Problem A. Alluka's Curse

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

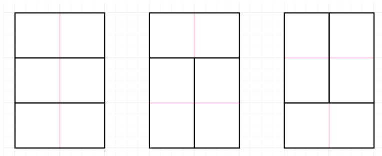
`Alluka` is possessed by a mysterious creature known as `Nanika`.

`Nanika` has a very powerful ability: she can make anything come true once someone asks her.

Of course magic does not exists. `Nanika` has a secret company that helps her fulfilling the wishes. People very talented work at her command and everything can become possible under talented hands.

You're very talented and work for `Nanika`, so you have to help her fulfill the wishes as soon as she asks you.

The last wish, as weird as it might sound, is to know how many distinct ways you can fill a $3 \times n$ grid using only 1×2 pieces.



As the answer can be very large, output it modulo $10^9 + 7$.

Input

The first line of input contains one integer, n ($1 \leq n \leq 10^7$) — the dimension of the grid.

Output

Print how many distinct ways you can fill the grid.

Examples

standard input	standard output
2	3
3	0
4	11

Note

Image in statement represents test case #1.

Problem B. Battle Sequence

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

You are one of the 1000 lucky people invited to test *Sword Art Online* closed beta.

The game is very unique and extremely beautiful. But it's still a game.

While battling *Illfang, the Kobold Lord*, you recognize some attack patterns:

- *Illfang* only have two attacks, `Chop` and `Slice`.
- If *Illfang* uses the same attack 3 times in a row, just before the third attack it stops to charge.
- If *Illfang* is attacked during it's charge, it does not continue that attack.

Knowing these patterns you came up with an strategy to defeat it:

- Every time *Illfang* uses a `Chop` attack, you will defend with a `Block`.
- Every time *Illfang* uses a `Slice` attack, you will `Dash`.
- If *Illfang* begins to charge a `Chop`, you will attack with `Plunge`.
- If *Illfang* begins to charge a `Slice`, you will attack with `Tackle`.

Your guild gathered information about the attack sequence of `Illfang` and your job is to determine your attack sequence to defeat it.

Input

The only line of input contains one string s ($|s| \leq 100$) — *Illfang*'s attack sequence. `'C'` means *Illfang* will use a `Chop` attack. `'S'` means *Illfang* will use a `Slice` attack.

Output

Print one string — your attack sequence to defeat *Illfang*. `'B'` means you will use a `Block`. `'D'` means you will use a `Dash`. `'P'` means you will use a `Plunge`. `'T'` means you will use a `Tackle`.

Examples

standard input	standard output
CSCCCS	BDBBPD
CCSSSSSSC	BBDDTDDTB

Problem C. Chimera Ant King

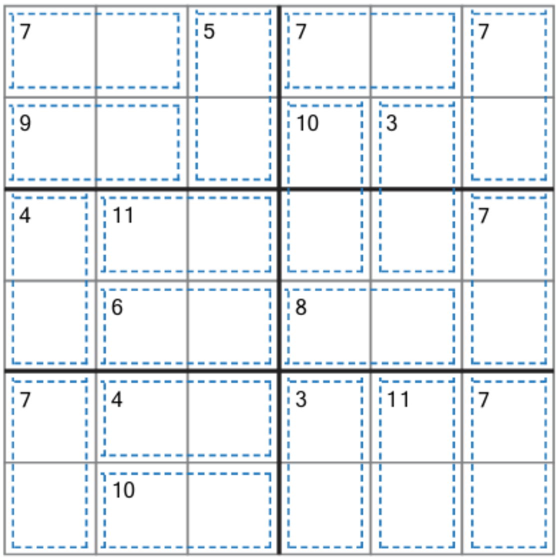
Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

`Meruem`, the King of *Chimera Ants*, decided to be the best player of every game in the world.

Of course some games are played by a single player, so it's not easy to decide which player is the best at it. The criteria `Meruem` decided to use was how much time the player could finish the game, be lower the better.

You are to compete against `Meruem` on a very unique game: *Sudoku Killer*. He thinks you are the best player in the world, but just in case he is mistaken you are going to play the child's version.

Child Sudoku Killer is played on a 6×6 grid, divided in six 2×3 disjoint blocks. Also, the grid is divided into disjoint regions, each being a connected and have a value. Every grid cell is contained by exactly one block and one region.



To finish the game, the following rules must meet:

- Each grid cell must have exactly one value between 1 and 6, inclusive.
- Each lines, column, block and region's cells must have different values.
- The sum of region cell's values must be equal to the region value.

You must finish the game as fast as possible. Show `Meruem` the solution.

