... *starquakes!*

You are part of a group of scientists analyzing a locally-flat area on the surface of a neutron star. We divide this area into n **sections**, arranged in a row and labeled 1 to n from left to right, whose heights are given by $h_1, h_2, h_3, \ldots, h_n$. The height of the ith section is h_i. Note that h_i need not be positive—a height of 0 corresponds to "sea level", and a negative height corresponds to sections whose peaks are "below sea level".

A **landmass** is a non-empty collection of sections which can be "grouped together", in some sense. Consider an alien that can walk from one section to an adjacent section if and only if the absolute difference between their heights is at most 1. Two sections i and j are considered to be in the same landmass if and only if this alien can walk all the way from i to j.

For example, suppose that the heights of the sections are given by the sequence $\{-2,-1,0,-1,-2,-2,2,3,-1,1,0,0\}$. We illustrate this in the following image, where the dotted line represents sea level. Note that sections belonging to the same landmass have been given the same color. In particular, we highlight the fact that sections 1 and 6 belong to the same landmass, but sections 4 and 9 belong to separate landmasses.



Your program should be able to answer a landmass-counting range query while also being able to handle updates to the topology of the neutron star. Specifically, your program must be able to process the following commands:

- `QUERY` i j
  Considering only the sections $h_i, h_{i+1}, h_{i+2}, \ldots, h_j$, how many distinct landmasses are there in this range?
- `FISSURE` i j d
  Decrease each of the heights of sections $h_i, h_{i+1}, h_{i+2}, \ldots, h_j$ by d.
- `STARQUAKE` i j
  Decrease each of the heights of sections $h_i, h_{i+2}, h_{i+4}, \ldots, h_j$ by 1.
  Note this only decreases the height of every **other** section in the range.
  It is guaranteed that $j - i$ will be even.

Input

The input begins with a single line containing two space-separated integers n and c, the number of sections and the number of commands, respectively.

This is followed by a line with n space-separated integers $h_1, h_2, h_3, \ldots, h_n$.

Then, c lines follow, each describing a command. Each one is given in the format of one of the commands described previously in the problem statement.

Output

For each `QUERY` command, output a line containing a single integer—the number of distinct landmasses in the range indicated by that query.

Constraints

- $1 \leq n, c \leq 2.5 \times 10^5$
- $|h_i| \leq 10^7$ for each i
- For each command, $1 \leq i \leq j \leq n$.
- For each `FISSURE` command, $0 \leq d \leq 4000$.
- For each `STARQUAKE` command, $j - i$ is guaranteed to be even.
- There is always at least one `QUERY` command.

Sample Input

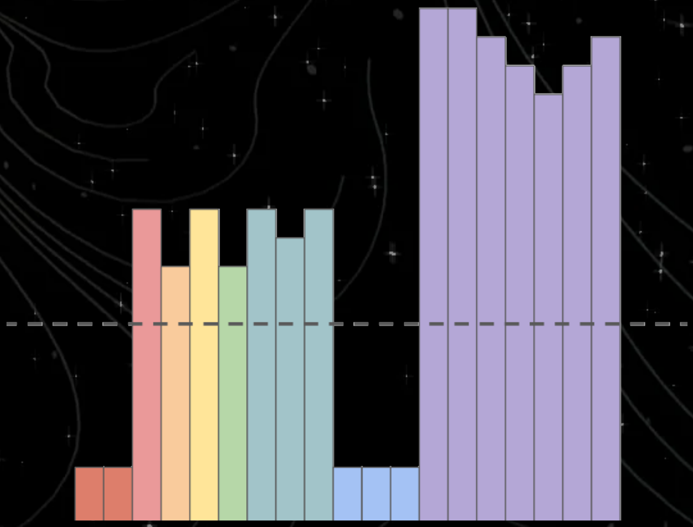
```
19 7
-5 -5 4 2 4 2 4 3 4 -5 -5 -5 11 11 10 9 8 9 10
QUERY 1 19
QUERY 2 11
STARQUAKE 3 11
QUERY 2 11
FISSURE 4 9 7
QUERY 2 11
QUERY 1 19
```

