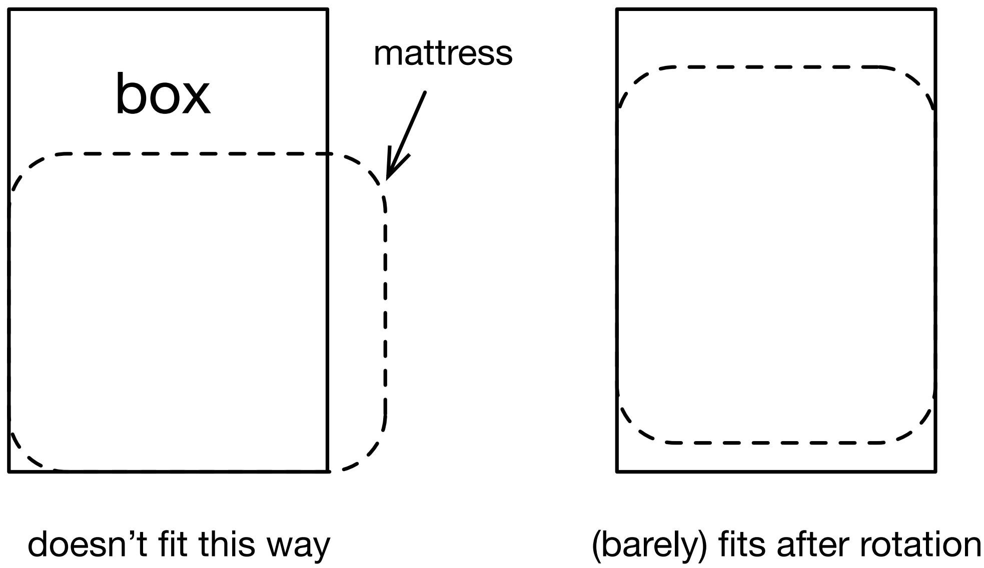
Problem A
Storing a Mattress
Problem ID: mattress
Time Limit: 3 seconds

You are moving and want your mattress to be stored nicely in a box for the move. Does it fit?

The height of the box is exactly the height of the mattress and does not factor into this problem. Given the lengths and widths of the box and mattress, can you fit the mattress in the box? You are allowed to rotate the mattress (i.e. exchange the width and length).



Input

Input consists of a single line containing four integers w_b, ℓ_b, w_m, ℓ_m in that order. Here, w_b and ℓ_b are the width and length of the box; w_m and ℓ_m are the width and length of the mattress. There will be a single space separating each of these integers. Each integer lies between 1 and 100 (yes, the mattress might have strange dimensions).

Output

Output a single line containing a single string `yes` or `no`, indicating if the mattress can fit in the box, perhaps after rotating it.

Tip

Do not output any prompt like `enter the mattress dimension`. This will be regarded as output and will not be accepted by the judging platform. When we say all integers will appear on a single line, we mean it. So do not write code that expects them to be on separate lines.

The output is case sensitive: notice all output in this question should be lowercase. Do not add extra spaces.

Finally, you should end the output line with a newline. This is done by default in Python using `print()`, but may need to be made done explicitly in other languages. You can print the single character `'\n'` in `C++` or `Java` to end the line with a newline. Of course, test before you submit!

Sample Input	Sample Output
`2 10 2 10`	`yes`

Sample Input	Sample Output
`2 10 10 2`	`yes`

Sample Input	Sample Output
`2 10 5 2`	`yes`

Sample Input	Sample Output
`2 10 5 4`	`no`

Problem B
Hot Potato
Problem ID: potato
Time Limit: 3 seconds

Some children are playing Hot Potato. They are arranged a circle. The child that is initially holding the potato is numbered 0 and the remaining children are numbered 1 through $n-1$ as you go clockwise around the circle from 0.

They take turns passing the potato around the circle. Each child can pass the potato either to the child on their left or the child on their right.

Given a sequence of left/right passes, output which child ended up holding the potato.