Input

The first line contains one integer, n ($6 \le n \le 36$) — the number of regions.

The second line contains n integers, v_i ($1 \le v_i \le 21$) — the value of the i-th region.

The next 6 lines contains, each, 6 integers, r_{ij} ($1 \le r_{ij} \le n$) — the region of the cell in i-th line and j-th column.

It's guaranteed that the given board satisfies all game rules and that it has at least one solution.

Output

Print 6 lines, each having 6 space-separated integers between 1 and 6, inclusive — the solution of the game.

If there are multiple solutions, print any of them.

Examples

standard input	standard output
18 7 5 7 7 9 10 3 4 11 7 6 8 7 4 3 11 7 10 01 01 02 03 03 04 05 05 02 06 07 04 08 09 09 06 07 10 08 11 11 12 12 10 13 14 14 15 16 17 13 18 18 15 16 17	6 1 2 3 4 5 4 5 3 6 1 2 3 6 5 4 2 1 1 2 4 5 3 6 5 3 1 2 6 4 2 4 6 1 5 3
6 21 21 21 21 21 21 01 01 01 01 01 01 02 02 02 02 02 02 03 03 03 03 03 03 04 04 04 04 04 04 05 05 05 05 05 05 06 06 06 06 06 06	1 2 3 4 5 6 4 5 6 1 2 3 2 1 4 3 6 5 3 6 5 2 1 4 5 3 1 6 4 2 6 4 2 5 3 1

Problem D. Doll Collector

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Everybody loves *Pokémon* dolls. They are the climax of every *Pokémon* game.

Ash started collecting some pokédolls and got addicted. Now he wants to spend all of his *Pokédollars* on new pokédolls to have the largest collection as possible.

Since *Pokémon* types grows very quickly, all *Pokémon* ID from 1 to 10^{18} exists and the pokédoll of the *Pokémon* with ID i costs i *Pokédollars*.

You don't know exactly how many *Pokédollars* Ash has, so you will calculate the maximum number of distinct pokédolls he can buy for n amounts of *Pokédollars*.

Input

The first line of input contains one integer, n ($1 \le n \le 10^5$) — the number of queries.

The next n lines contains, each, one integer, s_i ($0 \le s_i \le 10^{18}$) — the amount of *Pokédollars* you estimate Ash has.

Output

For each query, print the maximum number of distinct pokédolls Ash could buy if he had the amount of *Pokédollars* of given query.

Examples

standard input	standard output
3 2 3 9	1 2 3
2 52 60	9 10

Problem E. End Game

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	256 megabytes

The `Vanguard` guild has reached the last *Aincrad* floor: 100.

The guild wants to finish the game as soon as possible to free all the people trapped inside it, but you cannot rush, otherwise people may die fighting the last boss and you may lose the opportunity to finish it.

The `Vanguard` fought the final boss, `Lavos`, a number of times and now you know how it attacks and how should you fight it.

`Lavos` has a cyclic attack pattern p. Each attack is one of the following: *Rain Destruction*, *Lavos Needle* or *Earthquake*. As soon as you enter the boss room it starts it's attack in any position of it's pattern, uniformly distributed. After `Lavos` uses the attack at position i of the pattern, the next attack will be 1, if i is the last attack of the pattern, or $i + 1$, otherwise.

The only chance against this boss is one technique: the *Omnislash*. It consists of an attack sequence t, having three attack variations. Coincidently, each variation counters one of `Lavos` attacks.

Alpha variation counters *Rain Destruction* and causes a damage to `Lavos`. *Beta* variation counters *Lavos Needle* and causes b damage to `Lavos`. *Charlie* variation counters *Earthquake* and causes c damage to `Lavos`.

You can't stop the *Omnislash* attack sequence and *Omnislash* attacks have the same duration as `Lavos` attacks (each `Lavos` attacks happen at the same time as *Omnislash* attacks).

You will use *Omnislash* as soon as `Lavos` starts attacking and you can only use `Omnislash` once. Either `Lavos` dies during *Omnislash* or you die after *Omnislash* finishes. You cannot be killed in the middle of *Omnislash*.

You already know that `Lavos` has a total of h hit points.

The `Vanguard` needs to know what are the chances to defeat `Lavos`.

`Lavos` is considered defeated as soon as it's hit points get to zero or less. It loses 1 hit point for every 1 damage it receives.

Input

The first line contains 4 integers, a, b, c and h ($0 \le a, b, c \le 10^9$, $1 \le h \le 10^9$) — the damage *Alpha*, *Beta* and *Charlie* attack variations causes, and `Lavos` health.

