Spring 2021 mBIT Advanced Division

June 12, 2021

These problems are roughly ordered by difficulty. However, you should read and think about as many problems as you can in the time given. Good luck and happy coding!

Contents

1 Pokémon Permutation 4

2 Azran Tablets 5

3 Goomba Grouping 6

4 Cytus Craze 8

5 Knockout Tournament 9

6 Squid Art 10

7 Scribble Roads 12

8 Future Gadget Lab 13

9 Immortality Potion 14

10 Luigi's Mansion 16

11 Kirby's Buffet 18

12 Pillar Path 20

Program Specifications

The memory limit for every problem is 256 MB, unless otherwise specified.

Time Limits (seconds)			
Problem	**C++**	**Java**	**Python**
Pokémon Permutation	1	1	1
Azran Tablets	1	1	1
Goomba Grouping	1	1	1
Cytus Craze	1	1	1
Knockout Tournament	1	2	2
Squid Art	1	2	2
Scribble Roads	1	1	1
Future Gadget Lab	1	1	1
Immortality Potion	1	1	1
Luigi's Mansion	1	3	5
Kirby's Buffet	3	4	5
Pillar Path	5	5	5

Advice

Look at the pretests. You can access the first four pretests for each problem once you've made a submission. Some of the pretests are reduced in size to help you debug your program. Keep in mind that your final submission will be judged on a separate set of 40 hidden system tests for the official rankings.

Understand the new scoring system. Your program will only be submitted to the 40 system tests once it passes all 10 pretests. You will get one point for each system test you pass, **plus 20 points** if you get all of them correct (for a maximum of 60 points per problem). Results of the system tests will not be released until the end of the contest. Unlike last year, you will not get any points for programs that do not pass all pretests. Ties will be broken by the time of the last submission of a program which passes pretests.

Watch out for integer overflow. When a problem uses large values, make sure you use `long` (in Java) or `long long` (in C++). Python integers cannot overflow.

Use fast I/O. For problems with large input sizes, you may want to use faster I/O methods to prevent a time limit error. Here is how to use fast I/O in each language:

- In Python, write `from sys import stdin, stdout` at the top of your program. When reading input, use `stdin.readline()`. To write output, use `stdout.write()`.
- In Java, use a custom Scanner class as shown here.
- In C++, write `ios_base::sync_with_stdio(false); cin.tie(NULL);` at the top of your `main` method. Then you can use `cin` and `cout` as usual. Printing a single newline character (`\n`) is faster than `endl`.

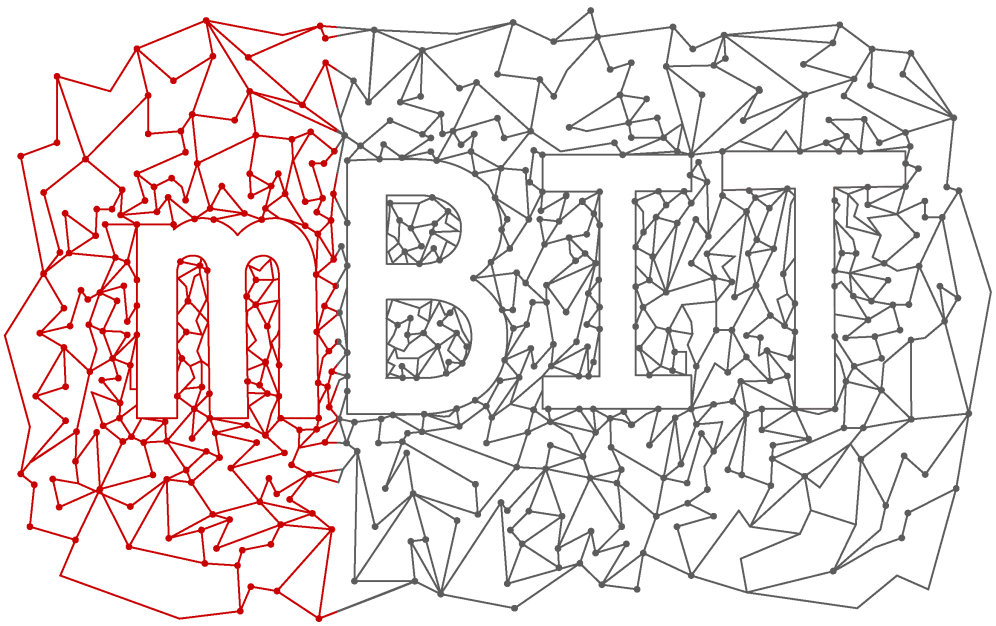
Print extra digits for non-integer values in C++. If you are printing a `double` value in C++, by default it will only output a few digits (which may result in a wrong answer from our grader). To output real values with more precision, write `cout << setprecision(16);` at the top of your program. Our grader will accept real values in

fixed **or** scientific notation (so `1234.56789`, `1.23456789E3`, and `1.23456789e+003` are treated the same). There will always be a tolerance for small relative errors between your solution and the correct answer.

