Problem A. Angel Beats!

Input file:	`angel.in`
Output file:	`angel.out`
Time limit:	2 seconds
Memory limit:	256 mebibytes

If you don't act, you'll be obliterated.

Tenshi has gained a new guard skill: Harmonics. Using this skill, Tenshi can create clones of herself, the clones can create other clones, and so on. But the skill went out of control, and the number of Tenshi's clones has grown rapidly. Now there are p copies of Tenshi, including herself. They are numbered so that Tenshi has number 0 and her clones have numbers $1, 2, \ldots, p-1$.

Yurippe wants to interfere. She has invisibly crawled in Tenshi's lair where the main computer which controls her skill is situated. There, she infected the computer with a virus with parameter s. The virus changes the skill's program in such a way that every few seconds, an integer a is chosen uniformly and randomly between 0 and $p-1$, inclusive, and clone number a merges with clone number $a^s \bmod p$. The resulting clone inherits all numbers of the initial ones: for example, if clone number 2 merges with clone number 3, any other clone attempting to merge with either clone 2 or clone 3 is in fact merged with the resulting clone. If for some a, the two numbers a and $a^s \bmod p$ already point to the same clone, nothing happens.

You have to calculate the number of ways of choosing the parameter s so that every clone eventually merges either with the first Tenshi's clone (numbered 1) or with the last one (numbered $p-1$). Thus, no more than two clones can remain. Note that Tenshi herself has number 0, so she never merges with any of her clones. The number s must be an integer such that $1 \leqslant s \leqslant p-1$.

Input

Prime number p, the amount of Tenshi's clones including herself ($2 \leqslant p \leqslant 2^{63}-1$).

Output

Number of ways to choose the parameter s.

Examples

angel.in	angel.out
3	2
13	2

Problem B. Burai Kaiji

Input file:	`burai.in`
Output file:	`burai.out`
Time limit:	2 seconds
Memory limit:	256 mebibytes

> I dug my own grave! Twice! Double Grave!

Summer night. Ocean. Luxurious liner. Gambling. Our hero aboard this liner. And tonight the stake is... his life.

This time Kaiji will have to play the game “Rock, scissors, paper, pencil, fire, water, air, earth...” There are N different types of cards in this game. All types are numbered from 1 to N. Kaiji stores his cards in decks. He always puts cards of the same type in the same deck, and cards of different types in different decks. Index of every deck is equal to the index of type of cards it contains. At any moment of time, Kaiji can have from 0 to 999 999 cards of each type. When not making a turn, he can buy and sell cards. The details of these operations are too complicated, so we will just assume that he can somehow set the amount of his cards of type i equal to v. During a turn, he can gamble using only decks with indices from segment $[i, j]$ where i and j are the parameters of the turn. After each turn, the amount of cards he has doesn’t change.

Kaiji has already examined the behavior and strategies of all gamblers and invented a winning strategy. All he needs now is to quickly find the answers to the following type of questions: on a turn with parameters i and j, what is the amount of cards in the k-th largest deck among the decks he can use? Help him answer these questions.

Input

First line contains an integer N, the number of types of cards ($1 \leqslant N \leqslant 50\,000$). Next line contains N integers a_i, the initial amount of cards of each type which Kaiji has ($0 \leqslant a_i < 10^6$). Next line contains an integer Q, the number of events occurred ($1 \leqslant Q \leqslant 10^5$). Q lines follow, each describing one event in one of the following ways:

- `s` i v, where $1 \leqslant i \leqslant N$ and $0 \leqslant v < 10^6$, means that Kaiji somehow sets the amount of his cards of type i equal to v;
- `g` i j k, where $1 \leqslant i \leqslant j \leqslant N$ and $1 \leqslant k \leqslant j - i + 1$, means that Kaiji wants to know the amount of cards in the k-th largest deck among his decks with indices from segment $[i, j]$.

Output

For each query of the second type, output the required amount of cards on a separate line.

Example

burai.in	burai.out
3	3
2 8 3	8
5	3
g 1 3 2	2
g 2 2 1	
s 2 3	
g 2 3 2	
g 1 2 1	

Problem C. Code Geass

Input file:	`code.in`
Output file:	`code.out`
Time limit:	2 seconds
Memory limit:	256 mebibytes

> People are not equal!

At last! It happened! Emperor Charles is affected by Geass! Lelouch has achieved his vengeance!