Input

Input consists of a single line containing an integer n and a string s consisting only of characters `L` (pass to the left) and `R` (pass to the right). Here, $2 \leq n \leq 100$ is the number of children and the string is the sequence of passes made by the children. The length of s is between 1 and 1000.

There will be exactly one space between n and the start of s.

For example, if $n = 5$ and $s =$ `RRLLL` the potato will pass through the following sequence of children

$$0 \xrightarrow{\texttt{R}} 4 \xrightarrow{\texttt{R}} 3 \xrightarrow{\texttt{L}} 4 \xrightarrow{\texttt{L}} 0 \xrightarrow{\texttt{L}} 1.$$

Output

Output consists of a single line with a single integer between 0 and $n-1$ indicating which child ended up with the potato.

Sample Input	Sample Output
`5 RRLLL`	`1`

Sample Input	Sample Output
`2 LRRRR`	`1`

Sample Input	Sample Output
`7 LRRLRRLRLLLLLLLLLRRL`	`6`

This page is intentionally left blank.

Problem C
I Hate Stairs!
Problem ID: stairs
Time Limit: 3 seconds

Have I ever mentioned that I hate stairs? Well, just going up them. Going down a flight of stairs does not bother me.

I work in a strange building. There are n rooms that are arranged in a circle and are numbered from 0 to $n-1$ in clockwise order. I am in room 0 and I have to go to a meeting in room m (some room number between 0 and $n-1$). So I can travel either clockwise from 0 until I reach m or counter-clockwise from 0 until I reach m.

The rooms are all at integer heights and adjacent rooms on the circle are connected by stairs. The number of stairs connecting two rooms is the difference in their height and, obviously, ascend from the lower room to the higher room. Help me determine the minimum number of stairs I must ascend to reach room m.

For example, if there are 7 rooms with heights

$$4, 5, -1, 3, 2, 2, 7$$

and the meeting is in room number 3 (which also has height three), then there are two ways I can reach room 3 from room number 0. The clockwise direction has me traverse rooms with heights:

$$4 \longrightarrow 5 \longrightarrow -1 \longrightarrow 3.$$

I climb 5 steps in total (1 for the $4 \rightarrow 5$ transition and 4 from the $-1 \rightarrow 3$ transition). Note some rooms can be underground and have negative height.

If I travel counterclockwise, then I traverse rooms with heights:

$$4 \rightarrow 7 \rightarrow 2 \rightarrow 2 \rightarrow 3$$

Here I only climb 4 steps in total (3 for the $4 \rightarrow 7$ transition and 1 for the $2 \rightarrow 3$ transition).

Input

The first line of input contains two integers n (the number of rooms) and m (the location of the meeting) You are guaranteed $3 \leq n \leq 1000$ and $0 \leq m < n$.

The second line of input contains n space-separated integers indicating the heights of the rooms (relative to the ground), starting from room 0. Each integer lies between -10^6 and 10^6.

Output

Output consists of a single line with a single integer indicating the minimum number of stairs that must be *ascended* when going from room 0 to room m.

Sample Input	Sample Output
7 3 4 5 -1 3 2 2 7	4

Sample Input	Sample Output
8 4 2 8 4 6 3 -1 3 -1	8

Sample Input	Sample Output
6 0 5 2 1 7 -2 10	0

Problem D
Bin Packing
Problem ID: bins
Time Limit: 3 seconds

Your job at the bin packing plant is to pack bins with items. The items come in various sizes, but all bins are identical: each can hold items with total size at most B.

To reduce shipping costs, you want to pack the items into the fewest bins possible. Unfortunately, this is a very hard problem: we do not know of an efficient algorithm to solve this problem!

So we resort to heuristics. Suppose the available bins are numbered $1, 2, 3, \ldots$ (you have more than enough bins). A popular heuristic is to pack the items in order from largest item to smallest item. For each item you consider, pack the item in the bin with the **lowest number** among all bins that can still hold the item (i.e. adding the item to the bin will still have the total size of items in the bin being at most B).