Special considerations for Python. Make sure you're using fast I/O methods as described above. You can increase the recursion limit if your functions use repeated recursion. Additionally, we strongly recommend submitting your solutions in **PyPy**, which is typically faster than Python. Make sure you strip your input of trailing white space (this is especially important for our grader). Finally, `exit()` and `quit()` do not work with our grader.

Language versions. Our grader uses C++17 (compiled with the `-O3` tag), Java 11, and Python 3.9. Our version of PyPy implements Python 3.7.

Ask for clarifications! If you are confused about a problem statement, do not hesitate to message us.



THERE ARE THREE IMPOSTORS AMONG US.

See if you can find the hidden *Among Us* references in this contest! All three references are contained in problems shared between the Standard and Advanced divisions. Message us if you have found them all to receive a special shout-out!

§1 Pokémon Permutation

The insect Stentorceps weedlei *was named after the hairy bug Pokémon Weedle.*

Ash ran into a Pokémon on his way to Viridian City. Missing all of his Poké Balls, he had no way to capture the creature. Only the sounds it made before it escaped were recorded. Normally, Ash would be able to recognize the Pokémon by its appearance, but this species didn't appear in his Pokédex. Given its sounds, Ash hopes to identify the unknown species in the research lab at Viridian City. Unfortunately, his notes got all scrambled up in his rush to get to the lab. Specifically, he only has a permutation (a reordering) of the original string that he recorded.

As with most Pokémon, Ash knows that the only sound the unknown creature could have made was a repetition of its name two or more times. For example, if he had seen a Ditto, Ash could have recorded `dittoditto` or `dittodittoditto`. Can you help Ash identify the Pokémon by finding one of the possibilities of the original periodic string? It is possible that his notes may be wrong – if no answer exists, tell him so.

Input Format:

The first and only line contains a string of lowercase English letters. The length of the string is between 1 and 10^5 characters, inclusive.

Output Format:

Output a periodic permutation of the string. If no such string exists, output `IMPOSSIBLE`.

If there are multiple correct answers, you may print any.

Sample Input:

```
atatatraartttaarttata
```

Sample Output:

```
rattatarattatarattata
```

A valid possibility for the original periodic string is `rattatarattatarattata`. Note that this is not the only answer, for example `aaatttraaatttraaatttr` would also work.

§2 Azran Tablets

The Professor Layton *game series revolves around solving various puzzles to progress through the story. The newest game has over 500 puzzles to tackle!*

Professor Layton and his assistant Luke have uncovered ancient Azran ruins, but the secrets inside are protected by a devious puzzle. In front of the entrance there is a row of N tablets, each inscribed with a lowercase letter. Altogether, the sequence of tablets can be represented by a string s of length N.

Layton and Luke must knock over all of the tablets by repeatedly performing an Azran ritual dance. To initiate each dance, Layton chooses some tablet L and Luke chooses some tablet R such that both tablets are still standing and have the same letter ($s_L = s_R$). Starting at their respective tablets, they then travel towards each other in a series of complicated dips and twirls. All tablets between L and R, inclusive, get knocked down during the performance (if any of these tablets had already been toppled, they remain down). Note that L is allowed to be equal to R, in which case they knock down a single tablet.

If Layton and Luke work optimally, what is the minimum number of times they must execute this dance to knock down all of the tablets?

Input Format:

The first and only line contains a string s consisting of lowercase English letters. The length of s is between 1 and 10^5, inclusive.

Output Format:

Output one line with the minimum number of dances needed to knock down all of the tablets.

Sample Input:

```
azrantablets
```

Sample Output:

```
4
```

Layton and Luke can knock down all the tablets with 4 executions of the dance in the following manner: `azrantablets` → `azra_tablets` → `azra______s` → `azra_______` → `__________`. It can be proven that this is optimal.

§3 Goomba Grouping

The famous character Mario first appeared in Donkey Kong, *where he had to rescue his girlfriend Pauline. Since then, he has appeared in more than 200 different video games.*

Bowser is coming up with a plan to stop his archenemy Mario. He commands an army of N Goombas of varying weights. To guard both entrances of his castle, he will split his army into two squadrons (each Goomba must be in exactly one squadron). Since bigger always means better, the strength of a squadron is equal to the sum of the weights of its Goombas. Bowser wants his two squadron strengths to be as close as possible to present a balanced defense.

Bowser comes up with the following greedy algorithm to split up his army. He first initializes two empty squadrons. He then iterates through his Goombas one by one in **decreasing** order of weight, each time adding it to the squadron that currently has the smaller total strength.

Bowser Jr. knows that his dad's solution will not always achieve the optimal splitting. To prove his dad wrong, Bowser Jr. wants to come up with a set of N Goomba weights for which an optimal splitting has squadrons with a strength difference of K, but his dad's algorithm results in a difference of more than K. Can you help Bowser Jr. construct such a set?