The next line contains one string, p ($1 \le |p| \le 2 \times 10^5$) — `Lavos` attack pattern. `'R'` means *Rain Destruction*, `'N'` means *Lavos Needle* and `'E'` means *Earthquake*.

The next line contains one string, t ($1 \le |t| \le |p|$) — *Omnislash* attack sequence. `'A'` means *Alpha* variation, `'B'` means *Beta* variation and `'C'` means *Charlie* variation.

Output

Print the probability to win against `Lavos` as two space-separated integers, p and q, which p/q is the answer and it's an irreducible fraction.

Examples

standard input	standard output
1 1 1 1 RNENR ABC	3 5
3 1 2 3 RNENER ABC	1 2

Problem F. Frieza Frenzy

Input file: `standard input`
Output file: `standard output`
Time limit: 1 second
Memory limit: 256 megabytes

"I doubt I need an introduction, but just in case, I am the mighty Frieza, and yes, all the horrible stories you've heard are true."

`Frieza` arrived on Earth and is seeking for `Goku`.

He's not so patient, so he decided to destroy roads, one by one, until `Goku` appears.

As `Frieza` destroy roads, some regions get disconnected and people on some places can't get to some other places. A set of places is said to be connected if, from any place on this set, it's possible to reach every other place on it and no place outside it.

The *Central City* has n places and m bidirectional roads that connect two distinct places. Initially the city is connected.

`Goku` doesn't care about `Frieza` at all and will let `Frieza` destroy all roads. Meanwhile, he decided to count the number of connected sets of places in *Central City* every time `Frieza` destroy a road. You don't care about `Frieza` also, so you will help `Goku`.

Input

The first line of input contains two integers, n and m ($2 \le n \le 10^5$, $n-1 \le m \le min(n \times (n-1)/2, 10^5)$ — the number of places and roads at *Central City*.

The next m lines contains, each, two integers, u_i and v_i ($1 \le u_i, v_i \le n$, $u_i \neq v_i$) — the places the i-th road connects.

The next line contains m numbers, a permutation of all numbers between 1 and m — the sequence of roads destroyed by `Frieza`.

It's guaranteed that there's no more than one road that connects u and v, for every u and v.

Output

For each road destroyed, print the number of connected sets of places in *Central City*.

Examples

standard input	standard output
6 6 1 2 1 3 2 3 3 4 4 5 4 6 5 4 2 1 6 3	2 3 3 4 5 6
4 6 1 2 1 3 1 4 2 3 2 4 3 4 3 4 1 5 2 6	1 1 1 2 3 4

Problem G. Gift Swords

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

`Asuna` wants give a gift to `Kirito`. She want to buy swords, but she doesn't know which yet.

Since `Kirito` is a dual-wielder (meaning than he can use two swords at the same time), `Asuna` will buy a pair of swords to him, but dual-wielders have to use compatible swords.

Each sword can be identified as one positive integer id ($id > 0$). Two swords are compatible if they do not have any characteristic in common. The characteristics of a sword are all positive integers c ($c > 1$) which $id = k \times c$, for some integer $k > 0$.

For example, the sword with $id = 15$ has two characteristics: 3 and 5.

The sword shop have n swords in total. As times passes, swords go out of stock and back to stock. `Asuna` wants to know, for each event, how many pairs of swords she can gift `Kirito`.

Input

The first line of input contains two integers, n and t ($1 \leq n, t \leq 2 \times 10^5$) — the number of swords the shop sells and how many events will happen.

The next line contains n space-separated integers, id_i ($1 \leq id_i \leq 5 \times 10^5$) — the identification of each sword. Swords does not necessarily have distinct identifications.

The next t lines contains, each, one integer, x ($1 \leq x \leq n$) — the x-th sword came out of stock or back to stock. If the x-th sword was in stock then it came out of stock, otherwise it came back to stock.

Initially no swords are in stock.

Output

For each event print how many pairs of swords `Asuna` can gift `Kirito`.

Examples

<table>
<tr><th>standard input</th><th>standard output</th></tr>
<tr><td><pre>5 6
1 2 3 4 6
1
2
3
4
5
1</pre></td><td><pre>0
1
3
5
6
2</pre></td></tr>
<tr><td><pre>3 5
1 2 4
1
2
3
2
1</pre></td><td><pre>0
1
2
1
0</pre></td></tr>
</table>

Problem H. Heaven's Arena

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Heaven's Arena is a huge building, said to have infinite floors up and down, that fighters can battle to earn money and have a private bedroom, if they are good enough.

This year, the *Heaven's Arena* changed it rules. Now you start at floor n and fight a total of a battles. Each time you win a battle you go to the next level, and each time you lose you go to the previous level. There's no tie on *Heaven's Arena*.

