Contest Problems

Philadelphia Classic, Fall 2024

University of Pennsylvania

Authors:

Nicholas Tarsis, Jason Liu, Kurtis Zhang, Michael Gao

Clarifications:

`https://rebrand.ly/pcl24FAclarifications`

Classic Competition:

`https://rebrand.ly/pcl24FAclassic`

Advanced Competition:

`https://rebrand.ly/pcl24FAadvanced`

Testers:

Aaron Mei (meiron03)
Haokai Ma (vgoofficial)
James Leung (thehunterjames)
Frank Wong (frank-cywong)

Cover Art:

Isabella Wang

Rules and Information

This document includes 12 problems.

Classic teams do problems 1-8; Advanced teams do problems 5-12.

Any team which submits a correct solution for any of problems 1-4 will be assumed to be a classic team. **If you are not a classic team, please skip problems 1-4.**

Problems 1-4 are easier than problems 5-8, which are easier than problems 9-12. The questions are ordered *roughly* by difficulty, but you may find that it is easier to solve them out of order.

You may use the Internet only for submitting your solutions, reading Javadocs, C++ or Python documentation, and referring to any documents we link you to. You may not use the Internet for things like StackOverflow, Google, AI assistance or outside communication. Please do not use ChatGPT.

You are responsible for handling Input and Output on your own.

You will receive 1 point per correct submission. In the case of a tie, the 'total penalty' will be used as a tiebreaker, which is defined to be the sum of time elapsed from the beginning of the contest to the first accepted submission for each problem, plus a 20-minute penalty for each previously rejected run for that problem. Wrong Answer, Time Limit Exceeded, and Runtime Errors all contribute towards the penalty. Unsolved problems do not contribute towards the penalty. There is no partial credit.

Some problems use Java's `long` type; if you are unfamiliar with them, they're like an `int`, but with a (much) bigger upper bound, and you have to add `L` to the end of an explicit value assignment:

```
long myLong = 1000000000000L;
```

Otherwise, the `long` type functions just like the `int` type.

In C++, the equivalent type is `long long` for large integers. You do not need to append anything to the value, but ensure that you use `long long` instead of just `int` to avoid overflow:

```
long long myLong = 1000000000000;
```

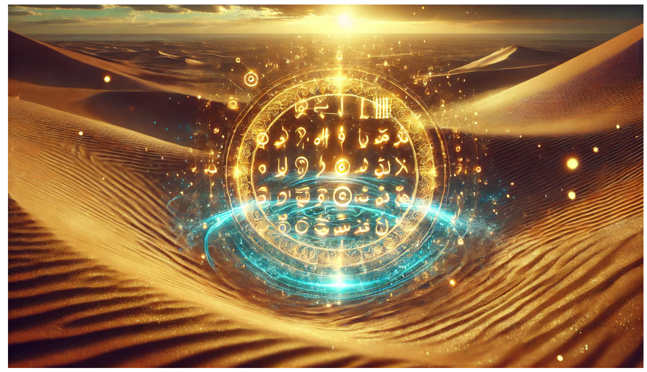
Introduction

The journey begins in a land scorched by the relentless sun and shrouded in mirages. Legends speak of a grand, hidden pyramid buried beneath shifting sands—a relic of an ancient civilization that guarded treasures and untold secrets.

A few brave souls, known as the "Enigma Explorers", have dared to search for it, navigating treacherous dunes, cryptic hieroglyphs, and shifting sandstorms. You decide to become an "Enigma Explorer" to find fame and fortune. Only those with the sharpest minds will survive the trials of the desert and the pyramid, earning the chance to claim the ancient treasure.

Problem A. Mysterious Mirage Message

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



You come across a shimmering vision that contains directions to your destination. However, the mirage is elusive, only revealing fragments of a cryptic message each time you look at it. The message is an ancient string s of length n. Each time you look, a portion of the message is obscured, leaving you with a fragment that gets shorter with each glimpse.

You must recover n fragments of the original string, where:

The 1st fragment reveals the entire string.

The 2nd fragment reveals all but the last character.

And so on, such that the i-th fragment reveals all except the last $i - 1$ characters.

Input

The first line contains n $(1 \leq n \leq 100)$ $-$ the length of the string.