Hovever, there is something strange in Emperor's head. While using Geass, Lelouch sees the neural network of a human brain. Right now, he sees that there are N neurons in Emperor's brain and M undirected synapses connecting pairs of neurons. Lelouch instantly numbered all neurons with integers from 1 to N. There are Q key pairs of neurons in Emperor's brain. The neurons in the first key pair have numbers A_1 and B_1, and the numbers of neurons in key pairs 2, 3, …, Q can be calculated in the following way:

$$\begin{aligned} A_i &= B_{i-1} \\ B_i &= (13\,579 \cdot A_{i-1} + 97\,531 \cdot B_{i-1}) \bmod N + 1 \end{aligned}$$

A key pair of neurons is safe if there are at least two non-overlapping chains between the neurons of the pair (see definitions below). Lelouch knows that if Geass was used successfully, all key pairs will be safe. On the other hand, if there are too many unsafe key pairs, things may get out of control, and in that case, Lelouch will have only two seconds before Emperor uses his own Geass on Lelouch. Therefore, he must quickly find the number of good key pairs of neurons. Help him find it.

A chain is a sequence of one or more neurons where every two successive neurons are connected by a synapse. Two chains are called non-overlapping if they don't have common neurons except the first one and the last one. Note that by this definition, non-overlapping chains can coincide if they both consist of one or two neurons.

Input

The first line contains two integers N and M, the number of neurons and synapses in Emperor's brain ($1 \leqslant N \leqslant 10^5$, $0 \leqslant M \leqslant 10^5$). Next follow M lines. Each of them describes one synapse and contains two integers U and V, the numbers of neurons connected by the synapse ($1 \leqslant U, V \leqslant N$, $U \neq V$). There is no more than one synapse between each pair of neurons. The last line contains three integers Q, A_1 and B_1 ($1 \leqslant Q \leqslant 25\,000\,000$, $1 \leqslant A_1, B_1 \leqslant N$).

Output

Output the number of safe key pairs of neurons.

Example

code.in	code.out

6 6
1 2
2 3
3 4
4 5
4 6
6 2
8 1 2
	6

P.S. Even Lelouch doesn't know how to use pseudo-random dependency between key pairs of neurons.

Problem D. Death Note

Input file:	`death.in`
Output file:	`death.out`
Time limit:	3 seconds
Memory limit:	256 mebibytes

> The human whose name is written in this note will die.
>
> Death Note: How to use it, first rule

Light: L!

L: Kira.

L and Light: *(in unison)* No matter what, I will find you and do away with you!

Light: I am...

L: I am...

L and Light: *(in unison)* Justice!

Criminals around the world are being mysteriously murdered by a serial killer, nicknamed Kira. The police has analyzed the time of murders, and it turned out that 68% of all deaths took place on weekdays between 4 PM and midnight. Using these data, detective L concluded that Kira is a schoolboy. In order to falsify his assuming, Light (who is actually Kira) decided to kill people in random moments of time.

After L got Kira's message "Shinigami eat only apples", he decrypted it and understood that Kira will kill criminals according to the following pattern: one will be killed at moment T_1, another one at moment T_2, and the rest at moments $T_i = (T_{i-1} \cdot T_{i-2}) \bmod 10\,007$ for consecutive integers $i = 3, 4, \ldots, k$. Each of these numbers is the time of day at which a murder will happen. However, Kira did not reveal the numbers T_1 and T_2; as for the number k, it is only guaranteed that $k \geqslant 2$. The order of the murders is also not specified.

After that, L got the data from the police: the actual moments at which the criminals died. L is certain that this data contains all criminals which were killed by Kira according to the pattern, but it can contain some irrelevant murders as well. Now he needs to find out how much of them could have been killed by Kira.

Input

The first line contains an integer N, the number of criminals killed ($2 \leqslant N \leqslant 10^4$). The second line contains N integers a_i, the moments of time in which deaths of criminals took place ($0 \leqslant a_i < 10\,007$). Bear in mind that any number of people can be killed (by Kira or otherwise) at the same time of day.

Output

Output the maximum amount of criminals who could have been killed by Kira.

Example

death.in	death.out

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

Kira can be responsible for three deaths if, for example, $T_1 = 3$, $T_2 = 2$ and $k = 3$ (so that $T_3 = 6$).

Problem E. Evangelion

Input file:	`eva.in`
Output file:	`eva.out`
Time limit:	2 seconds
Memory limit:	256 mebibytes

Both of you, dance like you want to win!

After they lost battle against Israfel, the seventh Angel, Shinji and Asuka have to learn how to work in team. They can defeat the Angel only if they act synchronously. In NERV headquarters, they invented an attack plan.

