Problem A. Cards

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

A certain deck consists of n special cards. For each card, one face is colored white, while the other is colored black. Each face of each card has a number written on it. Each of the numbers from 1 to n is written exactly once among the white card faces.

Find the largest set of cards that can be chosen in such a way that no number appears among both the white and the black faces of the cards!

Input

The first line of input contains a single integer n ($1 \leqslant n \leqslant 10^6$), the number of cards in the deck. The i-th of the following n lines contains a single integer a_i ($1 \leqslant a_i \leqslant 2 \cdot 10^6$), the number written on the black face of the i-th card. The white face of the i-th has the number i written on it.

Output

On the first line of output print a single integer m, the maximum number of cards that can be chosen. On each of the next m lines output a single integer, the number on the white face of one card. If multiple solutions exist, output any one of them.

Example

standard input	standard output
6 3 4 4 7 5 5	3 1 2 6

Problem B. Doors

Input file:	*standard input*
Output file:	*standard output*
Time limit:	3 seconds
Memory limit:	512 mebibytes

There are several consecutive doors in Scrooge's house. Each door has some number of locks (at least one). There are k locks in total numbered from 1 to k. When leaving the house, Scrooge passes through the doors in the same order each day and closes some of the locks. If a door has more than one lock, the order in which they are closed can differ from day to day. Some of the locks can be left open, and some of the doors can be left open as well by leaving open all the locks on that door.

Given the order in which the locks have been closed on each of n days by Scrooge leaving the house, determine the maximum possible number of doors in his house, along with the lock distribution on the doors.

Input

The first line of the input contains two integers n and k ($1 \leqslant n, k \leqslant 10^5$): the number of days and the total number of locks, accordingly. The i-th of the following n lines contains the description of the lock closing process on the i-th day. The first number of such description, s_i ($1 \leqslant s_i \leqslant k$), describes the number of locks closed on the i-th day. The following distinct s_i integers describe the order in which the locks were closed, the j-th of these numbers being the number of the j-th closed lock. The sum of s_i does not exceed 10^6. All numbers on each line are separated by spaces.

Output

On the first line of output, print an integer m: the maximum possible number of doors in Scrooge's house. In the next m lines, output the distribution of the locks on the doors. On each of these lines, output the information about a single door: first, the number of locks s on that door, then s integers which are the numbers of the locks on that door. You may output the lines and the numbers of the locks in any order. If there are multiple solutions, output any of them.

Note that in your solution, each of the k locks must be placed on exactly one door.

Examples

standard input	standard output
2 9 9 6 3 5 8 9 7 2 4 1 8 3 6 7 9 5 8 1 4	4 1 2 2 6 3 2 4 1 4 5 9 7 8
3 5 3 5 3 1 2 3 2 3 3 5 2	4 1 1 1 2 2 5 3 1 4

Problem C. Fine Brochures

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	512 mebibytes

A printed version of a certain exam consists of a set of k pages, namely 1, 2, …, k. The exam has been copied m times, and all the pages have been consecutively arranged into a list:

$$\underbrace{1\ 2\ \dots\ k\ 1\ 2\ \dots\ k\ \dots\ 1\ 2\ \dots\ k}_{m \text{ sets of } k \text{ pages}}$$

A brochure machine makes a single brochure out of the leftmost n pages of the page list (if there are less than n pages, the brochure is made of all the remaining pages in the list). The order of the pages in the brochure remains the same as it was in the list. We call a brochure *fine* if it contains k consecutive pages with correct ordering (i. e., $1\ 2\ \dots\ k$ in this order).

Your task is to determine the total number of fine brochures after the brochure machine processes the whole page list.

Input

The single line of the input contains three space-separated positive integers k, m and n ($1 \le k \cdot m \le 9 \cdot 10^{18}$ and $1 \le n \le 9 \cdot 10^{18}$): the number of exam pages, the number of exam copies and the number of pages the brochure machine takes from the list.

Output

On the single line of the output, print the total number of fine brochures.