The second line contains the string s of length n. s consists of lowercase and uppercase English alphabet letters (a-zA-Z).

Output

On the $(i + 1)$-th line, output the i-th string. The first line will be s with 0 characters missing, the second line will be s with the last character missing, and so on.

Examples

standard input	standard output

`8`
`Pclassic`
	`Pclassic`
	`Pclassi`
	`Pclass`
	`Pclas`
	`Pcla`
	`Pcl`
	`Pc`
	`P`

`1` `E`	`E`

Note

In the first example, the first line is the original string. Each subsequent line has an additional character missing from the end.

In the second example, the first line is the original string. Since $k = 1$, only one string should be printed.

Problem B. Camel Cargo Comparison

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

After deciphering the mirage, you make your way deeper into the desert towards its coordinates. You realize that it points towards an ancient town. It is believed that the inhabitants of this town trade with all those that enter, and those that enter will leave with a much needed item.

After several days of travel, you are beginning to run out of supplies. Luckily, you have spotted the town! You start to fill up on all the necessities for the rest of your journey, until you suddenly spot a bunch of camels. A camel would be a massive asset for the rest of your journey.

However, the town's animal master is cautious and will only lend camels to those who can complete his test. He presents you with a list of n ancient weights and asks you to measure how far each weight is from a weight k that represents a camel's ideal load. To earn the camels, you must rank these weights in order of their absolute difference from the ideal weight.

Input

The first line contains n $(1 \leq n \leq 5 * 10^4)$ − the number of weights, and k $(0 \leq k \leq 10^6)$ − the ideal weight, separated by a space.

The second line contains n integers $w_1, ..., w_n$ separated by spaces, where w_i $(-10^6 \leq w_i \leq 10^6)$ is the i-th weight.

Output

The absolute difference in weights $d_1, ..., d_n$ from k separated by spaces, in order of the difference. d_1 is the difference of the weight closest to k, d_2 is the next closest, and so on.

Examples

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

Note

In the first example, the absolute difference of each w_i and k is simply the absolute value, since $k = 0$. The output consists of these sorted absolute differences.

In the second example, the absolute differences from the ideal weight are as follows: $|-1-4| = 5$, $|3-4| = 1$, $|4-4| = 0$, $|6-4| = 2$, $|10-4| = 6$. The output consists of these sorted differences.

In the third example, the absolute difference from the ideal weights are: $|3-3| = 0$, $|3-(-3)| = 6$. The output consists of these sorted differences.

Problem C. Desert Digit Discovery

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

After solving the riddle provided by the animal master, you are congratulated by the town's leader. He is a wise elder whose knowledge of the desert's secrets has guided travelers for generations. He knows the hidden paths that lead to the great pyramid, but he does not give directions freely. "The journey is perilous," he warns. "Beyond the sands lie oases, and beyond them, treasures lost in time."

To prove your worth, you must demonstrate mastery over an ancient numerical test. The elder recounts tales of a legendary oasis halfway to the pyramids, one where the ancestors of the town left instructions carved in stones—a map to the pyramid itself. This symbol, the elder tells you, is said to hold the greatest value in guiding travelers through the harshest terrains.

The elder hands you a pair of a number n and base b marked in desert runes from a long-forgotten civilization. "Show me that you can find the largest digit in the civilization's base-b representation of n, and I will trust you with the coordinates to the oasis."

Input

The first line contains t $(1 \leq t \leq 100)$ − the number of test cases.

The next t lines each contain a single test case: an integer n $(1 \leq n \leq 10^9)$ − the number, and b $(2 \leq b \leq 10)$ − the base, separated by a space.

Output

The largest digit in the base-b representation of n.

Examples

<table>
<tr><th>standard input</th><th>standard output</th></tr>
<tr><td>1
789 10</td><td>9</td></tr>
<tr><td>1
121 4</td><td>3</td></tr>
<tr><td>2
11 10
22 10</td><td>1
2</td></tr>
</table>

Note

In the first example, 789 in base-10 (denoted 789_{10}) is just 789, so the largest digit is 9.

In the second example, 121_{10} is 1321_4, so the largest digit is 3.

In the third example, the two numbers are already in base-10, so the output is 1 and 2, on separate lines.