For example, if the items have size 7, 6, 2, 2, 2, 2, 1 and each bin can hold a total size of $B = 10$, then:

- The item with size 7 goes in bin 1
- The item with size 6 goes in bin 2 (it can't fit in bin 1).
- The item with size 2 goes in bin 1.
- The next item with size 2 goes in bin 2 (bin 1 is holding a total size of 9 by now so it cannot accommodate this item).
- The last item with size 2 goes in bin 2.
- Finally, the item of size 1 can fit in bin 1.

In this case, all bins are packed perfectly. In general, though, this algorithm does not always use the minimum number of bins (see the second sample).

Your job is to simulate the algorithm and report how many bins are used.

Input

The first line of input consists of two integers n, B. Here $1 \leq n \leq 1000$ and $1 \leq B \leq 1000$.

The second line consists of n integers indicating the size of the n items you must pack, each between 1 and B. A single space will separate consecutive integers on each line.

Output

Output a single line containing a single integer. This will be the number of bins used by the algorithm described above.

Sample Input	Sample Output
6 10 2 1 2 6 2 7	2

Sample Input	Sample Output
7 10 5 4 3 2 2 2 2	3

Problem E
Continued Fractions
Problem ID: continuedfractions
Time Limit: 3 seconds

Continued fractions are very cool! A continued fraction has the following form:

$$a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{a_3 + ...}}}$$

where each a_i is an integer and $a_i > 0$ for $i \geq 1$ (the ... may or may not terminate). Every rational number has a continued fraction representation with a finite number of terms.

In fact, many irrational numbers, which are well-known to have non-terminating and seemingly random decimal representations, can be expressed very nicely as continued fractions. For example, the square root of 2 can be expressed as

$$\sqrt{2} = 1 + \cfrac{1}{2 + \cfrac{1}{2 + \cfrac{1}{2 + \cfrac{1}{2 + \cfrac{1}{2 + \cfrac{1}{2 + ...}}}}}}$$

Even *more cool* is the tie between continued fractions and Pell's equation, in which we search for **integer pairs** (x, y) that satisfy

$$x^2 - ny^2 = 1$$

To solve the case $n = 2$, we can consider the series of *convergents* of the continued fraction for $\sqrt{n}$, formed by truncating the continued fraction at each point, and then simplifying to the form p/q.

- $1 + \frac{1}{2} = [1; 2] = \frac{3}{2}; 3^2 - 2 \cdot 2^2 = 1$
- $1 + \frac{1}{2+\frac{1}{2}} = [1; 2, 2] = \frac{7}{5}; 7^2 - 2 \cdot 5^2 = -1$
- $1 + \frac{1}{2+\frac{1}{2+\frac{1}{2}}} = [1; 2, 2, 2] = \frac{17}{12}; 17^2 - 2 \cdot 12^2 = 1$

We see that the convergents $\frac{3}{2}$ and $\frac{17}{12}$ yield the solutions $(3, 2)$ and $(17, 12)$. It turns out we can generate an infinite number of solutions in this way. Note that in the above examples we also introduce a more compact notation for writing continued fractions involving square brackets. See the Output section for more details on this format.

All of this is very cool indeed, but let's keep things simple for this problem. Given a *rational* number, can you output the continued fraction representation?

Input

The input consists of a single line, containing two integers p, q, where $1 \leq q < 2^{64}$ and $1 < p < 2^{64}$. This represents the rational number p/q.

Output

Output consists of a single line that gives the continued fraction representation in the following format:

- The first integer in the list is the whole number part of p/q. This is followed by a semicolon, if necessary (i.e. p/q is not an integer).
- The next integers represent the values that appear in successive denominators of the continued fraction. These numbers are separated by commas.
- The entire list is enclosed in square brackets
- There are no spaces in the list.

See the problem description and sample input/output for examples of this format. To be clear, the last number in the list should never be followed by a comma.

Note that we follow the convention that the final integer in the list should never be 1 for rational numbers, since a list ending in $...n, 1]$ could instead be written as $...(n + 1)]$, since $n + \frac{1}{1} = n + 1$.

