

A. Anti-Diagonal Game

1 second, 1024 megabytes

Alice and Bob are playing a game!

Alice and Bob both know some string S of length $N + 1$ that consists of only the characters **A** and **B**. The game is played on a $(N + 1) \times (N + 1)$ matrix, with rows and columns numbered from 0 to N . We denote the cell in row i and column j as (i, j) .

Initially, a token is placed on $(0, 0)$. Each turn,

- Suppose the token is on cell (i, j) . The turn player must move the token to either $(i + 1, j)$ or $(i, j + 1)$.

The game ends when the token reaches the anti-diagonal of the matrix, which is the set of cells $\{(i, j) \mid i + j = N\}$. In other words, there will be a total of N turns in the game. The winner of the game is determined by the string S :

- If the token ends on the anti-diagonal cell $(i, N - i)$, and $S_i = \text{A}$, then Alice wins.
- If the token ends on the anti-diagonal cell $(i, N - i)$, and $S_i = \text{B}$, then Bob wins.

Alice moves first. Among the 2^{N+1} possible strings S , how many strings are there such that Alice wins the game, assuming both players play optimally? Since the answer can be very large, output the answer modulo $10^9 + 3233$.

Input

The input contains a single integer N ($1 \leq N \leq 100\,000$).

Output

Output a single integer denoting the number of strings S such that Alice wins the game, modulo $10^9 + 3233$, assuming both players play optimally.

input
2
output
3

The three strings S such that Alice wins the game are **AAA**, **AAB**, and **BAA**.

B. Brilliance of Wings

1.5 s, 1024 megabytes

Mashiro has a tree with N vertices and $N - 1$ unweighted edges. It is known that any two vertices in the tree are connected.

Mashiro wants to perform a Morfonication on the tree. To put it simply, Mashiro will do several operations to transform the original tree to a new tree to her liking. Each operation Mashiro does is of the following:

- Select an edge from the tree and remove it. After this step, the tree becomes disconnected to two components.
- Add an edge between any two vertices such that the structure is reconnected into a tree.

After every operation Mashiro does, the tree must stay as a tree, meaning it must have exactly $N - 1$ edges and any two vertices in the tree are connected.

Mashiro has come back from practice and she wants to know the exact operations of the Morfonication. Can you show her the operations to do so in the least number of operations possible?

Input

The first line of input contains one integer N ($2 \leq N \leq 10^5$), the size of the two trees.

The second to N -th lines of the input contain two integers U_i and V_i ($1 \leq U_i, V_i \leq N, U_i \neq V_i$), the edges of the first tree.

The $(N + 1)$ -th to $(2N - 1)$ -th lines of the input contain two integers X_i and Y_i ($1 \leq X_i, Y_i \leq N, X_i \neq Y_i$), the edges of the second tree.

Output

The first line of output contains one integer denoting the minimum number of operations, K .

The next K lines contain four integers A_i, B_i, C_i , and D_i ($1 \leq A_i, B_i, C_i, D_i \leq N, A_i \neq B_i, C_i \neq D_i$) denoting an operation of removing edge (A_i, B_i) and adding edge (C_i, D_i) . The operations should follow the rules given in the problem description, otherwise a Wrong Answer verdict is given.

If there are multiple possible answers, print any of them.

input
4
1 2
2 3
3 4
3 1
4 1
2 4
output
3
4 3 2 4
3 2 1 3
2 1 1 4

C. Chimchar Defense

1 second, 1024 megabytes

In Piplup Nation, they are under attack by a horde of Chimchars. The battlefield can be represented by a line with N areas labeled 1 to N from left to right.

There are N Chimchars, with the i^{th} Chimchar currently at area i and having H_i health remaining.

To defend the nation, there are N Piplups, with the i^{th} Piplup stationed at area i . Each Piplup can unleash a powerful Hydro Pump leftward, damaging Chimchars that are on its left (including its own area). Notably, the i^{th} Piplup can fire a Hydro Pump that deals D_i damage to all Chimchars in area $j \leq i$. However, each Chimchar only takes $\min(H, D_i)$ damage from the attack, where H is the current health of the Chimchar.

Since Hydro Pump is a very powerful move, it has low PP, and each Piplup can only fire it once. Additionally, due to its low accuracy, the i^{th} Piplup requires a certain number of X-Accuracy items, costing C_i , to successfully launch the attack.

As the commander of the Piplup army, your goal is to maximize the following value:

$$(\text{Total damage dealt to Chimchars}) - (\text{Total cost})$$

Input

The first line of input contains an integer N ($1 \leq N \leq 5000$), the number of areas.