Problem D. Python Prime Procedure

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

As you continue your journey through the desert, the heat grows more oppressive and the sand shifts unpredictably beneath your feet. You find a shaded spot near a large rock outcrop to rest, weary from your travels. The promise of the halfway oasis fills your dreams as you close your eyes for a brief moment of respite.

Suddenly, in the dead of night, you feel a cold pressure coil around you. You open your eyes to find a massive desert snake, its scales glistening under the moonlight, constricting around your body. Its venomous fangs hover dangerously close, and it hisses ominously. The snake's eyes glow with a strange, knowing intelligence.

You struggle to break free, but the snake's grip tightens. As your strength wanes, a faint voice whispers from within the sands. "Solve the riddle of the serpent to escape its grasp. Find how many positive integers less than n share only one prime factor with n."

You realize that solving the riddle will save your life and grant you passage to the oasis that leads to the pyramid. The snake hisses again, waiting for your answer.

Input

The first line contains one integer t $(1 \le t \le 10^4)$ $-$ the number of test cases.

The next t lines each contain one integer n $(1 \le n \le 10^5)$.

It is guaranteed that the sum of n across all test cases does not exceed 10^5.

Output

For each test case, output one integer $-$ the number of positive integers less than n that share one prime factor with n.

Example

standard input	standard output
2 12 3	6 0

Note

In the first test case, the prime factors of 12 are 2 and 3. The numbers that share a factor of 2 with 12 are 2, 4, 8, and 10. The numbers that share 3 are 3 and 9. Thus, the total number of integers is 6. Note we do not count 1.

In the second test case, no positive integers less than 3 share a prime factor with it.

Problem E. Supply Sequence Search

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

At the halfway point oasis, you are finally able to catch your breath. But as you step into the clearing, you notice another figure ahead. It's another traveler, someone who shares the same path as you—a fellow Enigma Explorer. Their eyes narrow as they spot you. "So, you've made it to the oasis too," they say cautiously. The other explorer introduces themselves as Ramses, and as the conversation unfolds, you realize that the oasis is not a single oasis. It's a series of smaller oases connected by hidden paths, each one offering supplies and resources needed for the journey ahead.

However, Ramses doesn't fully trust you. In an attempt to gauge your worth, Ramses sets a test: "You say you understand the landscape of this place, but do you understand how to gather the most supplies from these oases? I'll give you a problem. Solve it, and I'll trust you to help gather the resources we both need to continue."

He continues on, "Let's say I give you a series of n oases. The i-th oasis provides s_i units of supplies. You can only move to an oasis with a higher index (from i to j, where $i < j$), and for every such move, you collect $(j - i) * s_i$ units of supplies. Your task is to find the maximum number of supplies you can gather by making the best possible moves."

Input

The first line contains a single integer n $(1 \leq n \leq 10^5)$ $-$ the number of oases.

The second line contains n integers $s_1, ..., s_n$ separated by spaces, where s_i $(0 \leq s_i \leq 10^4)$ represents the units of supplies at oasis i.

Output

A single integer representing the maximum number of supplies obtainable.

Examples

standard input	standard output
6 3 4 2 1 5 3	20
5 3 1 2 3 6	12
1 10	0

Note

In the first example, you can travel from oasis 1 to oasis 2 and gain $3*(2-1) = 3$ units of supplies. Then travel from oasis 2 to 5 and gain $4*(5-2) = 12$ units. Finally, travel from oasis 5 to 6 and gain $5*(6-5) = 5$ units. Total $3 + 12 + 5 = 20$ units of supplies.

In the second example, travel from oasis 1 to 5 and gain $3*(5-1) = 12$ units of supplies.

In the third example, it is impossible to leave oasis 1 to go to another higher-numbered oasis, so you cannot gain any supplies.

Problem F. Cursed Cactus Crossing

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

As you and your fellow traveler rest at the oasis, you come upon a strange stone tablet, half-buried in the sands near the water's edge. It's covered in ancient symbols, and the inscription speaks of a forgotten path guided by cacti that winds through the desert, stretching out beyond the horizon. This path is infinite, but fraught with hidden dangers. The inscription reads:

"Begin at cactus $i = 0$, the first step of the journey. From there, you may move forward by a step size j to cactus $i + j$. j starts at 1 and increases by 1 with every move. Beware of the n cursed mirages that have misled many travelers before you. These mirages are marked by cacti at indices m_l, and crossing them would lead you astray to your deaths. You must choose your steps carefully, for the journey cannot take more than k moves. The sands are unpredictable, and the path ahead is uncertain, but those who navigate wisely will find their way to the ancient treasures buried deep within the desert."