Examples

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

Note

In the first example, each brochure contains a correctly ordered set of exam pages (underlined):

$$\boxed{\underline{1\ 2}\ 1} \quad \boxed{2\ \underline{1\ 2}} \quad \boxed{\underline{1\ 2}\ 1} \quad \boxed{2\ \underline{1\ 2}}$$

In the second example, only the first and two last brochures are fine:

$$\boxed{\underline{1\ 2\ 3\ 4}\ 1} \quad \boxed{2\ 3\ 4\ 1\ 2} \quad \boxed{3\ 4\ 1\ 2\ 3} \quad \boxed{4\ \underline{1\ 2\ 3\ 4}} \quad \boxed{\underline{1\ 2\ 3\ 4}}$$

Problem D. Fragile Keyboard

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	512 mebibytes

Arnold has a fragile old keyboard. Each key works only the first time it is pressed. For example, typing the word "ABRACADABRA" produces "ABRCD". The same result is also obtained with the words "ABARACADABA" and "ABRRRRBRRACAD".

For the given resulting word, Arnold writes down all the words that can produce the result in a special order. In his order, shorter words come before longer ones, and words of the same length are ordered lexicographically. For example, Arnold's list for the resulting word "ARC" begins with:

ARC, AARC, ARAC, ARCA, ARCC, ARCR, ARRC, AAARC, AARAC, ...

Given the resulting word and the index of the word in Arnold's list, find the word at that index that produces the given result!

Input

The first line of input contains two space-separated integers n and k ($2 \leqslant n \leqslant 26$, $1 \leqslant k \leqslant 10^{15}$), the length of the resulting word and the position index of the original word in Arnold's list. The second line of input contains a string of length n, the resulting word. It contains only uppercase English letters, and all the characters of the string are pairwise distinct.

Output

Output a single string: the original word at the k-th position in Arnold's list.

Examples

standard input	standard output
3 9 ARC	AARAC
2 17 OH	OHOHO

Problem E. Goodness

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	512 mebibytes

A certain deck consists of n special cards. For each card, one face is colored green, while the other is colored red. Each face of the card has a number written on it. For two cards a and b, the *goodness* of a with respect to b is calculated as the product of the number written on the green face of a and the number written on the red face of b.

For instance, if a has the numbers 10 and 3, while b has the numbers 7 and -2 written on the green and red faces, accordingly, then the goodness of a with respect to b is $10 \cdot (-2) = -20$. On the other hand, the goodness of b with respect to a is $7 \cdot 3 = 21$. The difference of these relative goodnesses is either 41 or -41.

Given the numbers written on each face of each card, find the maximum possible difference of relative goodnesses for some two cards.

Input

The first line of input contains a single integer n ($2 \leqslant n \leqslant 10^5$), the number of cards in the deck. The i-th of the following n lines contains two space-separated integers g_i and r_i ($-10^9 \leqslant g_i, r_i \leqslant 10^9$), the numbers written on the green and red faces of the i-th card, accordingly.

Output

Output a single integer: the maximum possible difference of relative goodnesses of some two cards from the deck.

Example

standard input	standard output
5 9 -1 7 8 -2 4 9 -6 3 5	114

Note

The 2nd and the 4th cards give the maximum goodness difference:

$$9 \cdot 8 - 7 \cdot (-6) = 114.$$

Problem F. Hotel 11

Input file:	*standard input*
Output file:	*standard output*
Time limit:	3 seconds
Memory limit:	512 mebibytes

In a hotel with an unusual name "11" all the rooms are traditionally numbered only with the so-called *hotel* numbers. A hotel number is a positive integer whose decimal representation contains a substring that, when viewed as a number, is a multiple of 11. For example, the first few hotel numbers are 10, 11, 20, 22, 30, 33.

Given two positive integers a and b ($a \leqslant b$), determine the total number of hotel numbers in the interval $[a; b]$.

Input

The first line of input contains a single integer $1 \leqslant n \leqslant 10000$, the number of test cases. The i-th of n following lines contains two space-separated integers a_i and b_i ($1 \leqslant a_i \leqslant b_i \leqslant 10^{18}$).

Output

For the i-th test case, on the i-th line of output print the total number of hotel numbers in the interval $[a_i; b_i]$ (inclusive).

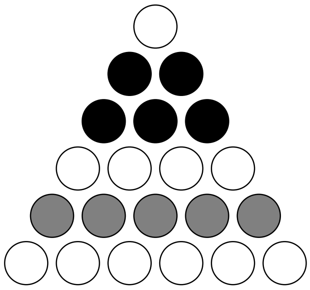
Example

standard input	standard output
3 231 253 10 20 17 17	7 3 0

Problem G. Mosaic

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	512 mebibytes

In this problem you have to make a mosaic using a set of colored marbles. There are exactly w white, g gray and b black marbles in the set. The mosaic consists of marble rows and has a triangular shape. The i-th row from the top contains i marbles and each row contains only marbles of the same color. All of the given marbles should be used in the mosaic.