Sample Input	Sample Output
`333 106`	`[3;7,15]`

Sample Input	Sample Output
`18 9`	`[2]`

Sample Input	Sample Output
`5 12`	`[0;2,2,2]`

Problem F
Recording Shows
Problem ID: pvr
Time Limit: 5 seconds

There are too many interesting TV shows these days! Your New Year's resolution is to watch them all, eventually.

However, you are stuck in the early 2000s and have not yet signed up for any streaming service. What you do have is a lot of PVRs (personal video recorders), devices that can record shows even when you are not watching them.

Each PVR is capable of recording one show at a time. Each show that is recorded has to be recorded continuously on a single PVR - you cannot change which device is recording a show midway through the show. However, once a PVR has stopped recording one show it can then record another.

You have listed all of the shows you want to record over the next few months, specified precisely by their start and end time in seconds from the start of this new year. What is the minimum number of PVRs required to ensure every show gets recorded?

Note that a PVR is not able to instantly switch from one show to the next. It takes one second to do this. So if it stops recording a show that ends at second t then any other show it records must start at second $t + 1$ or greater.

Input

The first line of input is a single integer n indicating the number of shows you want to watch. Then n lines follow where the ith such line contains two integers s_i, t_i indicating the start and end times of a show.

Bounds: $1 \leq n \leq 100,000$ and $0 \leq s_i < t_i \leq 31,536,000$.

Output

Output consists of a single integer b that is the minimum number of PVRs required to ensure all shows are recorded.

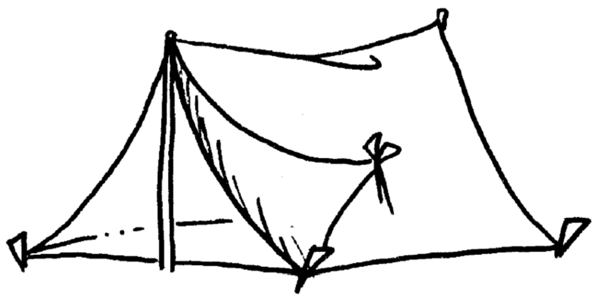
Sample Input	Sample Output
4 0 5 2 10 8 14 11 14	2

Sample Input	Sample Output
2 0 1 1 2	2

Sample Input	Sample Output
7 0 1 2 3 4 5 6 7 0 3 4 7 0 7	3

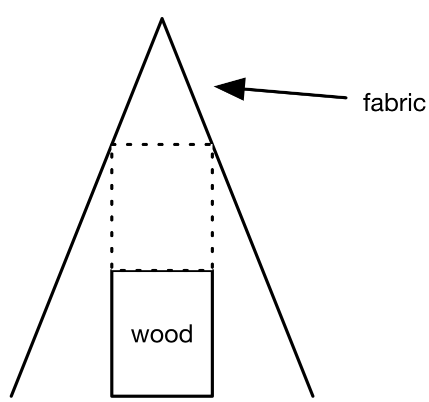
Problem G
Bob's Burgers
Problem ID: bob
Time Limit: 3 seconds

Bob is taking his friends and family to a camp out and barbeque. Unfortunately for Bob, the weather station is forecasting heavy rain on the day of their planned barbeque. Bob already spent all his money on food and can't afford to buy rain protection for the barbeque. So, Bob plans to get the biggest sheet of waterproof fabric he can find and to hoist a tent with a triangular cross section over his barbeque.



Bob will build a rectangular grill under the tent and fill it up completely with wood. The more wood he can burn, the faster the food will cook. Wood will not be stacked higher than the grill, and the grill will be flat on top (you don't want burning wood rolling off the grill and into the tent). In addition, Bob can't build the grill up to touch the tent or the tent will burn! The rule of thumb is you must leave space above the grill equal to height of the grill to prevent anything from burning. Help Bob figure out how fast he can cook his food.