Your task is to determine the furthest cactus you can reach without falling victim to any of the mirages.

Input

The first line contains two integers separated by a space: n $(1 \le n \le 10^5)$ $-$ the number of mirages, and k $(1 \le k \le 10^9)$ $-$ the maximum number of steps you can take.

The second line contains n integers $m_1, ..., m_n$ separated by spaces, where m_l $(1 \le m_l \le 10^{18})$ represents that mirage l is at cactus m_l.

Output

A single integer representing the furthest cactus you can reach without visiting any mirage cacti.

Examples

standard input	standard output
`4 9` `7 15 18 12`	`10`
`1 1` `1`	`0`
`3 1` `2 3 4`	`1`

Note

In the first example, on step $j = 1$, move from $i = 0$ to $i = 1$. On the next step, $j = 2$ and move to $i = 3$. Then $j = 3$ and move to $i = 6$, $j = 4$ and move to $i = 10$. At step $j = 5$ you could move to $i = 15$, but cactus 15 is marked as a mirage, so you must stop at $i = 10$ and forfeit your remaining moves.

In the second example, on step $j = 1$, you could move from $i = 0$ to $i = 1$, but cactus 1 is a mirage, so you must stop before making any moves.

In the third example, on step $j = 1$ move to $i = 1$. At this point, the allowed $k = 1$ steps have been completed, so the highest reachable cactus is 1.

Problem G. Sandstorm Speed Survival

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

After traveling as far as you can through the path of cacti, you and Ramses continue your journey towards the pyramids. The landscape begins to shift, and a constant storm of swirling sand blocks your view. As you look ahead, the desert reveals an ancient, almost mystical part of the terrain. The sand dunes here are massive and stretch endlessly.

"The ancients used to call this place The Velocity Dunes," Ramses shouts, his voice still barely audible over the howling wind. Ramses pauses and studies the shifting sands around you. "To move through this land, you need to find the right starting velocity. Before traveling across each sand dune, the wind may either speed you up or slow you down. Your velocity must always stay positive; if it drops to zero or less even for a moment, you will be blown away!"

There are n sand dunes, and you know that dune i will change your velocity by d_i (positive or negative). You may use your strength to double your velocity and overcome the windiest dunes at most once, if necessary. To save your energy, find the smallest speed you must start with to safely traverse all the sand dunes while maintaining positive speed at all times.

Input

The first line contains a single integer n $(1 \leq n \leq 10^5)$ − the number of sand dunes.

The second line contains n integers $d_1, ..., d_n$ separated by spaces, where d_i $(-10^9 \leq d_i \leq 10^9)$ represents the velocity change that results from crossing the i-th sand dune.

Output

A single integer representing the minimum starting speed to maintain positive velocity at all times.

Examples

standard input	standard output
3 1 2 -2	1
3 -2 -3 -5	6
3 -2 5 -10	3

Note

In the first example, if you start with velocity $v = 1$, your velocity before each of the dunes will be 2, 4, and 2, respectively. Note that you cannot move if your starting speed is 0. You do not need to double your speed in this situation.

In the second example, if you start with velocity $v = 6$, you may double it at the start to reach $v = 12$, and your speed before each dune will be 10, 7, and 2, respectively. Note that if you start with $v = 5$ and doubled it at the start, your speed before the third dune would be 0, and you would be blown away.

In the third example, start with veloctiy $v = 3$. It changes to $v = 1$ before the first dune and $v = 6$ before the second dune. At this point, double it to $v = 12$ so your speed before the third dune is $v = 2$ and positive.

Problem H. Cursed Colliding Carts

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	32 megabytes

The air is still heavy with the scent of sand, and the sun is high in the sky as you and Ramses finally approach the entrance of the legendary pyramid. But the path to the treasure is blocked by a strange barrier. Near the entrance lies a rectangular grid of n rows and m columns, where each cell is marked and some of the grid cells contain ancient minecarts.