The field of battle will be the city, in fact, the city ruins. Map of the city is divided into equal squares. Obstacles are marked by '`#`', shafts for Evangelions by '`*`', and all other squares by '`.`'. The Angel is situated strictly in the center of the map. Two Evangelions Eva01 and Eva02 (driven by Shinji and Asuka) must first be launched on surface from different shafts. After that, they must synchronously move and continuously attack the Angel with long-range weapons during the whole period of movement. Additional requirement is that the paths of the Evangelions to be symmetric relative to the Angel. Finally, they must return back to the underground base using some pair of different shafts. These shafts must be different from the starting shafts, too (so that exactly four different shafts are used).

Trajectories of Evangelions can not intersect each other, go through obstacles or go outside the field of battle. Additionally, they are unable to pass through one square more than once. Moving through shafts and attacking through obstacles is however possible. An Evangelion can move from one square to another if they share a common edge and spends one second per such action. The energy reserves of an Evangelion are enough for five minutes of action.

Input

The first line contains an even number N, the size of the map ($2 \leqslant N \leqslant 50$). Next N lines consist of N characters each and describe the map. Each of these characters is either '`#`' for an obstacle, '`*`' for a shaft or '`.`' for all other squares.

Output

Output just one word "`Impossible`" if there is no successful plan. Otherwise, output "`Possible`" on the first line. Then output two numbers on the second line: row and column numbers of Eva01 starting shaft. Rows are numbered from top (1) to bottom (N), columns are numbered from left (1) to right (N). On the third line output the path itself as a sequence of commands. Each command is either '`L`' (move left), '`R`' (move right), '`U`' (move up) or '`D`' (move down). If there are several possible solutions, you can output any of them.

Example

eva.in	eva.out

`6`
`*.....`
`#..#*.`
`....#.`
`.#...#`
`.*#...`
`.....*`
	`Possible`
	`6 6`
	`LLLLU`

Eva02 will move from the square $(1, 1)$ along the path `RRRRD`.

Problem F. Fullmetal Alchemist

Input file:	`fma.in`
Output file:	`fma.out`
Time limit:	2 seconds
Memory limit:	256 mebibytes

The genuine truth behind the truths.

Shou Tucker, the Sewing-Life Alchemist, loves his daughter Nina very much. Almost as much as his job. But life is cruel. And now Nina is transformed into a chimera. So Tucker must make a miracle in order to transform her back to human.

Human transmutation is very hard. To achieve success, Tucker must draw a "transmutation circle" in his square laboratory of size $2L \times 2L$. This "circle" must be a strictly convex polygon with at least $\lfloor 5.4 \cdot L^{2/3} \rfloor$ vertices situated in points with integer coordinates. A polygon is strictly convex if it is convex and no three vertices lie on the same line. The origin of coordinate system of the laboratory is situated in the center of the room and its axes are parallel to the walls. It is possible to put a vertex near the very wall of the laboratory.

Input

An integer L ($1 \leqslant L \leqslant 10^6$).

Output

In the first line output N, the number of vertices in the suitable transmutation circle. Next N lines must contain coordinates of vertices of this circle in counterclockwise order.

Example

fma.in	fma.out

3	12
	-3 -2
	-2 -3
	-1 -3
	1 -2
	2 -1
	3 1
	3 2
	2 3
	1 3
	-1 2
	-2 1
	-3 -1

Problem G. Gyakkyou Burai Kaiji

Input file:	`gyakkyou.in`
Output file:	`gyakkyou.out`
Time limit:	4 seconds
Memory limit:	256 mebibytes

> You shouldn't let kings like myself draw twice.

Once before he was there. Kaiji lost everything. The only thing he kept was his miserable life.

The rules of this game are almost the same. There are N different types of cards. All types are numbered from 1 to N, inclusive. Kaiji stores his cards in decks. He always puts cards of the same type in the same deck, and cards of different types in different decks. Index of every deck is equal to the index of type of cards it contains.

At any moment of time, Kaiji can have from 0 to 999 999 999 cards of each type. However, now gamblers cannot buy, sell or exchange cards. Thus the amount of cards of each type Kaiji has stays the same during the whole game. During a turn, Kaiji can gamble using only decks with indices from segment $[i, j]$ where i and j are the parameters of the turn.