Below is the a sketch of a cross-section of the tent in a valid solution. Notice that the height between the top corners of the grill and the point of the tent directly above the corners is equal to the height of the grill itself. Your goal is to create a tent from the a rectangular sheet of fabric of a given size, in order to maximize the total volume of the grill underneath the tent.



Input

Input consists of a single line containing two integers L, W representing the side lengths of the rectangular sheet of fabric, in mm. These will be separated by a single space. Note both L and W will lie between 1 and 10^9.

Output

Output the maximum volume of wood that Bob can burn in his tent in mm^3. This should be output with precisely six decimal places of precision even if the answer is an integer. You should consider the actual grilling surface on top of the grill to have height 0 (it is a very flat cast-iron grill so we can disregard its height).

Hint: You may consult the language reference for your preferred programming language to see how to print a particular number of decimal places in fixed-point format.

Sample Input	Sample Output
`100 100`	`31250.000000`

Sample Input	Sample Output
`3 11`	`11.343750`

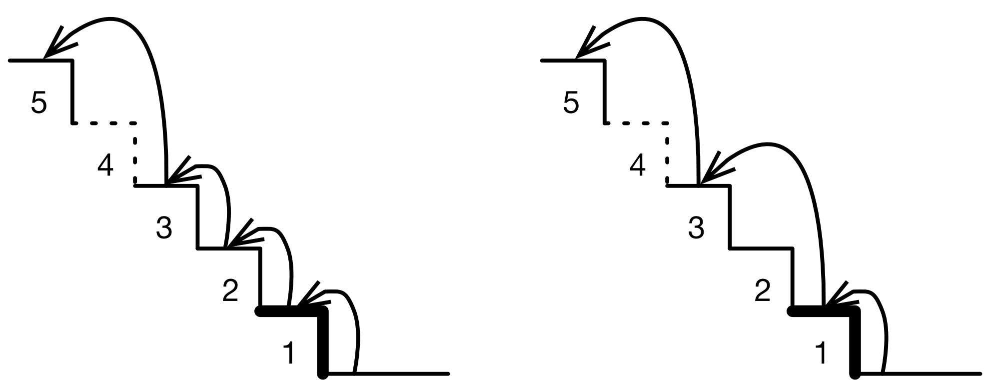
Problem H
Emma
Problem ID: emma
Time Limit: 5 seconds

I hate stairs! So let's make them interesting. Suppose I can climb stairs by stepping up either one or two steps at a time. How many different ways are there to reach the top?

Well, this might be a bit on the easy side so let's make it even more interesting. Some steps are broken: I cannot step on them for fear of falling through. Some steps are essential! For some reason, I have to step on them.

There are n steps, numbered 1 through n. My goal is to end up exactly on step n and I start just below the first step (one could think of this as step 0).

The picture below depicts the first sample input. There are precisely two possible ways to get to the top. Notice both ways ignore step 4 as it is broken, and both ways use step 1 as it is required.



Input

The first line of input contains three space-separated integers n, b, m, denoting the total number of steps, the total number of broken steps, and the total number of essential steps, respectively. The second line contains b distinct space-separated integers indicating broken steps. The third line contains m distinct space-separated integers indicating which steps are essential. Each step on these last two lines is represented by an integer between 1 and n.

Bounds: $1 \leq n \leq 20,000$, and $1 \leq b \leq n, 1 \leq m \leq n$.

Output

Let k denote the number of ways to climb these n steps where each essential step is stepped on yet no broken step is stepped on. The output should consist of a single line with the value k mod $1,000,000,007$.

As a special case, if there is no way to reach the top (i.e. k is 0 before reducing mod $(10^9 + 7)$) then you should instead output `Impossible` on a single line.

Sample Input	Sample Output

`5 1 1`
`4`
`1`
	`2`

Sample Input	Sample Output

`10 1 1`
`5`
`5`
	`Impossible`

Sample Input	Sample Output

`10 5 3`
`1 3 5 7 9`
`2 4 6`
	`1`

Problem I
A Friend of My Friend
Problem ID: rc
Time Limit: 10 seconds

Consider a group of people numbered from 1 to n. Think of these values as ranking the people some way. Then say the people are **hierarchically well-connected** if the following holds.