According to ancient construction codes, a grid cell marked with a 0 is empty, a 1 means the cell has a cart which only moves right across a row, and a 2 means the cell has a cart which only moves down along a column. Carts move in a single, constant direction. They move in sync, at the same pace of one stone slab per unit time. As they travel, though, they could crash into one another. If 2 carts crash, the curse of the pyramid causes them to instantly disintegrate into sand, so other carts cannot crash into them in the future. Carts which never collide will exit the grid.

The mechanism demands a precise amount of payment in gold coins—which you are fortunate enough to still have—equal to the number of carts which will eventually exit the grid (and successfully deliver their cargo). You must find this number to pass the barrier.

Input

The first line contains two integers separated by a space: n ($1 \le n \le 10^3$) and m ($1 \le m \le 10^3$) − the number of rows and columns, respectively.

The next n lines each contain m integers separated by spaces, representing the grid: $G_{i,1}, ..., G_{i,m}$, where $G_{i,j}$ represents the cell at the i-th row and j-th column. $G_{i,j} = 0$ means the grid cell is empty, 1 means it has a cart that can only move right, and 2 means it has a cart that can only move down.

Output

A single integer, representing the number of carts which will eventually exit the grid without being vaporized.

Examples

standard input	standard output
2 3 0 2 0 1 0 1	1
3 3 0 0 2 1 1 2 1 1 2	3

Note

In the first example, the cart at row 1, column 2 (denoted $(1,2)$) moves down, and the cart at $(2,1)$ moves right, so they will crash after 1 time step. The cart at $(2,3)$ moves right and exits the grid unobstructed.

In the second example, the initial configuration at time step $t = 0$ is this (with cart movement directions represented by arrows):

		↓
→	→	↓
→	→	↓

At time $t = 1$, the cart at (3,3) has exited, and 2 pairs of carts crash (represented by "x") and are all vaporized. The two right-moving carts at (2,1) and (3,1) each move right:

	→	x
	→	x

At time $t = 2$, the two remaining carts continue to move right:

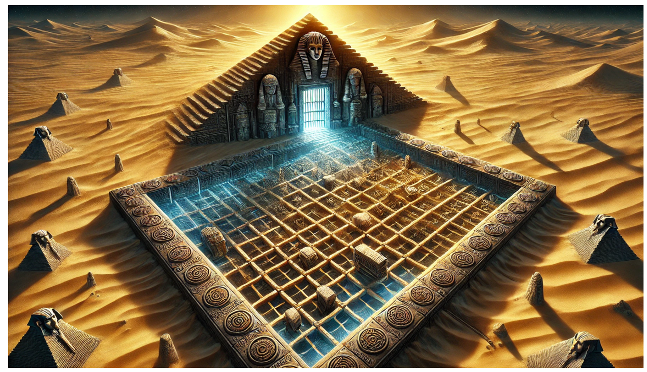
		→
		→

At time $t = 3$, the two remaining carts exit the grid.

In this situation, three cars exit the grid without being vaporized.

Problem I. Ramses' Resilient Ruin

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	32 megabytes



Just after you find the answer and move to deposit your gold coins, your clumsy friend Ramses "accidentally" trips and presses a hidden button, causing all the carts to begin moving. Something has gone wrong with the curse though, because now, when carts crash in the grid, they are not vaporized. Instead, they are stuck in the crash site and pose an obstacle to other carts, which will also crash if they enter the cell of the crash site after the collision occurred. Carts which never collide will still exit the grid as normal, though.

The payment required has now changed—you must now account for the persistence of crashed carts. However, everything else is the same: Some carts are in a rectangular grid of n rows and m columns. A grid cell marked with a 0 is empty, a 1 means the cell has a cart which only moves right across a row, and a 2 means the cell has a cart which only moves down along a column. Carts move in a single, constant direction, and in sync, at the same pace of one stone slab per unit time.

How many carts will eventually exit the grid now?

Input

The first line contains two integers separated by a space: n $(1 \le n \le 10^3)$ and m $(1 \le m \le 10^3)$ − the number of rows and columns, respectively.