Your program will have to answer multiple test cases.

Input Format:

The first line contains T, the number of test cases ($1 \leq T \leq 100$).

The only line of each test case contains N and K ($1 \leq N \leq 20$; $0 \leq K \leq 10^9$). It is guaranteed that the sum of N over all T test cases will not exceed 100.

Output Format:

Output T lines. For each test case, if an answer exists, output N integers. Each integer must be between 1 and 10^{18}, inclusive. If no answer exists, output `-1`.

If there are multiple correct answers, you may print any.

Sample Input:

```
2
8 1
1 10
```

Sample Output:

```
9 13 16 15 19 20 15 18
-1
```

For the first test case, Bowser will take the following steps to build his squadron:

```
A = [], B = []
A = [20], B = []
A = [20], B = [19]
A = [20], B = [19, 18]
A = [20, 16], B = [19, 18]
A = [20, 16, 15], B = [19, 18]
A = [20, 16, 15], B = [19, 18, 15]
A = [20, 16, 15, 13], B = [19, 18, 15]
A = [20, 16, 15, 13], B = [19, 18, 15, 9]
```

The strength of squadron A is 64 and the strength of squadron B is 61, so Bowser's difference is 3. However, we can achieve a better answer of 1 with the following splitting:

```
A = [20, 15, 15, 13], B = [19, 18, 16, 9]
```

The strength of squadron A is 63 and the strength of squadron B is 62, so the difference is 1, which is indeed K (and less than Bowser's difference of 3).

For the second test case, Bowser's algorithm always achieves the optimal difference for all possible sets of size 1.

§4 Cytus Craze

Cytus *is a rhythm game in which the user must tap circles while following the music. There are more than 500 playable songs between* Cytus *and* Cytus II.

NEKO#ϕωϕ (or just NEKO, for short) is playing a game of *Cytus*, in which she taps beats along with the rhythm of a song to earn points. She is currently on one of ROBO_Head's fiendishly difficult songs, which has a very strange scoring system. There are N beats in the song, numbered $1, \ldots, N$, and her score starts at 0. She can either hit or miss each beat. If she hits the i-th beat her score is XORed with i, but if she misses the beat her score is instead XORed with $(i-1)$. For example, if $N = 4$ and she only hits the first and fourth beats, her final score would be $1 \oplus (2-1) \oplus (3-1) \oplus 4 = 6$. Here, $\oplus$ denotes the bitwise XOR operation.

ROBO_Head will only give NEKO the coveted Million Master title if she achieves a final score greater than K. Distressed, NEKO has asked you to figure out how many ways she could fail. In other words, how many of the 2^N ways to hit or miss the beats result in a score less than or equal to K? Since the answer could be very large, compute it modulo $10^9 + 7$.

Input Format:

The first and only line contains N and K ($1 \leq N, K \leq 10^{18}$).

Output Format:

Output the number of ways NEKO could end with a score of at most K, modulo $10^9 + 7$.

Sample Input:

```
3 2
```

Sample Output:

```
6
```

There are $2^3 = 8$ possibilities of hitting and missing each beat.

Miss,	Miss,	Miss:	$0 \oplus 1 \oplus 2 = 3$
Miss,	Miss,	Hit:	$0 \oplus 1 \oplus 3 = 2$
Miss,	Hit,	Miss:	$0 \oplus 2 \oplus 2 = 0$
Miss,	Hit,	Hit:	$0 \oplus 2 \oplus 3 = 1$
Hit,	Miss,	Miss:	$1 \oplus 1 \oplus 2 = 2$
Hit,	Miss,	Hit:	$1 \oplus 1 \oplus 3 = 3$
Hit,	Hit,	Miss:	$1 \oplus 2 \oplus 2 = 1$
Hit,	Hit,	Hit:	$1 \oplus 2 \oplus 3 = 0$

Exactly 6 of these outcomes produce scores less than or equal to 2.

§5 Knockout Tournament

Street Fighter *started off as a simple arcade game, but since then it has become one of the most popular fighting games ever.*

There are 2^N martial artists gathered for the *Street Fighter V* championship. Each contestant has a strength level a_i. In a match between two contestants, the fighter with a higher strength always wins. If the two fighters have the same strength level, a winner is randomly chosen.

The championship is structured as an N-round single-elimination tournament. In each round, all remaining contestants are randomly paired up into matches (all possible pairings are equally likely). The loser of each match then drops out of the tournament. In this way, the number of contestants halves after each round.

The *unfairness* of a match between a fighter of strength x and a fighter of strength y is $|x - y|$. The total unfairness is the sum of the unfairness of all $2^N - 1$ matches. What is the expected value of the total unfairness of this street fighter tournament? Output your answer as a rational number modulo $10^9 + 7$ (see below).