Consider any three people, let's call them A, B, C. If person A is friends with both persons B and C **and** both A < B and A < C, then B is also friends with C.

Input

The first line of input contains a single integer $1 \leq T \leq 20$ indicating the number of groups of people to process (i.e. the number of test cases).

Each test case begins with a line containing two integers n and m. Here, $1 \leq n \leq 100,000$ and $0 \leq m \leq 200,000$. This means there are n people in the group and m pairs of them are friends.

Following this are m lines, each containing two integers $1 \leq i, j \leq n$ indicating person i and person j are friends.

You are guaranteed no pair of friends is specified more than once in the input and that $i \neq j$ for every pair. On every line, the integers are separated by a single space.

Finally, the total number of people over all test cases is at most $300,000$ and the total number of pairs of friends that are specified over all test cases is at most $600,000$.

Output

For each group, output a single line with `YES` if the group is hierarchically well-connected and output `NO` otherwise.

Sample Input	Sample Output
2 5 4 1 2 2 5 1 5 3 4 3 2 1 2 1 3	YES NO

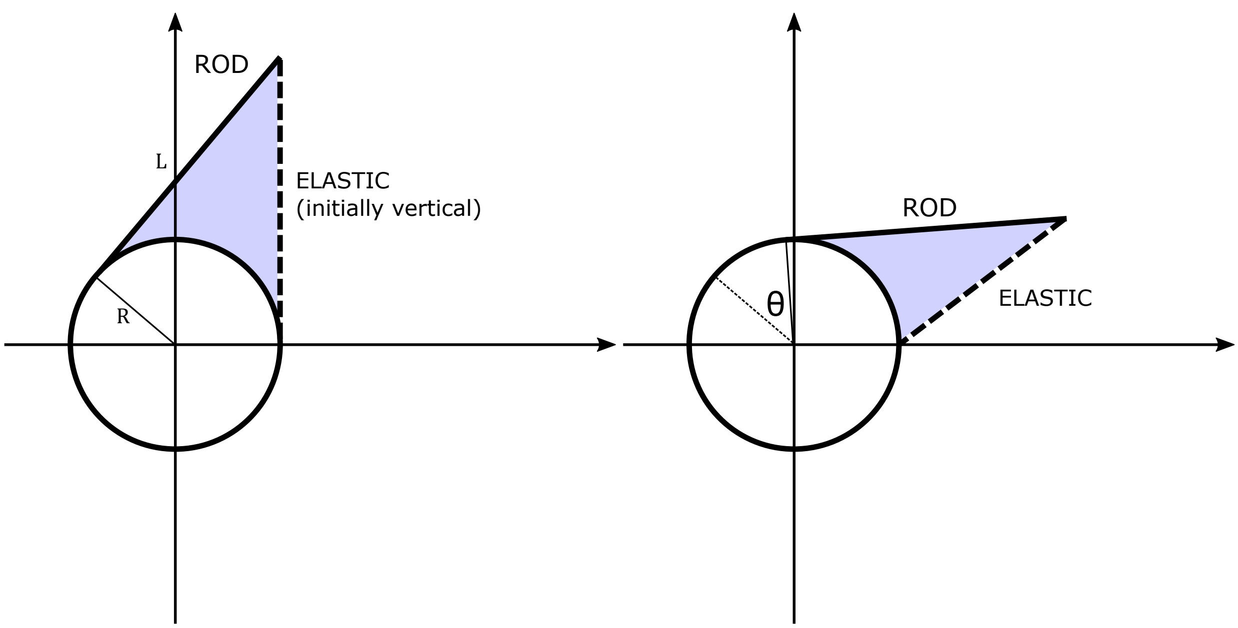
This page is intentionally left blank.