For instance, if $w = 11$, $g = 5$ and $b = 5$, then it is possible to construct a mosaic as shown on the figure. There are also other ways of arranging the given marbles into a mosaic in this example. On the other hand, if $w = 3$, $g = 4$ and $b = 5$, then it is impossible to make any mosaic.

Given the number of marbles of each color, find an arrangement of the marbles into a single mosaic!

Input

The first line of input contains three space-separated numbers w, g and b ($0 \leqslant w, g, b < 2^{31}$). It is guaranteed that $w + g + b > 0$.

Output

If the given marbles cannot be arranged in a mosaic, print "`Impossible`" (without quotes). Otherwise print a single string s describing the mosaic. The i-th character of s describes the i-th row of the mosaic from the top and should be one of the "`W`", "`G`" or "`B`", if the marble color in the i-th row is white, gray or black, accordingly. If multiple solutions exist, you may output any one of them.

Examples

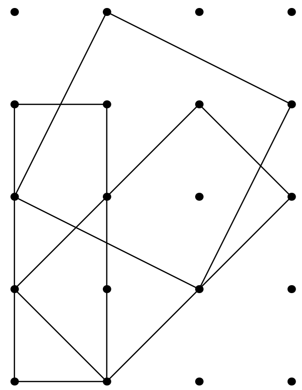
standard input	standard output
`11 5 5`	`WBBWGW`
`3 4 5`	`Impossible`

Problem H. Rectangle Count

Input file:	*standard input*
Output file:	*standard output*
Time limit:	4 seconds
Memory limit:	512 mebibytes

Given an $n \times m$ square lattice, find the number of rectangles with vertices at the lattice points!

For instance, the following figure shows some of the possible rectangles on a 5×4 lattice:



Input

The only line of input contains two space-separated integers n and m ($2 \leqslant n, m \leqslant 5000$), the width and the height of the lattice, accordingly.

Output

Output a single integer: the total number of non-degenerate rectangles with vertices at the given lattice points.

Examples

standard input	standard output
4 2	6
3 5	33

Problem I. Second Trip

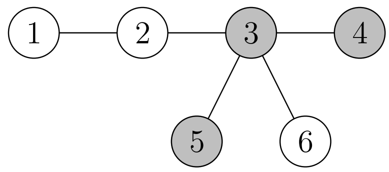
Input file: *standard input*
Output file: *standard output*
Time limit: 2 seconds
Memory limit: 512 mebibytes

Crocodile Island is a widely popular place for tourism. There are n cities on the island numbered from 1 to n. All cities are connected by a network of roads. For each two cities, there is a single path of the roads that connects them. For convenience, each city hosts an airport for tourists to arrive to and depart from.

Travel agency "New Sights" offers the organization of trips around Crocodile Island that proceed as follows: first, a tourist arrives at some city a by plane. Then the tourist may visit other cities by car, each city at most once. At the end of the trip the tourist departs home from city b, the last city he visited. The tourist may also stay in one city for the whole trip.

At this time "New Sights" are concerned with a problem: many tourists have already visited Crocodile Island once and are willing to make another visit. However, each tourist does not want to visit any city that he has already been to. For each of the tourists, the agency wants to know the number of different trips described earlier that it can offer to the tourist. Two trips are considered different if the sets of the cities in each trip visited are not equal. For instance, a trip that starts in a and ends in b is the same as a trip that starts in b and ends in a.

Consider a road network as shown on the figure. If, for instance, $a = 5$ and $b = 4$, the possible second trips are $(1, 1)$, $(1, 2)$, $(2, 2)$ and $(6, 6)$.



Given the description of the first trips of q tourists, calculate the number of different second trips the agency can offer to each of the tourists.

Input

The first line of the input contains two integers n and q ($1 \leqslant n, q \leqslant 10^5$): the number of cities and the number of queries, accordingly. The i-th of the following $n - 1$ lines contains two integers x_i and y_i: the cities that are connected by the i-th road ($x_i \neq y_i$, $1 \leqslant x_i, y_i, \leqslant n$). Each road is described exactly once. It is guaranteed that for each two cities, there is a single path of the roads that connects them.

