A. King of Thieves

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

In this problem you will meet the simplified model of game King of Thieves.

In a new ZeptoLab game called "King of Thieves" your aim is to reach a chest with gold by controlling your character, avoiding traps and obstacles on your way.

An interesting feature of the game is that you can design your own levels that will be available to other players. Let's consider the following simple design of a level.

A dungeon consists of n segments located at a same vertical level, each segment is either a platform that character can stand on, or a pit with a trap that makes player lose if he falls into it. All segments have the same length, platforms on the scheme of the level are represented as `'*'` and pits are represented as `'.'`.

One of things that affects speedrun characteristics of the level is a possibility to perform a series of consecutive jumps of the same length. More formally, when the character is on the platform number i_1, he can make a sequence of jumps through the platforms $i_1 < i_2 < \ldots < i_k$, if $i_2 - i_1 = i_3 - i_2 = \ldots = i_k - i_{k-1}$. Of course, all segments $i_1, i_2, \ldots i_k$ should be exactly the platforms, not pits.

Let's call a level to be *good* if you can perform a sequence of **four** jumps of the same length or in the other words there must be a sequence $i_1, i_2, \ldots, i_5$, consisting of **five** platforms so that the intervals between consecutive platforms are of the same length. Given the scheme of the level, check if it is good.

Input

The first line contains integer n ($1 \le n \le 100$) — the number of segments on the level.

Next line contains the scheme of the level represented as a string of n characters `'*'` and `'.'`.

Output

If the level is *good*, print the word `"yes"` (without the quotes), otherwise print the word `"no"` (without the quotes).

Examples

input
16 .**.*..*.***.**.
output
yes

input
11 .*.*...*.*.
output
no

Note

In the first sample test you may perform a sequence of jumps through platforms $2, 5, 8, 11, 14$.

B. Om Nom and Dark Park

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Om Nom is the main character of a game "Cut the Rope". He is a bright little monster who likes visiting friends living at the other side of the park. However the dark old parks can scare even somebody as fearless as Om Nom, so he asks you to help him.

The park consists of $2^{n+1} - 1$ squares connected by roads so that the scheme of the park is a full binary tree of depth n. More formally, the entrance to the park is located at the square 1. The exits out of the park are located at squares $2^n, 2^n + 1, \ldots, 2^{n+1} - 1$ and these exits lead straight to the Om Nom friends' houses. From each square i ($2 \le i < 2^{n+1}$) there is a road to the square $\lfloor \frac{i}{2} \rfloor$. Thus, it is possible to go from the park entrance to each of the exits by walking along exactly n roads.

To light the path roads in the evening, the park keeper installed street lights along each road. The road that leads from square i to square $\lfloor \frac{i}{2} \rfloor$ has a_i lights.
Om Nom loves counting lights on the way to his friend. Om Nom is afraid of spiders who live in the park, so he doesn't like to walk along roads that are not enough lit. What he wants is that the way to any of his friends should have in total the same number of lights. That will make him feel safe.

He asked you to help him install additional lights. Determine what minimum number of lights it is needed to additionally place on the park roads so that a path from the entrance to any exit of the park contains the same number of street lights. You may add an arbitrary number of street lights to each of the roads.

Input

The first line contains integer n ($1 \le n \le 10$) — the number of roads on the path from the entrance to any exit.

The next line contains $2^{n+1} - 2$ numbers $a_2, a_3, \ldots a_{2^{n+1}-1}$ — the initial numbers of street lights on each road of the park. Here a_i is the number of street lights on the road between squares i and $\lfloor \frac{i}{2} \rfloor$. All numbers a_i are positive integers, not exceeding 100.

Output

Print the minimum number of street lights that we should add to the roads of the park to make Om Nom feel safe.

Examples

input
2 1 2 3 4 5 6
output
5

Note

Picture for the sample test. Green color denotes the additional street lights.

C. Om Nom and Candies

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

A sweet little monster Om Nom loves candies very much. One day he found himself in a rather tricky situation that required him to think a bit in order to enjoy candies the most. Would you succeed with the same task if you were on his place?

One day, when he came to his friend Evan, Om Nom didn't find him at home but he found two bags with candies. The first was full of blue candies and the second bag was full of red candies. Om Nom knows that each red candy weighs W_r grams and each blue candy weighs W_b grams. Eating a single red candy gives Om Nom H_r joy units and eating a single blue candy gives Om Nom H_b joy units.