Input Format:

The first line contains N ($1 \leq N \leq 17$).

The second line contains 2^N integers $a_1, \ldots, a_{2^N}$ ($0 \leq a_i \leq 10^9$).

Output Format:

It can be proven that you can express the expected total unfairness as a rational number p/q, where q is not divisible by $10^9 + 7$. Output pq^{-1} modulo $10^9 + 7$, where q^{-1} represents the modular inverse of q.

Sample Input:

```
2
2 3 5 1
```

Sample Output:

```
666666678
```

It can be shown that the expected unfairness of a random tournament is $\frac{20}{3}$, which is equivalent to 666666678 modulo $10^9 + 7$.

§6 Squid Art

Callie and Marie, the squid sisters in Splatoon, *are named after the squid dish calamari.*

Callie is playing a game of *Splatoon* on an $N \times M$ grid of hexagons, where each cell is either red or blue. She wants to paint the entire grid blue using a special bucket of paint. With a single use of said bucket, she can turn any contiguous component of blue hexagons red, or any contiguous component of red hexagons blue (this is a similar behavior to the paint bucket tool in most image editing programs). What is the minimum number of times Callie has to use the bucket to turn the **entire grid blue**?

Input Format:

The first line contains N and M ($1 \leq N \cdot M \leq 10^5$).

The next N lines each contain a binary string of length M. Red and blue cells are denoted with 0s and 1s, respectively. The hexagonal grid is arranged such that the first cell in the top row is adjacent to the first two cells in the second row (see the diagram).

Output Format:

Output one line with the minimum number of times Callie has to use the paint bucket to turn the entire grid blue.

Sample Input:

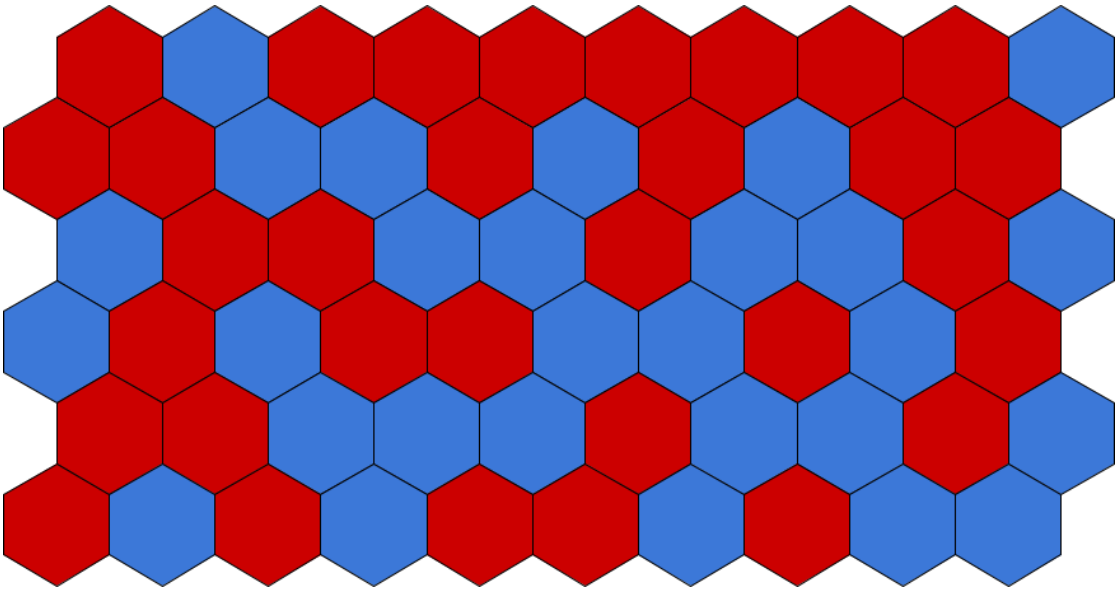
```
6 10
0100000001
0011010100
1001101101
1010011010
0011101101
0101001011
```

Sample Output:

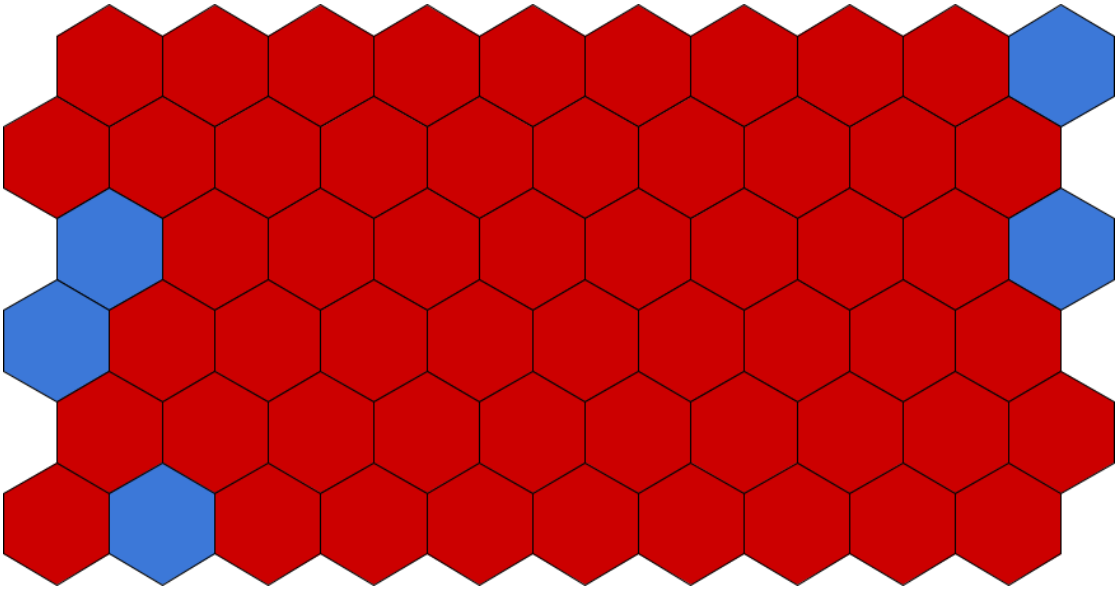
```
2
```

See the diagrams on the following page.

The initial coloring of the grid is shown here:



Callie can first paint the large blue component red. After she does so, the grid will look like this:



She can then paint the remaining red component blue. This process will turn every hexagon blue in 2 moves, which we can show is optimal.

§7 Scribble Roads

General Tanya is a recurring mBIT character, first appearing in a problem from the 2019 contest called Secret Base, *and later in the 2020 fall problem* Tanya's Revenge.

General Tanya has found herself trapped in the world of *Scribblenauts*! Her only way to escape is to beat Maxwell in a game of "Connect the Graph." The game starts with N cities (numbered $1, \ldots, N$) and M undirected roads among them. Tanya and Maxwell will take turns using magical notebooks to draw roads between cities, with Tanya going first. On a player's turn, they must draw a road between two distinct cities that do not already have a road between them. The winner is the first player to end their turn with the entire city graph being connected — that is, for every pair of cities there is a sequence of roads connecting them. By convention, if the cities are already connected before Tanya makes her first move, she loses.

If both Tanya and Maxwell play optimally, can you determine whether Tanya will win or lose? Your program will have to answer multiple test cases.

Input Format:

The first line contains T, the number of test cases ($1 \leq T \leq 100$).

The first line of each test case starts with N and M ($1 \leq N \leq 10^5$; $0 \leq M \leq \min(\binom{N}{2}, 2 \cdot 10^5)$). It is guaranteed that the sum of N over all T test cases does not exceed 10^5, and the sum of M over all test cases does not exceed $2 \cdot 10^5$.

The next M lines of each test describe the roads that exist at the start of the game. There is at most one road between any two cities, and there is no road from a city to itself.

Output Format:

Output T lines containing `WIN` or `LOSE`, with each line corrresponding to whether Tanya will win or lose on that test case.

Sample Input:

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

Sample Output:

```
WIN
LOSE
```

In the first case, Tanya can draw the road between city 1 and city 2 to win.

In the second game, it can be shown that no matter what Tanya does on her first turn, Maxwell can win the game on the second turn. Therefore, Tanya will lose.

§8 Future Gadget Lab

Steins;Gate *is a visual novel game that later got adapted into an anime. It follows the story of a scientist who creates a machine that can send messages into the past to change the present.*

The mad scientist Hyounin Kyouma is traveling through space-time with his microwave time machine! Hyounin's universe consists of N independent timelines numbered $1, \ldots, N$. Through his intense experiments, he has discovered that the timelines are linked together in the form of a tree rooted at timeline 1. As such, each timeline has zero or more children in the timeline tree, and no timeline is the child of more than one parent timeline.

Hyounin's time machine acts in a peculiar manner. When he uses the machine in some timeline u, he gets transported to one of the children of u uniformly at random (all children are equally likely). If u has no children, Hyounin will instead jump all the way back to the timeline 1. The *depth* of a timeline is its distance from timeline 1 in the timeline tree (the depth of timeline 1 is 0).

Hyounin begins his journey in timeline 1, then activates his machine's autopilot feature before taking a nap. As he sleeps, his machine executes J^K time jumps — an extraordinarily large number (see bounds)! When he wakes from his nap, what is the expected value of the depth of his current timeline?

Input Format:

The first line contains N, J, and K ($2 \leq N \leq 10^5$; $2 \leq J \leq 10^9$; $10^5 \leq K \leq 10^9$).

The next $N - 1$ lines describe the edges of the timeline tree. Each line contains two integers u and v, denoting that v is a child of u ($1 \leq u, v \leq N$).