The next n lines each contain m integers separated by spaces, representing the grid: $G_{i,1}, ..., G_{i,m}$, where $G_{i,j}$ represents the cell at the i-th row and j-th column. $G_{i,j} = 0$ means the grid cell is empty, 1 means it has a cart that can only move right, and 2 means it has a cart that can only move down.

Output

A single integer, representing the number of carts which will eventually exit the grid without crashing.

Examples

standard input	standard output
2 3 0 2 0 1 0 1	1
3 3 0 0 2 1 1 2 1 1 2	1

Note

In the first example, the cart at row 1, column 2 (denoted $(1, 2)$) moves down, and the cart at $(2, 1)$ moves right, so they will crash after 1 time step. The cart at $(2, 3)$ moves right and exits the grid unobstructed.

In the second example, the initial configuration at time step $t = 0$ is this (with cart movement directions represented by arrows):

		↓
→	→	↓
→	→	↓

At time $t = 1$, the cart at (3,3) has exited, and 2 pairs of carts crash (represented by "x"). The two right-moving carts at (2,1) and (3,1) each move right:

	→	x
	→	x

At time $t = 2$, the remaining 3 carts all crash into existing wreckage:

		x
		x

In this situation, only one car exits the grid unharmed.

Problem J. Light the Linked Labyrinth

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

You finally choose the correct number of coins, and the barrier dissolves to let you pass. You and Ramses step into the pyramid's echoing silence. The path ahead is a mystery, with countless dark passages branching out like veins in stone. You both know that only one route will lead to the map room, which hides the treasure's location. But the pyramid's ancient architects were cunning—the passageways are a maze. An inscription on the wall reads:

To reach the map, find the hidden pathways in the labyrinth which pass by the sacred altars. A guide through the darkness, the altars stand as your sentinels.

There are n chambers in the maze, numbered $1, ..., n$, connected by m passageways. The entire labyrinth is connected; it is possible to reach any chamber from any other chamber. There are two sacred altars, a and b, in two different chambers. To decide the best path to proceed, count the number of pairs of rooms (s, t) in the labyrinth where any path between s and t must pass through both rooms containing the sacred altars, and such that neither is itself an altar room.

Input

The first line contains four integers separated by spaces: n $(4 \le n \le 10^5)$ − the number of chambers, m $(3 \le m \le 10^5)$ − the number of passageways, and a and b − the two rooms with sacred altars. It is guaranteed that $a \ne b$.

The next m lines each contain two integers u and v separated by a space, where u, v represent two chambers which are connected by a passageway.

Output

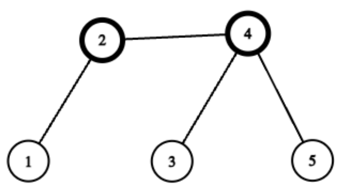
A single integer, representing the number of pairs of chambers (s, t) such that every path between s and t must pass through both a and b, and neither s nor t is a or b.

Examples

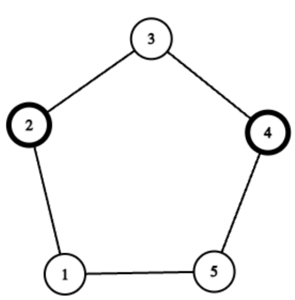
standard input	standard output
5 4 2 4 1 2 2 4 3 4 4 5	2
5 5 2 4 1 2 2 3 3 4 4 5 1 5	0

Note

In the first example, the (s, t) pairs are $(1, 3)$ and $(1, 5)$. $a = 2$ and $b = 4$ are marked in the graphic below, where circles represent chambers and lines represent connecting passageways.



In the second example, no pair of rooms among $\{1, 3, 5\}$ require that any path between the pair must go through both $a = 2$ and $b = 4$. a and b are marked in the graphic below.



Problem K. Precise Prime Patterns

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

After countless twists and turns through the shadowed corridors, you and Ramses finally reach the map room—a grand chamber lit by a single shaft of light from the stone ceiling. In the center of the room lies an ancient tablet, its surface covered in a sprawling depiction of the desert, intricate symbols, and faint lines that may hint at the key to opening the door ahead.

But as you approach the tablet, Ramses shifts uneasily, glancing around as if the walls themselves are watching. "I... don't like this place," he mutters, backing away. Despite your protests, he turns and hurries back toward the entrance, leaving you alone in the eerie stillness of the map room.