Candies are the most important thing in the world, but on the other hand overeating is not good. Om Nom knows if he eats more than C grams of candies, he will get sick. Om Nom thinks that it isn't proper to leave candy leftovers, so he can only eat a whole candy. Om Nom is a great mathematician and he quickly determined how many candies of what type he should eat in order to get the maximum number of joy units. Can you repeat his achievement? You can assume that each bag contains more candies that Om Nom can eat.

Input

The single line contains five integers C, H_r, H_b, W_r, W_b ($1 \le C, H_r, H_b, W_r, W_b \le 10^9$).

Output

Print a single integer — the maximum number of joy units that Om Nom can get.

Examples

input
10 3 5 2 3
output
16

Note

In the sample test Om Nom can eat two candies of each type and thus get 16 joy units.

D. Om Nom and Necklace

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

One day Om Nom found a thread with n beads of different colors. He decided to cut the first several beads from this thread to make a bead necklace and present it to his girlfriend Om Nelly.

Om Nom knows that his girlfriend loves beautiful patterns. That's why he wants the beads on the necklace to form a *regular* pattern. A sequence of beads S is *regular* if it can be represented as $S = A + B + A + B + A + \ldots + A + B + A$, where A and B are some bead sequences, " $+$ " is the concatenation of sequences, there are exactly $2k + 1$ summands in this sum, among which there are $k + 1$ "A" summands and k "B" summands that follow in alternating order. Om Nelly knows that her friend is an eager mathematician, so she doesn't mind if A or B is an empty sequence.

Help Om Nom determine in which ways he can cut off the first several beads from the found thread (at least one; probably, all) so that they form a *regular pattern*. When Om Nom cuts off the beads, he doesn't change their order.

Input

The first line contains two integers n, k ($1 \le n, k \le 1\,000\,000$) — the number of beads on the thread that Om Nom found and number k from the definition of the regular sequence above.

The second line contains the sequence of n lowercase Latin letters that represent the colors of the beads. Each color corresponds to a single letter.

Output

Print a string consisting of n zeroes and ones. Position i ($1 \le i \le n$) must contain either number one if the first i beads on the thread form a regular sequence, or a zero otherwise.

Examples

input
7 2 bcabcab
output
0000011

input
21 2 ababaababaababaababaa
output
000110000111111000011

Note

In the first sample test a regular sequence is both a sequence of the first 6 beads (we can take A = "", B = "bca"), and a sequence of the first 7 beads (we can take A = "b", B = "ca").

In the second sample test, for example, a sequence of the first 13 beads is regular, if we take A = "aba", B = "ba".

E. Transmitting Levels

time limit per test: 3 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Optimizing the amount of data transmitted via a network is an important and interesting part of developing any network application.

In one secret game developed deep in the ZeptoLab company, the game universe consists of n levels, located in a circle. You can get from level i to levels $i - 1$ and $i + 1$, also you can get from level 1 to level n and vice versa. The map of the i-th level description size is a_i bytes.

In order to reduce the transmitted traffic, the game gets levels as follows. All the levels on the server are divided into m groups and each time a player finds himself on one of the levels of a certain group for the first time, the server sends all levels of the group to the game client as a single packet. Thus, when a player travels inside the levels of a single group, the application doesn't need any new information. Due to the technical limitations the packet can contain an arbitrary number of levels but their total size mustn't exceed b bytes, where b is some positive integer constant.

Usual situation is that players finish levels one by one, that's why a decision was made to split n levels into m groups so that each group was a continuous segment containing multiple neighboring levels (also, the group can have two adjacent levels, n and 1). Specifically, if the descriptions of all levels have the total weight of at most b bytes, then they can all be united into one group to be sent in a single packet.

Determine, what minimum number of groups do you need to make in order to organize the levels of the game observing the conditions above?

As developing a game is a long process and technology never stagnates, it is yet impossible to predict exactly what value will take constant value b limiting the packet size when the game is out. That's why the developers ask you to find the answer for multiple values of b.

Input

The first line contains two integers n, q ($2 \le n \le 10^6$, $1 \le q \le 50$) — the number of levels in the game universe and the number of distinct values of b that you need to process.

The second line contains n integers a_i ($1 \le a_i \le 10^9$) — the sizes of the levels in bytes.

The next q lines contain integers b_j ($\max\{a_i\} \le b_j \le 10^{15}$), determining the values of constant b, for which you need to determine the answer.

Output

For each value of k_j from the input print on a single line integer m_j ($1 \le m_j \le n$), determining the minimum number of groups to divide game levels into for transmission via network observing the given conditions.