You know you have a p percent chance to win each battle. What is the expected floor you will end up at after all your battles?

Input

The first line of input contains three integers, n, a and p ($0 \le n, a \le 10^9$, $0 \le p \le 100$) — the floor you start, the number of battles you will fight and the percent chance to win.

Output

Print the expected floor you will end up at after all your battles.

The error between the answer and the printed value must not exceed 10^{-6} in absolute value.

Examples

standard input	standard output
5 2 75	6.000000000
100 7 21	95.940000000

Problem I. Important Equipment

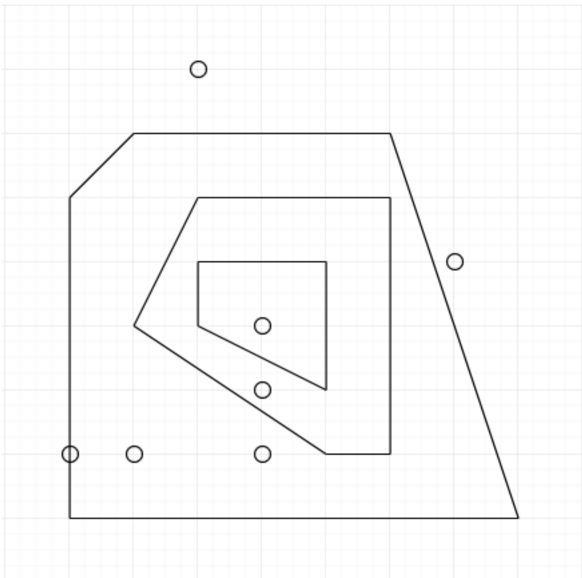
Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Humanity lives inside 3 convex walls. The inner one is *Wall Sheena*, the second one is *Wall Rose* and the outer one is *Wall Maria*.

As a member of *Survey Corps* you are risk your life for humanity. The next mission is to gather some important equipment.

The *Survey Corps* have a list of n equipment. You know the position of every equipment and you need to tell the *Survey Corps* where are each equipment, inside *Wall Sheena*, *Wall Rose*, *Wall Maria* or outside.

A position in the border of a wall is considered inside the wall.



Input

The first line of input contains three integers, s, r and m ($3 \le s, r, m \le 1000$) — the number of points that define *Wall Sheena*, *Wall Rose* and *Wall Maria*, respectively.

The next s lines contains, each, two integers, x_i and y_i ($-10^6 \le x_i, y_i \le 10^6$) — the points that define *Wall Sheena*, in counter-clockwise order. No three points are collinear.

The next r lines contains, each, two integers, x_i and y_i ($-10^6 \le x_i, y_i \le 10^6$) — the points that define *Wall Rose*, in counter-clockwise order. No three points are collinear.

The next m lines contains, each, two integers, x_i and y_i ($-10^6 \le x_i, y_i \le 10^6$) — the points that define *Wall Maria*, in counter-clockwise order. No three points are collinear.

The next line contains one integer, n ($1 \le n \le 1000$) — the number of important equipment.

The next n lines contains, each, two integers, x_i and y_i ($-10^6 \le x_i, y_i \le 10^6$) — the position of i-th equipment.

It's guaranteed that all walls are convex, that *Wall Sheena* is strictly inside *Wall Rose* and that *Wall Rose* is strictly inside *Wall Maria*.

Output

For each equipment, print `"Sheena"`, `"Rose"`, `"Maria"` or `"Outside"`, meaning that the equipment is inside the wall printed and not inside any inner one, or outside of all walls.

Examples

standard input	standard output
4 4 4 -1 -1 1 -1 1 1 -1 1 -2 -2 2 -2 2 2 -2 2 -3 -3 3 -3 3 3 -3 3 5 0 0 1 0 2 0 3 0 4 0	Sheena Sheena Rose Maria Outside
4 5 5 2 4 2 3 4 2 4 4 5 1 5 5 2 5 1 3 4 1 0 0 7 0 5 6 1 6 0 5 7 2 7 1 1 6 4 3 1 3 3 0 1 3 2	Outside Maria Outside Maria Sheena Maria Rose

Note

Image on statement represents test case #2.

Problem J. Jaeger Training

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

To become a *Survey Corps* people need to first graduate on *Training Corps*.

It's not an easy task, the outside is full of huge fearless titans, so the *Training Corps* needs to assert only the best cadets go through.

During one of the exams, n cadets must traverse a circuit and score some points. The grade the i-th cadet earns is n minus the number of cadets that finished the circuit before the him and scored at least the same amount of points.