The second line of input contains N integers, H_i ($1 \leq H_i \leq 5000$) the initial health of the i^{th} Chimchars.

The third line of input contains N integers, D_i ($1 \leq D_i \leq 5000$) the attack damage of the i^{th} Piplup.

The fourth line of input contains N integers, C_i ($0 \leq C_i \leq 10^9$) the attack cost of the i^{th} Piplup.

Output
Output a single integer representing the maximum possible value of
(Total damage to Chimchars) – (Total Cost).

input
5 6 3 2 5 1 1 3 3 2 3 1 5 1 10 2
output
12

If we choose to make Piplup 3 and 5 fire, the damage dealt to the chimchars would be [6, 3, 2, 3, 1].

Thus, the total score would be $6 + 3 + 2 + 3 + 1 - 1 - 2 = 12$

D. Double String

2 s, 1024 megabytes

SoCCat has a string S consisting of lowercase English letters a to z. For their Final Year Project, SoCCat is interested in the so-called double string.

Formally, a string T is a double string if and only if:

- T has even length.
- The first half of T is equal to the second half of T . In other words,
 $T_i = T_{i + \frac{|T|}{2}}$ for all $1 \leq i \leq \frac{|T|}{2}$.

Help SoCCat count the number of pairs of indices (i, j) such that $1 \leq i < j \leq |S|$ and the substring $S[i, j] = S_i S_{i+1} \dots S_j$ is a double string!

Input
The input contains a string S ($1 \leq |S| \leq 200\,000$) consisting of lowercase English letters a to z.

Output
Output a single integer denoting the number of pairs of indices (i, j) such that $1 \leq i < j \leq |S|$ and the substring $S[i, j]$ is a double string.

input
mississippi
output
5

input
aaaaa
output
6

input
soc
output
0

In the first sample, $S = \text{mississippi}$. All the substrings that are double strings are:

- ississ, with $i = 2$ and $j = 7$.
- ssissi, with $i = 3$ and $j = 8$.
- ss, with $i = 3$ and $j = 4$.

- ss, with $i = 6$ and $j = 7$.
- pp, with $i = 9$ and $j = 10$.

Thus, the answer is 5.

In the second sample, $S = \text{aaaaa}$. All pairs of indices (i, j) such that $j - i + 1$ is even are double strings. Thus, the answer is 6.

In the third sample, $S = \text{soc}$. There are no double strings in this case, so the answer is 0.

Statement
is not
available
in
English
language

F. Fair Forgery

1 second, 1024 megabytes

The 2025 Pokemon Association Presidential Elections have just concluded! There were N candidates running for the position of president, and M votes were recieved. Each vote i is a ranking of the N candidates, which is a permutation $p_{i,1}, p_{i,2}, \dots, p_{i,N}$ of the integers from 1 to N , with $p_{i,1}$ ranked the highest, followed by $p_{i,2}$, and so on, until the lowest ranked $p_{i,N}$.

Selina is in charge of tallying the votes. However, when she checked the list of voters, she realised that there were supposed to be K voters, not M voters! In other words, some voters may have submitted more than one vote, some may have not voted, or some non-voters may have somehow cast their vote. Selina does not want to redo the election due to suspicions in legitimacy of the votes, so she decided to forge K votes to replace the M votes recieved. However, the K votes should not be too suspicious.

After some deliberation, she decided that if the K votes satisfy the following condition, then they are probably not too suspicious:

For all integers $1 \leq t \leq K$, $1 \leq l \leq N$, if a candidate appears in the highest l ranks in at least $\lceil \frac{t \cdot M}{K} \rceil$ of the M received votes, then he/she should appear in the highest l ranks in at least t of the K forged votes.

Selina proved non-constructively that she can always forge K such votes regardless of the M votes she receives. However, she does not know how to actually forge these K votes. Please help Selina with her forgery!

Input
The first line of the input contains 3 integers N, M, K ($1 \leq M \leq 10^4$, $1 \leq N, K \leq 100$, $M \neq K$).

The next M lines of the input each contain N integers $p_{i,1}, p_{i,2}, \dots, p_{i,N}$, which is a permutation of 1 to N .

Output
Output K lines, where the i^{th} line ($1 \leq i \leq K$) contains N integers $q_{i,1}, q_{i,2}, \dots, q_{i,N}$ forming a permutation of 1 to N , denoting the i^{th} forged vote.

The K votes should satisfy the condition mentioned.