Sample Output

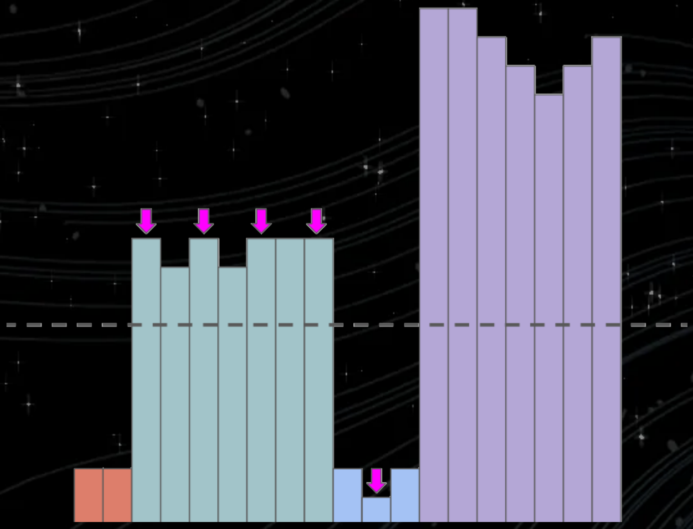
```
8
7
3
3
4
```

Explanation

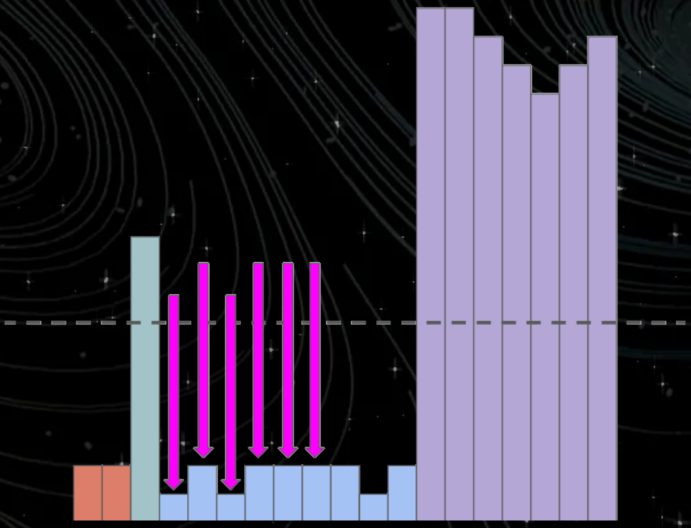
Here is an illustration of the neutron star’s initial topography:



Here is the topography after the STARQUAKE event on sections 3 to 11:



Here is the topography after the `FISSURE` event of strength 7 on sections 4 to 9:



Problem M
TheBuzz
Time Limit: 2 seconds

TheBuzz is a decentralized intergalactic hacktivist collective primarily known for sharing confidential information about governments, institutions, agencies, and corporations.

Periodically, a series of cryptic tweets would be posted on Tatooiter containing nothing more than a single character (`A`, `B`, or `C`) and two numbers. For those in the know, each one of these tweets represents hidden dealings between major organizations. TheBuzz would assign n organizations a distinct integer label from 1 to n. The single character at the start of each message would indicate the type of relationship between organizations x and y:

- `A` x y – There is an alliance between organizations x and y
- `B` x y – There is a battle brewing between organizations x and y
- `C` x y – Organizations x and y are mutually considering to combine together

You can assume that TheBuzz's information is complete. That is, there is no relationship between organizations outside of what they have already tweeted.

As a hacktivist yourself, you maintain the political blog *Blanc Space* and have dealt with these organizations before. You know the names of all the organizations and have personal records on the relationships between them. You were not told which label that TheBuzz had given to each organization, but maybe you can *deduce* it. You realize that you can use TheBuzz's information to cross-reference and verify the correctness of both your information sources.

Given your own personal records, is it possible to match each organization name with an assigned number from TheBuzz? And if there's a match, what is it?

Input

The first line of input contains two space-separated integers n and r, the number of organizations and the number of relationships between organizations (as observed by you and the TheBuzz).

Then n lines follow, each containing a single string o_i describing the name of an organization.

Then r lines follow, each containing a character followed by two strings, separated by single spaces, representing your own personal records:

- p_i – the relationship between the organizations
- x_i – the name of the first organization in this relationship
- y_i – the name of the second organization in this relationship

Here p_i can either be `A`, `B`, or `C`, their definitions described above. Note that all relationship types are symmetric, so x_i and y_i are interchangeable.

Finally, another r lines follow, representing TheBuzz tweets, in a similar format p_j a_j b_j where a_j and b_j are integers representing two organization IDs.

Output

If there is no possible way to match your personal records with TheBuzz tweets, output the string `IMPOSSIBLE`.

If there is more than one possible way to match records, output the string `TOO MANY`.

Otherwise output n lines, describing a proper assignment of organization IDs to names. The first line should contain the name of the organization referred by ID 1, the second line should contain the name of the organization referred to by ID 2, and so on.

Constraints

- $2 \leq n \leq 10$
- $1 \leq r \leq n(n-1)/2$
- Each organization name contains at least 1 and at most 10 lowercase English letters.
- $1 \leq a_i, b_i \leq n$
- $x_i \neq y_i$
- $a_i \neq b_i$
- p_i is one of: `A`, `B`, or `C`.
- The relationship between two organizations will appear at most once within your personal records, and also at most once within TheBuzz's tweets.

Sample Input 1

```
3 3
chrysler
ford
gm
A chrysler ford
B ford gm
C chrysler gm
A 1 3
C 2 3
B 2 1
```

Sample Output 1

```
ford
gm
chrysler
```

Sample Input 2

```
4 4
demetrius
hermia
helena
lysander
A lysander hermia
B helena demetrius
B demetrius hermia
C lysander helena
B 1 2
C 1 4
A 3 4
B 1 3
```

Sample Output 2

```
IMPOSSIBLE
```

Sample Input 3

```
2 1
good
evil
B good evil
B 1 2
```

Sample Output 3

```
TOO MANY
```