Kaiji has already examined the behavior and strategies of all gamblers and invented a winning strategy. All he needs now is to quickly find the answers to the following type of questions: on a turn with parameters i and j, what is the amount of cards in the k-th largest deck among the decks he can use? Help him answer these questions.

Input

The first line contains an integer N, the amount of types of cards ($1 \leqslant N \leqslant 450\,000$).

The second line is used to generate integers a_i, the initial amount of cards of each type which Kaiji has ($0 \leqslant a_i < 10^9$). It contains three integers a_1, l and m ($0 \leqslant a_1, l, m < 10^9$); for $2 \leqslant i \leqslant N$,

$$a_i = (a_{i-1} \cdot l + m) \bmod 10^9.$$

The third line contains an integer B, the number of opponents ($1 \leqslant B \leqslant 1000$). B lines follow, each describing the set of games with a particular opponent. Each set is described by ten integers. The first one is G, the number of games played with that opponent. Then follow x_1, l_x and m_x, then y_1, l_y and m_y, and finally, k_1, l_k and m_k ($1 \leqslant x_1 \leqslant y_1 \leqslant N$, $1 \leqslant k_1 \leqslant y_1 - x_1 + 1$, $0 \leqslant l_x, m_x, l_y, m_y, l_k, m_k < 10^9$). These are used to generate auxiliary sequences x_g and y_g and the actual parameters i_g, j_g and k_g for $1 \leqslant g \leqslant G$:

$$\begin{array}{rcll} x_g & = & ((i_{g-1} - 1) \cdot l_x + m_x) \bmod N) + 1, & 2 \leqslant g \leqslant G \\ y_g & = & ((j_{g-1} - 1) \cdot l_y + m_y) \bmod N) + 1, & 2 \leqslant g \leqslant G \\ i_g & = & \min(x_g, y_g), & 1 \leqslant g \leqslant G \\ j_g & = & \max(x_g, y_g), & 1 \leqslant g \leqslant G \\ k_g & = & (((k_{g-1} - 1) \cdot l_k + m_k) \bmod (j_g - i_g + 1)) + 1, & 1 \leqslant g \leqslant G \end{array}$$

The generated parameters mean that in g-th game with the current opponent, Kaiji wants to know the amount of cards in the k_g-th largest deck among his decks with indices from segment $[i_g, j_g]$.

The total number of games played by Kaiji doesn't exceed 600 000.

Output

For each game g with each opponent b, find the number of cards in k_g-th biggest deck among his decks with indices from segment $[i_g, j_g]$. Output one number: the sum of these values.

Example

gyakkyou.in	gyakkyou.out

5
1 1 1
5
1 1 0 0 3 0 0 2 0 0
1 2 0 0 5 0 0 3 0 0
1 1 0 0 5 0 0 5 0 0
1 3 0 0 3 0 0 1 0 0
1 1 0 0 4 0 0 1 0 0
	15

Kaiji has i cards of i-th type for all $i = 1, 2, 3, 4, 5$. Each of these types was chosen exactly once. So the answer is 15.

Problem H. Haruhi Suzumiya No Yuutsu

Input file:	`haruhi.in`
Output file:	`haruhi.out`
Time limit:	3 seconds
Memory limit:	256 mebibytes

> — I have no interest in ordinary humans
>
> Haruhi Suzumiya

Haruhi: — Hey. Take a look at this.

Kyon: — What is this?

H: — Can't you tell?

K: — I can't. I have no idea.

H: — It's my SOS Brigade emblem!

K: — Emblem? Looks like a grumbling, drunken tapeworm to me.

H: — Look closer! See? It says SOS Brigade in the center, right? Come on, Kyon. Make this symbol display at the top of the site.

K: — You do it yourself.

H: — I'm the brigade chief! The chief's job is to give orders. You should try to use your head every once in a while.

Kyon reluctantly took master artist Haruhi-sama's portrait of a tapeworm, did some half-assed resizing, pasted it onto the file and uploaded it...

Next day.

H: — Look at this. Something's weird. Maybe the computer's acting up.

K: — Looks like the file on the server is corrupt.

H: — Isn't this strange? Maybe someone's committing cyber-terrorism against the SOS Brigade! Who is it?

It is hard to believe that JPEG compression algorithm can spoil pictures so much. Although...

The algorithm is based on DCT (Discrete Cosine Transform). If one has a matrix of numbers A (original picture) of size $n \times n$ where $n = 2^k$, then the transformed matrix B (the matrix used for JPEG compression) is computed like this:

$$b_{ij} = \sum_{k=0}^{n-1} \sum_{l=0}^{n-1} a_{kl} \cos\left(\frac{\pi i(2k+1)}{2n}\right) \cos\left(\frac{\pi j(2l+1)}{2n}\right)$$

Perhaps this time, the matrix B was damaged, and this caused the loss of quality.