Output Format:

Output Hyounin's expected depth after using his time machine J^K times. Your answer will be accepted if its absolute or relative error is less than 10^{-6}.

Sample Input:

```
4 2 100000
1 2
2 3
1 4
```

Sample Output:

```
0.8
```

It can be shown that the expected depth of the timeline Hyounin wakes up in is approximately $\frac{4}{5} = 0.8$.

§9 Immortality Potion

The character Zelda from The Legend of Zelda *is named after the famous author Scott Fitzgerald's wife.*

Before his great battle with Ganon, Link wants to brew the legendary Immortality Potion. He has determined that this potion can be formed by mixing three liquid chemicals a, b, and c in the ratio $A : B : C$. In other words, the potion must contain A parts of chemical a for every B parts of b and C parts of c. Link has an unlimited supply of all three pure chemicals, as well as three empty cups labeled 1, 2, and 3. The cups have volumes of 1, 2, and 3 liters respectively, but have no gradations along their sides. Link has no other way to measure volume. As a result, he can only perform the following actions with his cups:

- Fill some cup x all the way to the top with some pure chemical y.
- Pour as much of the contents of some cup x into a different cup y as he can (stopping only when cup x becomes empty or cup y becomes full).
- Empty a cup x of all its contents.

Note that, under these actions, the volume of liquid in any cup will always be an integer. You may assume that whenever Link pours one liquid into another, the resultant liquid immediately becomes thoroughly and uniformly mixed.

Link wants to perform some sequence of these actions that results in a cup containing chemicals in the desired ratio (it may have any positive volume). He will then use one final action to drink this cup and, hopefully, gain immortality. Since Link is in a bit of a rush, he only has time to perform at most 300 actions, including the final action of drinking a cup. Can you help him? Your solution will be accepted if the concentration of each chemical is within 10^{-6} of the desired value. It can be shown that a solution always exists.

Input Format:

The first and only line contains three integers A, B, and C ($0 \leq A, B, C \leq 10^6$; $A + B + C \geq 1$).

Output Format:

On the first line output N, the number of moves your solution uses ($2 \leq N \leq 300$).

On each of the next N lines, output one of the following:

- `F` x y to fill cup x with chemical y (x must be $1, 2$, or 3; y must be `a`, `b`, or `c`).
- `P` x y to pour the contents of cup x into cup y (x and y must be $1, 2$, or 3).
- `E` x to empty out the contents of cup x (x must be $1, 2$, or 3).
- `!` x drink the contents of cup x. You must use this as your last move.

Sample Input:

```
0 4 2
```

Sample Output:

```
5
F 1 c
F 2 b
P 1 3
P 2 3
! 3
```

Link needs a final ratio of $0 : 4 : 2$, which is equivalent to $0 : 2 : 1$. He first fills cup 1 with 1 liter of chemical c and cup 2 with 2 liters of chemical b. He then pours both cups into cup 3 and drinks the result.

§10 Luigi's Mansion

In the game Luigi's Mansion, *Luigi has to explore a haunted mansion to rescue Mario, who has been captured by the ghosts.*

The memory limit for this problem is 512 MB.

Luigi is on a quest to defeat the evil King Boo and rescue his brother Mario! King Boo is hiding from Luigi in some room of his vast mansion, which consists of of three long hallways radiating from a central foyer. Each hallway is made from N rooms strung together end-to-end. Formally, each room can be labelled with a pair (i, j) for $1 \le i \le N$ and $1 \le j \le 3$. Room (i, j) is adjacent to rooms $(i-1, j)$ and $(i+1, j)$, except for the rooms (N, j) (which are only adjacent to $(N-1, j)$) and rooms $(1, j)$ (which are adjacent to the foyer instead of $(0, j)$). It may help to think of the mansion as a star graph with $3N + 1$ nodes.

Although Luigi isn't sure where King Boo is hiding, he knows the probability that he is in any given room. Specifically, he knows that King Boo is hiding in room (i, j) with probability a_i/D. Note that this probability is independent of j, so King Boo never has a preference of one hallway over another. It is guaranteed that $3 \cdot (a_1 + \cdots + a_N) = D$.

Luigi is currently in the foyer. Since the doors are heavy, it takes Luigi one minute to move from a room to an adjacent room, including the foyer. Luigi will use an optimal search route to minimize the expected value of the time it takes for him to find King Boo. What is this minimum expected time?

Input Format:

The first line contains N and D ($1 \le N \le 5000$; $3 \le D \le 1.5 \cdot 10^{10}$; D is divisible by 3).

The next line contains N nonnegative integers $a_1, \ldots, a_N$ ($0 \le a_i \le 10^6$). It is guaranteed that $3 \cdot (a_1 + \cdots + a_N) = D$.

Output Format:

The expected number of minutes it will take Luigi to find King Boo can be written in the form $\frac{x}{y}$, for relatively prime positive integers x and y (y may be 1). Output a single line containing `x/y`.