In the next q lines, the description of the queries follows. The i-th query line consists of two integers a_i and b_i: the starting and ending cities of the i-th tourist's first trip, accordingly ($1 \leqslant a_i, b_i, \leqslant n$). In each line the numbers are separated by spaces.

Output

The output must contain q lines. In the i-th line, output a single integer: the total number of different trips that the agency can offer to the i-th tourist.

Example

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

Problem J. Ship

Input file:	*standard input*
Output file:	*standard output*
Time limit:	4 seconds
Memory limit:	**128 mebibytes**

A ship has been assigned to transport cargo across the river Daugava (Western Dvina) that has the capacity to carry at most t tons of cargo. There is a hangar on the right bank of the river that contains n cargo containers to be transported by the ship. The weight of each container is an integer number of tons and it does not exceed t tons. The containers are arranged in a single line from the start to the end of the hangar. A container can be carried onto the ship by a liftloader that can pick only a container either at the start or at the end of the line. The total weight of the containers placed on a ship cannot exceed the capacity of the ship. None of the containers are allowed to be opened, thus it is impossible to redistribute the cargo among the containers. It is also expensive to move the containers so once the container has been picked up by the liftloader, it is then directly carried onto the ship. Only then the liftloader can pick another container from the hangar. Once the ship is loaded, it can transport the containers to the left bank and return to the right bank empty.

Write a program that calculates the minimum possible number of trips the ship has to make to transport all of the containers to the left river bank!

Input

The first line of input contains two integers t and n ($1 \leqslant t \leqslant 10^9$, $1 \leqslant n \leqslant 10^4$), the capacity of the ship and the number of containers in the hangar, accordingly. The next line contains n integers, the weight of the containers in tons. The i-th number w_i ($1 \leqslant w_i \leqslant t$) denotes the weight of the i-th container from the start of the hangar. On each line the numbers are separated by spaces.

Output

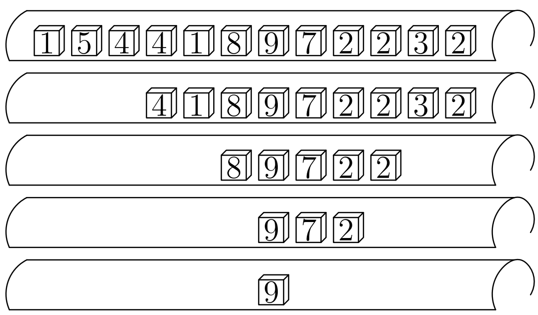
Output a single integer: the minimum possible trip number for the ship to transport all of the containers to the other river bank.

Examples

standard input	standard output
`10 12` `1 5 4 4 1 8 9 7 2 2 3 2`	`5`

Note

The following figure shows the remaining containers in the hangar before and after each trip in the given example.



Problem K. Table

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	512 mebibytes

For the preparation of the Great Conference, the organizers have manufactured a great table. To plan the arrangement of the seats at the table, the overview of the table has been drawn on a square grid. The table appears to be a polygon with the sides lying along the grid lines. It is now necessary to count the number of guests that can be seated around the table. A seat can be placed at each cell that shares at least one cell side with the perimeter of the table. For instance, the following figure shows the maximum number of seats that can be placed around each of the tables:

Given the overview of the table, find the maximum number of seats that can be placed around the table!

Input

The description of the table is given by the counter-clockwise traversal of the perimeter and begins from the lowest corner of the polygon (if there are multiple such corners, the leftmost of them) which is marked by a dot in the given figure.

The first line of input contains a single integer n ($4 \leqslant n \leqslant 10^5$), the number of polygon sides. In the second line n space-separated integers follow. The i-th integer s_i ($1 \leqslant |s_i| < 2^{31}$) denotes the i-th side in the clockwise traversal of the polygon perimeter. If $s_i > 0$, then the traversal is directed along the side upwards or to the right, and downwards or to the left, if $s_i < 0$. The length of the side equals $|s_i|$. It is guaranteed that the border of the table is a simple polygon, i. e. it has no self-intersections and no self-touchings.

Output

Output a single integer: the maximum number of seats that can be placed around the table. It is guaranteed that the answer is strictly less than 2^{31}.

Examples

standard input	standard output
12 5 1 1 1 -1 1 -2 -1 -1 1 -2 -3	16
4 1 3 -1 -3	8