Problem J
Modern Art
Problem ID: modernart
Time Limit: 3 seconds

An artist has been commissioned to create some modern art, so she comes up with the following inspired design. Start with a circular base with radius R centred at the origin, and a rod of length L that lies perfectly tangent to some point on the top half of the circle at its left endpoint. Then, affix an elastic between the point $(R, 0)$ and the other endpoint of the rod. The circular base is rotated such that the elastic is perfectly vertical. Finally, the area bounded by the rod, the elastic, and the circular base is painted, as shown in the image on the left below.

However, she has enlisted your help as a programmer because she has a problem: she does not have the exact amount of paint needed to cover the area! Of course, she needs enough paint to cover the area, but she also does not want to waste any paint by having less area than can be covered by her supply. Her compromise is to allow a rotation of the entire structure so that the painted area changes. She will only tolerate **clockwise** rotations, and the right endpoint of the rod **must** remain above the x-axis. Your task is to help her determine an angle of rotation such that the enclosed area can be covered **exactly** by the supplied paint, if possible. The image on the right below shows the resulting design after a legal rotation by angle θ.

Note that one end of the elastic is affixed to the exact point $(R, 0)$ even after rotation, and it does not move with the base. Furthermore, the elastic always retains tension irrespective of rotation, and never falls slack.



Input

The input consists of a single line, containing three numbers R, L, A, where $0.0 < R \leq 100.0$, $R < L \leq 10000.0$, and $A > 0.0$. These denote the radius of the circular base in cm, the length of the tangential rod in cm, and the area that the artist can cover with the current paint supply in cm^2.

Output

Output consists of a single value, giving the angle θ, in rad, by which the artist should rotate the circular base clockwise, subject to the aforementioned constraints, so that she can perfectly cover the desired area with the paint supply that she has.

If multiple angles of rotation can satisfy this condition, output the smallest. If no rotation under the constraints can satisfy this condition, simply output **impossible** (with no punctuation).

Your answer will be considered correct if it is within a relative or absolute error of 10^{-6}.

Sample Input	Sample Output
`10.0 27.0 111.578533738495551`	`0.8`

Sample Input	Sample Output
`1.0 1.0 0.2`	`0.2532742561`

Sample Input	Sample Output
`1.0 1.0 0.05`	`impossible`

Problem K
Rectangle Coverage
Problem ID: rectarea
Time Limit: 10 seconds

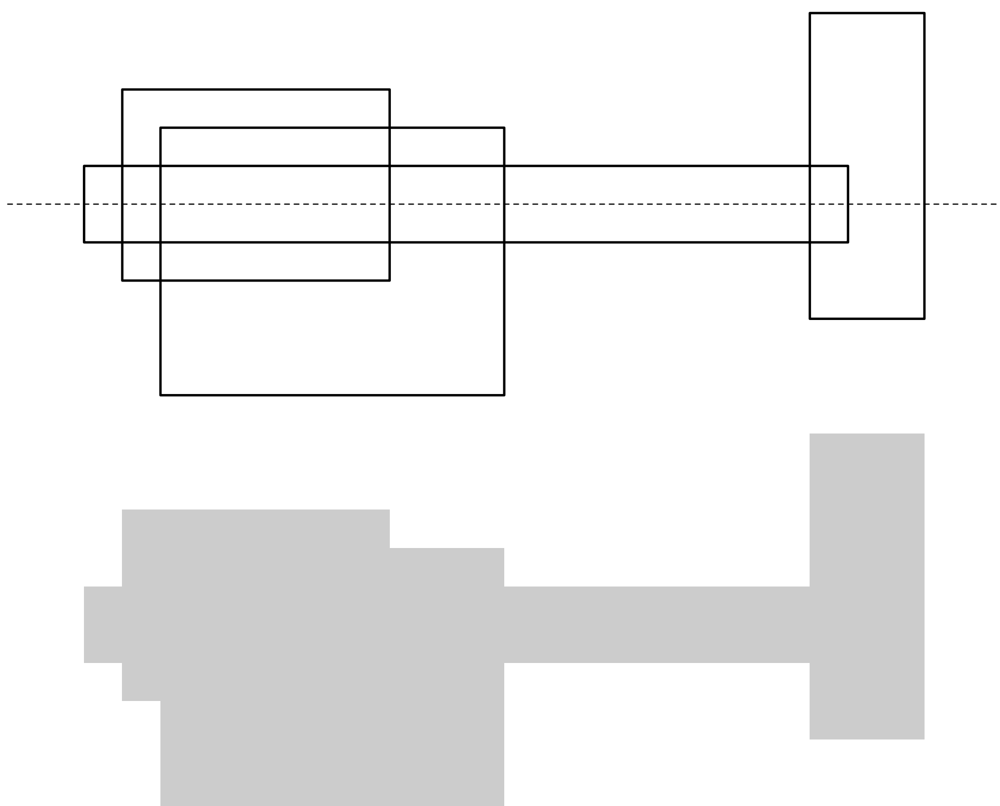
It is time for more geometry!