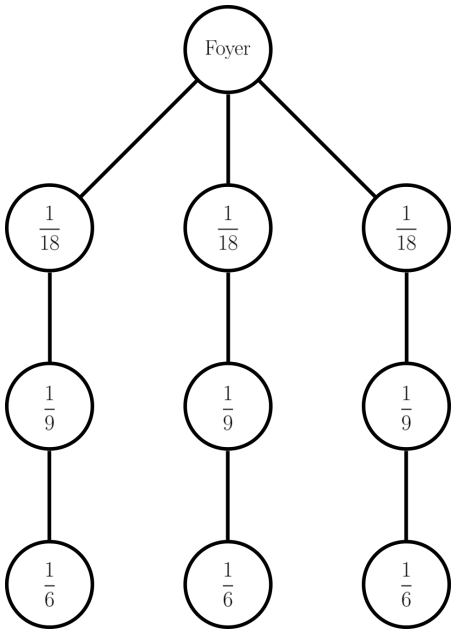
Sample Input:

```
3 18
1 2 3
```

Sample Output:

```
25/3
```

The structure of the mansion is shown here:



As displayed in the diagram, the probability that King Boo is in each of the three rooms closest to the foyer is $\frac{1}{18}$, the probability that he is in each of the three rooms a distance of 2 from the foyer is $\frac{1}{9}$, and the probability that he is in each of the three farthest rooms is $\frac{1}{6}$. It can be shown that with the optimal searching route, the expected number of minutes it takes Luigi to find King Boo is $\frac{25}{3}$.

§11 Kirby's Buffet

Kirby's character design was originally a placeholder for a future protagonist, but it was later decided that his simplistic form fit the game design well.

Kirby has just defeated King Dedede and saved Dreamland! To celebrate, the chefs have created a fantastic buffet consisting of N dishes in a row numbered $1, \ldots, N$. The i-th dish has a flavor of a_i. Kirby's taste buds behave in a strange manner: if he eats a non-empty subset of dishes, the food combines in his mouth to yield a flavor equal to the bitwise AND of the original flavors.

The chefs have Q potential table settings they can use. The i-th setting allows Kirby to reach all dishes with indices from L_i to R_i, inclusive. The chefs also have an ideal flavor X_i associated with each setting that they believe will provide the best culinary experience for Kirby. For each table setting, can you determine whether there is a non-empty subset of dishes reachable by Kirby that has a bitwise AND of X_i?

Input Format:

The first line contains N and Q ($1 \le N, Q \le 2 \cdot 10^5$).

The next line contains N integers $a_1, \ldots, a_N$ ($0 \le a_i < 2^{17}$).

The next Q lines describe table setting queries. Each line contains three integers L_i, R_i, and X_i ($1 \le L_i \le R_i \le N$; $0 \le X_i < 2^{17}$).

Output Format:

Output Q lines containing `YES` or `NO` depending on whether there exists such a subset.

Sample Input:

```
5 5
7 11 9 13 5
2 5 12
1 5 1
1 3 3
2 4 11
1 5 4
```

Sample Output:

```
NO
YES
YES
YES
NO
```

For the first and fifth table settings, there are no subsets of dishes in the given intervals that have the appropriate AND.

For the second setting, the dishes with flavors 11 and 5 AND to 1.

For the third setting, the dishes with flavors 11 and 7 AND to 3.

For the fourth setting, there is a dish with the flavor 11 by itself.

§12 Pillar Path

Genshin Impact *is a popular gacha game that came out in 2020. It is estimated that players have collectively spent more than $15 million US dollars to acquire Zhongli.*

The memory limit for this problem is 512 MB. Python/PyPy and Java solutions are unlikely to run within the time limit.

The powerful dragon Azhdaha is locked in a epic battle with the great god Zhongli! Azhdaha is currently at the location (s_x, s_y) on the coordinate plane, and Zhongli is at (e_x, e_y). Azhdaha is about to charge forward to pounce on his opponent.

To slow down Azhdaha's murderous advance, Zhongli has erected N circular pillars, with the i-th pillar represented by a circle centered at (x_i, y_i) with radius r_i. Each pillar also has a durability d_i. No two pillars intersect or contain each other.

To reach Zhongli, Azhdaha will first use his fiery breath to destroy zero or more pillars. He will then take some path from (s_x, s_y) to (e_x, e_y) that doesn't cross into the interior of any remaining pillars. This path may be curved or straight, as long as it stays within the two dimensions of the coordinate plane. The total energy Azhdaha expends is equal to the sum of the durabilities of the pillars he destroys, plus the length of his path. Can you calculate the minimum energy Azhdaha must use to reach Zhongli? Your answer will be judged correct if its absolute or relative error is less than 10^{-6}.

Input Format:

The first line contains N ($1 \leq N \leq 700$).