You take a deep breath, as you begin to study the tablet. Its markings are complex and cryptic, and you soon notice that an inscription is carved around its edge:

The paths to my treasures are many, yet only the worthy shall find the way. Know this: in the realm of numbers, from my vault of treasures—a list vast and diverse— You must extract the greatest hoard that obeys these rules: Exactly k primes must grace at least one, yet no prime shall reign over all. Prove your worth and reveal the length of this grandest of collections.

It dawns on you that the tablet demands something precise: find the length of the longest subarray in a given array of numbers, that satisfies the given condition: exactly k primes divide at least one of the elements and no prime divides all of the elements.

Input

The first line contains two space-separated integers, n ($1 \leq n \leq 10^5$), the length of the array, and k ($1 \leq k \leq 10^5$), the number of primes that should divide some number in a subarray.

The next line contains n space-separated integers $a_1, ..., a_n$, where a_i ($1 \leq a_i \leq 10^6$) is the i-th element of the array.

Output

Output one positive integer, the length of the longest such subarray.

Examples

standard input	standard output
5 3 2 6 5 8 14	4
4 2 3 3 3 3	0
4 4 2 3 5 7	4

Note

In the first example, the answer is 4, with one such subarray containing the first 4 elements $[2, 6, 5, 8]$. The primes $2, 3, 5$ divide at least one of the elements, but no prime divides them all.

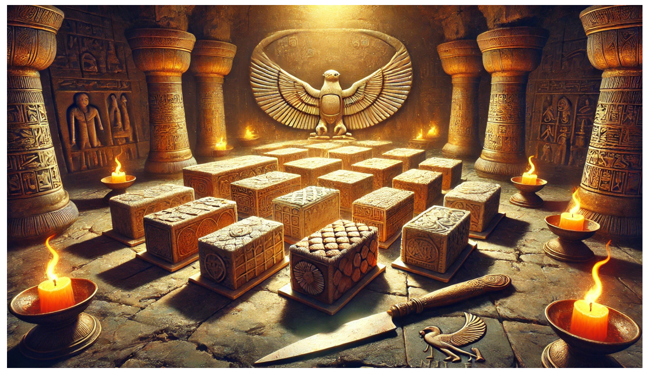
In the second example, the answer is 0 since no such subarray exists.

For the third example, the answer is 4, since the entire array is just 4 unique primes.

Problem L. Putrid Perimeter Pastries

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

After deciphering the map room and braving many trials, you arrive at the Pharaoh's tomb, and finally reach the treasure you seek: the Pharaoh's thousand-year-old birthday cakes with Nile fish icing. Due to aging, the n cakes have flattened into two-dimensional rectangles, with the i-th cake having length L_i and height H_i. Your goal is to apply icing along the perimeters of all the cakes, but your stomach grumbles as you realize you only have enough icing to cover a total perimeter of length P!



Except—you have one last tool up your sleeve: the sacred pastry knife of Aten, which makes perfectly straight slices. For each of the rectangular cakes, you can choose to slice the cake exactly once into two pieces (not necessarily rectangles) of equal area. Find the maximum amount of icing you can apply on the perimeters of the (possibly sliced) cake pieces without running out.

Input

The first line consists of two space separated integers, n ($1 \le n \le 10^5$) − the number of cakes, and P ($1 \le P \le 10^9$) − the perimeter of icing available.

The next n lines each contain two space separated integers, with the i'th line having L_i and H_i, or the length and height of the i'th rectangle. It is given that $P \ge \sum_{i=1}^{n} 2(L_i + H_i)$.

Output

Please output one number representing the maximum possible sum of perimeters. Any value that is within 10^{-6} of the correct answer will be judged as correct.

Examples

standard input	standard output
1 11 1 2	10.47213595499958
2 10 1 1 1 1	10

Note

For the first testcase, if we cut the cake along the main diagonal, we will have two right triangles each with a perimeter of $3 + \sqrt{5}$. Thus, the sum of perimeters is $6 + 2\sqrt{5}$.

For the second testcase, we can cut the first cake into two $1 \times \frac{1}{2}$ rectangles.