K: — And how would I check it?

H: — Is it that difficult to find the matrix B? Don't be lazy, Kyon. I give you five hours. Do whatever you want, but calculate that matrix!

Input

First line contains an integer n, the size size of matrix ($n = 2^k$, $0 \leqslant k \leqslant 10$). Each of the next n lines contains n integers a_{ij}, the elements of matrix A ($-128 \leqslant a_{ij} \leqslant 127$).

Output

Output n lines with n integers in each: the elements of matrix B rounded to nearest integer using standard rounding rules (a fractional part of 0.5 is rounded so that the absolute value of a number becomes larger, e. g. 0.5 is rounded to 1, -0.5 is rounded to -1).

Example

haruhi.in	haruhi.out

2
1 2
3 4
	10 -1
	-3 0

Problem I. Ikkitousen

Input file:	`ikki.in`
Output file:	`ikki.out`
Time limit:	2 seconds
Memory limit:	256 mebibytes

> You called me stupid! The one who says stupid is the stupid one!

The Big Fighters Tournament is coming, and N students from every school must participate in it.

There are $2N - 1$ great fighters studying in Nanyo Academy. Each of them has a unique number from 1 to $2N - 1$ and is characterized by two main features: magatama power and breast size. Both of them are measured as integers from 1 to 10^9 (wow!).

Sonsaku Hakufu claims that magatama power is the best. But Saji Genpou disagrees with her. He is absolutely sure that breast size is significantly more important. In order to prevent a fight between them which would leave more than half of the fighters with severe wounds, Shuyu Koukin suggested to make a team so that both magatama power and breast sizes of the team are greater than the respective features of the remaining group of fighters (for a group of fighters, the value of a feature is the sum of values of this feature for individual fighters in this group). Of course, nobody believed it is possible. But two seconds passed, and Koukin already stood with an appropriate list of N members. Can you do the same?

Input

The first line contains an integer N, the size of the team ($1 \leqslant N \leqslant 10^5$). In the second line there are $2N - 1$ integers a_i, magatama powers of i-th candidate ($1 \leqslant a_i \leqslant 10^9$). The third line contains $2N - 1$ integers b_i, breast sizes of i-th candidate ($1 \leqslant b_i \leqslant 10^9$).

Output

Output exactly N different integers c_i, indices of students according to Koukin's selection ($1 \leqslant c_i \leqslant 2N - 1$).

Example

`ikki.in`	`ikki.out`

`3`
`800 700 500 900 400`
`1 5 3 4 2`
	`3 2 4`

Problem J. Jyu Oh Sei

Input file:	`jyu.in`
Output file:	`jyu.out`
Time limit:	2 seconds
Memory limit:	256 mebibytes

Only the strong survive here

A muse is a regular neighbor of human settlements on planet Chimaera. It is a huge tree which grows quickly to enormous size and causes massive damage. Nevertheless, it is still the main source of fresh water on the planet.

Recently the inhabitants of Ochre Ring received two unique artifacts, the Extractors, and now they can gain water more easily. A muse consists of nodes, some pairs of nodes are connected by branches. All they need is to install the Extractors into two (maybe the same) nodes of a muse. After that, all water is extracted from all branches on the shortest path between these nodes.

For each branch, we can determine the amount of water in it. Until the end of the day, Ochre Ring must get about K liters of water. Lack of water causes strict economy of it, excess of it may damage Extractors. So, it is required to find the amount of water closest to K that can be extracted from the neighboring muse. Once installed, the Extractors cannot be taken out.

Input

The first line contains two integers N and K, the number of nodes in the tree and the desired volume of water in liters ($1 \leqslant N \leqslant 50\,000$, $0 \leqslant K \leqslant 10^9$). After that, $N - 1$ lines follow describing the branches. Each of them contains three integers A, B and D ($1 \leqslant A, B \leqslant N$, $0 \leqslant D \leqslant 10^9$) where A and B are the nodes connected by the branch and D is the amount of water in that branch in liters.

Output

Output the integer $|K - L|$ where L is the amount of water closest to K which can be extracted from the tree using Extractors.

Example

jyu.in	jyu.out

6 10
1 2 3
1 3 4
1 4 1
2 5 5
2 6 2
	1