The next line contains four integers s_x, s_y, e_x, and e_y, the coordinates of the starting and ending locations ($0 \leq s_x, s_y, e_x, e_y \leq 10^6$; $(s_x, s_y) \neq (e_x, e_y)$).

The next N lines each contain four integers x_i, y_i, r_i, and d_i, describing the center, radius, and durability of each pillar ($0 \leq x_i, y_i \leq 10^6$; $1 \leq r_i \leq 10^6$; $1 \leq d_i \leq 10^6$). It is guaranteed that no two circles intersect or contain each other. The starting and ending points do not lie on or inside any circle.

Output Format:

Output the minimum energy Azhdaha must expend to get from (s_x, s_y) to (e_x, e_y).

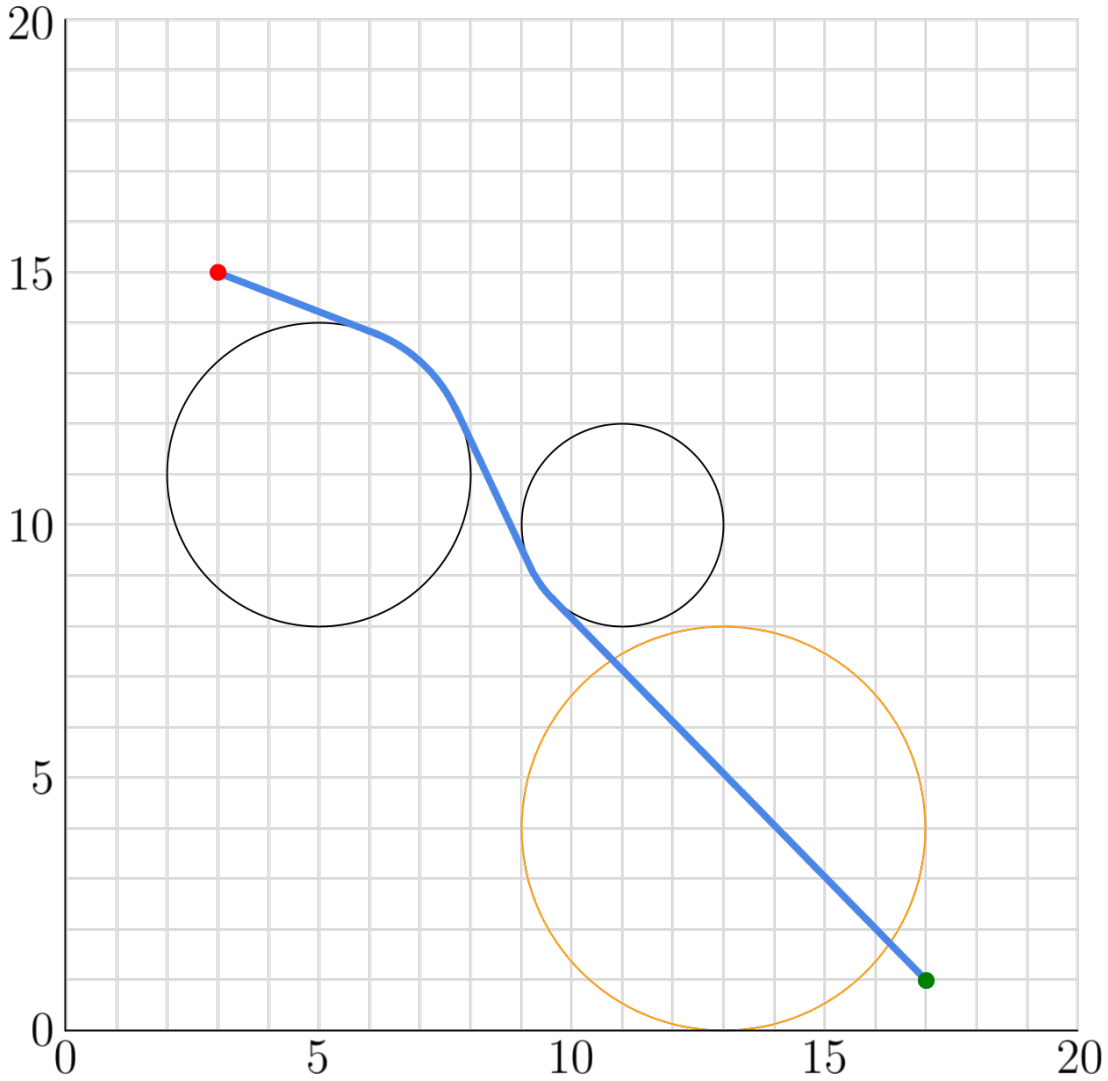
Sample Input:

```
3
3 15 17 1
5 11 3 5
11 10 2 4
13 4 4 1
```

Sample Output:

```
21.3519917504
```

Here is the diagram for this configuration:



The shortest path is shown in blue. Note that Azhdaha destroys the third pillar.