Examples

input
6 3 2 4 2 1 3 2 7 4 6
output
2 4 3

Note

In the test from the statement you can do in the following manner.

- at $b = 7$ you can divide into two segments: $2|421|32$ (note that one of the segments contains the fifth, sixth and first levels);
- at $b = 4$ you can divide into four segments: $2|4|21|3|2$;
- at $b = 6$ you can divide into three segments: $24|21|32|$.

F. Pudding Monsters

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

In this problem you will meet the simplified model of game Pudding Monsters.

An important process in developing any game is creating levels. A game field in Pudding Monsters is an $n \times n$ rectangular grid, n of its cells contain monsters and some other cells contain game objects. The gameplay is about moving the monsters around the field. When two monsters are touching each other, they glue together into a single big one (as they are from pudding, remember?).

Statistics showed that the most interesting maps appear if initially each row and each column contains exactly one monster and the rest of map specifics is set up by the correct positioning of the other game objects.

A technique that's widely used to make the development process more efficient is reusing the available resources. For example, if there is a large $n \times n$ map, you can choose in it a smaller $k \times k$ square part, containing exactly k monsters and suggest it as a simplified version of the original map.

You wonder how many ways there are to choose in the initial map a $k \times k$ ($1 \le k \le n$) square fragment, containing exactly k pudding monsters. Calculate this number.

Input

The first line contains a single integer n ($1 \le n \le 3 \times 10^5$) — the size of the initial field.

Next n lines contain the coordinates of the cells initially containing monsters. The i-th of the next lines contains two numbers r_i, c_i ($1 \le r_i, c_i \le n$) — the row number and the column number of the cell that initially contains the i-th monster.

It is guaranteed that all r_i are distinct numbers and all c_i are distinct numbers.

Output

Print the number of distinct square fragments of the original field that can form a new map.

Examples

input
5 1 1 4 3 3 2 2 4 5 5
output
10

G. Spiders Evil Plan

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Spiders are Om Nom's old enemies. They love eating candies as much as he does and that's why they keep trying to keep the monster away from his favorite candies. They came up with an evil plan to trap Om Nom.

Let's consider a rope structure consisting of n nodes and $n - 1$ ropes connecting the nodes. The structure is connected, thus, the ropes and the nodes form a tree. Each rope of the formed structure is associated with its length. A candy is tied to node x of the structure. Om Nom really wants to eat this candy.

The y spiders are trying to stop him from doing it. They decided to entangle the candy and some part of the structure into a web, thus attaching the candy to as large as possible part of the rope structure.

Each spider can use his web to cover all ropes on the path between two arbitrary nodes a and b. Thus, y spiders can cover the set of ropes which is a union of y paths in the given tree. These y paths can arbitrarily intersect each other. The spiders want the following conditions to be hold:

- the node containing the candy is adjacent to at least one rope covered with a web
- the ropes covered with the web form a connected structure (what's the idea of covering with a web the ropes that are not connected with the candy?)
- the total length of the ropes covered with web is as large as possible

The spiders haven't yet decided to what node of the structure they will tie the candy and how many spiders will cover the structure with web, so they asked you to help them. Help them calculate the optimal plan for multiple values of x and y.

Input

The first line contains numbers n and q ($1 \le n, q \le 10^5$) — the number of nodes in the structure and the number of questions that the spiders want to ask you.

The next $n - 1$ lines determine the rope structure. The i-th line contains three integers u_i, v_i, l_i ($1 \le u_i, v_i \le n$, $u_i \neq v_i$, $1 \le l_i \le 1000$), showing that there is a rope of length l_i between nodes u_i and v_i.

Next q lines describe the spiders' questions. As they want you to answer their question online, they encoded their messages in a special manner.

Each of the next q lines contains two numbers x_i, y_i. In the first question of the spiders $x = x_1, y = y_1$.

To calculate values x and y in the spiders' i-th ($2 \le i \le q$) question, you need to use the following formulas:

$$x = ((x_i + Ans_{i-1} - 1) \bmod n) + 1$$

$$y = ((y_i + Ans_{i-1} - 1) \bmod n) + 1$$

where Ans_{i-1} is the total length of the ropes covered by a web in the answer for the $(i - 1)$-th question.

The following inequality holds: $1 \le x_i, y_i \le n$.

Output

For each question of the spiders print on a separate line a single integer Ans_i — the total length of the ropes covered with web in the optimal plan.

Examples

input
6 3 1 2 2 2 3 2 3 4 2 4 6 1 3 5 10 3 1 2 5 1 1
output
14 13 17