You know that the area of a rectangle with side lengths w and h is $w \cdot h$. Computing the area of the intersection of two rectangles is slightly more annoying, but not so bad: their intersection (if not empty) forms a rectangle and this can be easily computed too.

But what about the area of the union of two rectangles. More annoyingly, what about the area of the union of many rectangles!

This is what you have to compute in this problem. But we'll make it slightly easier: the top edge of each rectangle lies no lower than the x axis and the bottom edge of each rectangle lies no higher than the x axis.

The picture depicts the second example below. The dashed line is the x-axis. The solid image below indicates the total area that has to be calculated for this problem.



Input

The first line of input consists of a single integer $1 \leq n \leq 10^5$ indicating the number of rectangles. Then n lines follow, each describing a rectangle. Each rectangle is given by four space-separated integers x_1, y_1, x_2, y_2 where:

- $-10^9 \leq x_1 < x_2 \leq 10^9$
- $-10^9 \leq y_1 < 0 < y_2 \leq 10^9$

This means the bottom-left corner of the rectangle is at (x_1, y_1) and the top-right corner of the rectangle is at (x_2, y_2).

Output

Output consists of a single line with a single integer indicating the total are covered by at least one rectangle.

Sample Input	Sample Output
2 -1 -1 1 2 -1 -1 2 1	8

Sample Input	Sample Output
4 -10 -1 10 1 -9 -2 -2 3 -8 -5 1 2 9 -3 12 5	116

Problem L
Generating Minimum Rotations
Problem ID: genmin
Time Limit: 10 seconds

A cyclic rotation of the string $s_0 s_1 \dots s_{k-1}$ is a string we can obtain by repeatedly moving the leftmost character to the right of the string.

For example, the cyclic rotations of `BCAB` are

`BCAB`, `CABB`, `ABBC`, `BBCA`

For another example, the cyclic rotations of `ABCABC` are simply

`ABCABC`, `BCABCA`, `CABCAB`

Say that a string is *canonical* if it is lexicographically smallest among all of its cyclic rotations.

So `ABBC` and `ABCABC` are canonical wherease `BCAA` and `CABCAB` are not canonical.

Your task is the following. Given two integers n and k you should print all canonical strings of length exactly k that use the first n uppercase characters of the alphabet (i.e. the first n characters among `A`, `B`, `C`, etc.)

Input

The input consists of a single line containing the two integers n, k. Here, $1 \leq n \leq 26$ and $1 \leq k \leq 10^4$.

Output

Output consists of all length-k canonical strings using the first n letters of the alphabet (`A` through `Z`). These should be printed in lexicographic order.

Note, you are also guaranteed in that the total number of characters of all canonical strings will not exceed 10^7 for any given test case.

Sample Input	Sample Output

`2 2`	`AA`
	`AB`
	`BB`

Sample Input	**Sample Output**
3 4	AAAA AAAB AAAC AABB AABC AACB AACC ABAB ABAC ABBB ABBC ABCB ABCC ACAC ACBB ACBC ACCB ACCC BBBB BBBC BBCC BCBC BCCC CCCC