You must calculate the grade each cadet earned during the exam.

Input

The first line of input contains one integer, n ($1 \le n \le 10^5$) — the number of cadets doing the exam.

The next n lines contains, each, one integer s_i ($0 \le s_i \le 10^6$) — how many points the i-th cadet earned. The i-th cadet finished the circuit before the j-th cadet, if $i < j$.

Output

For each cadet you must print his grade.

Examples

standard input	standard output
3 1 2 3	3 3 3
5 5 4 3 2 3	5 4 3 2 2

Problem K. Killua's Race

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

During the *Hunter's Exam*, you met `Gon` and `Killua`.

The hidden phase of *Hunter's Exam* (not shown on the anime, because it's hidden) consists of a race inside a city. The city has m bidirectional roads and n road intersections. All rookies have to go from intersection 1 to intersection n.

You, `Gon` and `Killua` are very fast and you don't care very much about the race results, so `Killua` challenged you two to race under certain constraints:

- You have to take a path whose the number of roads in it is divisible by 3.
- `Gon` has to take a path whose the number of roads in it minus 1 is divisible by 3.
- `Killua` to take a path whose the number of roads in it minus 2 is divisible by 3.

A path $(p_0, p_1, ..., p_n)$ has a total of n roads in it (repeating the same road is allowed and counts for each time you walk on the roads).

The three of you are smart enough to take the shortest path under the race constraints. Also you're allowed to walk the same road or intersection more than once, but as soon as you reach the intersection n you can't keep walking (it's part of the *Hunter's Exam*'s rules).

What are the race results?

Input

The first line of input contains two integers, n and m ($2 \le n \le 10^5$, $0 \le m \le min(n \times (n-1)/2, 10^5)$) — the number of road intersections and the number of roads in the city.

The next m lines contains, each, three integers, u_i, v_i and w_i ($1 \le u_i, v_i \le n$, $1 \le w_i \le 10^3$) — the road end-points and length.

It's guaranteed that there's no two roads i and j such that $(u_i, v_i) = (u_j, v_j)$ or $(u_i, v_i) = (v_j, u_j)$.

Output

Print three lines, containing the order you, `Gon` and `Killua` get to the intersection n.

Each line must be one of the three strings: `"me"`, `"Gon"` or `"Killua"` (without quotes).

If two or more reach the intersection n at the same time, you can output them in any order.

If someone can't reach the intersection n under certain constraints, it should be the last one.

If more than one can't reach the intersection n, both should be the last ones, in any order.

Examples

standard input	standard output
4 5 1 4 1 1 2 2 2 4 2 2 3 3 3 4 3	Gon Killua me
3 2 1 2 2 2 3 1	Killua Gon me

Problem L. Luffy's Route

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Your pirate crew entered the *Grand Line*, *Yahoo*!!!

The *Grand Line* consists of n islands, each located at some (x_i, y_i). The *Reverse Mountain*, the start of the *Grand Line*, is located at position $(0, 0)$ and you want to enter the *New World*, which is located at $(l, 0)$. All islands on *Grand Line* are located between the *Calm Belts* which are located at $y = -1000$ and $y = 1000$.

You navigate the *Grand Line* using the *Exp Pose* (The *Log Pose* is out-dated, nobody uses it nowadays). The *Exp Pose* is a multi-compass that points to all islands located at (x_j, y_j) if you are located at island located at (x_i, y_i) which $|x_i - x_j| + |y_i - y_j| \leq 5$ and $x_i < x_j$. `Exp Pose` also points to `New World` entrance, if all conditions apply. It's too dangerous to sail freely, thus your crew will always navigate only to islands that the *Exp Pose* is pointing to.

Your crew is very strong and fearless. To enjoy the most of the *Grand Line* you decided to visit as many islands as possible before entering the *New World*. What is the maximum number of islands your crew can visit?

Input

The first line of input consists of two integers, n and l ($0 \leq n \leq 10^5$, $1 \leq l \leq 10^5$) — the number of islands and the location of the *New World* entrance.

The next n lines contains, each, two integers, x_i and y_i ($0 < x < l$, $-1000 \leq y_i \leq 1000$) — the location of the i-th island.

It's guaranteed that no two islands are located at the same position.

It's guaranteed that there's always at least one path from the *Reverse Mountain* to the *New World* entrance using the *Exp Pose*.

Output

Print the maximum number of islands your crew can visit between the *Reverse Mountain* and the *New World* entrance.

Examples

standard input	standard output
2 5 1 1 4 1	2
5 10 2 -2 3 3 5 0 6 0 8 3	4