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

input
10 2 4 1 4 5 2 7 6 10 9 3 8 5 6 1 8 7 3 4 9 10 2
output
1 4 5 2 7 6 10 9 3 8 1 4 5 2 7 6 10 9 3 8 5 6 1 8 7 3 4 9 10 2 5 6 1 8 7 3 4 9 10 2

For the first sample, we can verify that the provided sample output satisfies the conditions. An example of an output that does not satisfy the conditions is:

1 2 4 5 3

1 2 5 3 4

This is because we may take $t = 1, l = 3$ in the condition: in the $l = 3$ highest ranks in the given M votes, candidate 3 appears $2 \geq \left\lceil \frac{t \cdot M}{K} \right\rceil = \left\lceil \frac{1 \cdot 4}{2} \right\rceil = 2$ times, so it should appear at least $t = 1$ times in the 3 highest ranks in the output K votes. However, it appears 0 times in the 3 highest ranks in the output K votes.

G. Game of Two Choices

1 second, 1024 megabytes

You are playing a game against SoCCat!

The game is played on a directed graph $G = (V, E)$ with N vertices, numbered from 1 to N . We denote $(u \rightarrow v)$ as a directed edge from vertex u to vertex v . We denote $N(u)$ as the set of vertices such that there is an edge from u to the vertices in $N(u)$. In other words, $N(u) = \{v \mid (u \rightarrow v) \in E\}$.

Initially, a token is placed on some vertex S , and the game score is 0. Each turn,

- Suppose the token is on vertex u .
- If $|N(u)| = 0$, the game ends.
- If $|N(u)| = 1$, the token moves to the only vertex in $N(u)$. The game score increases by 1.
- If $|N(u)| \geq 2$, you must choose **two different** vertices $v_1, v_2 \in N(u)$ such that $v_1 \neq v_2$. Then, SoCCat can choose to either move the token to v_1 or v_2 . The game score increases by 1.

You would like to play the game as long as possible, so you want to maximize the game score. SoCCat would like to end the game as soon as possible, so they want to minimize the game score.

Assuming both you and SoCCat play optimally, what is the maximum game score you can achieve? If the game can be played indefinitely, output -1 . Otherwise, output the maximum game score.

Compute the above for all S from 1 to N !

Input

The first line of the input contains two integers N and M ($1 \leq N \leq 200\,000, 1 \leq M \leq 400\,000$), the number of vertices and the number of edges in the graph.

The next M lines each contain two integers u_i and v_i ($1 \leq u_i, v_i \leq N, u_i \neq v_i$), denoting a directed edge $(u_i \rightarrow v_i)$ from vertex u_i to vertex v_i .

It is guaranteed that $(u_i \rightarrow v_i) \neq (u_j \rightarrow v_j)$ for all $i \neq j$. **It is possible for the graph to be disconnected or contain cycles.**

Output

Output N integers. The S -th integer ($1 \leq S \leq N$) should be the maximum game score you can achieve when the token is initially placed on vertex S , assuming both you and SoCCat play optimally. If the game can be played indefinitely when the token is initially placed on vertex S , output -1 instead.

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

input
3 3 1 2 2 3 3 1
output
-1 -1 -1

input
6 6 2 1 3 2 4 5 5 6 6 4 4 2
output
0 1 2 2 4 3

H. Help Eevee Pls Eh

1 second, 1024 megabytes

Eevee recently learned that its name is a palindrome! This means that it reads the same forward and backward.

However, Eevee is sad to discover that not many words are palindromes. One day, Eevee was visited by a friend. Upon hearing the friend's name, Eevee became curious.

Eevee wondered how many different ways there are to remove a single character from the friend's name so that the remaining characters, when concatenated, form a palindrome.

For example, in `eevae`, removing either `v` or `a` is valid, but removing any of the `e`'s is not.

We define a palindrome as the following: Given a string $S = s_1s_2 \dots s_n$, we pair $(s_1, s_n), (s_2, s_{n-1}), \dots, (s_{\lfloor \frac{n}{2} \rfloor}, s_{\lfloor \frac{n}{2} \rfloor + 1})$. S is a palindrome if and only if in all of these $\lfloor \frac{n}{2} \rfloor$ pairs the characters in the pairs are identical.

Input

The first line of input contains an single string S ($2 \leq |S| \leq 10^6$), the friend's name consisting of lowercase English letters `a` to `z`.

Output

Output a single integer representing the number of ways to remove a character from S such that it becomes a palindrome.

input
eevae
output
2

input
helppeeveeplseh
output
1

For the string `eevae` = $s_1s_2s_3s_4s_5s_6$, if we simulate removing each character:

Removing s_1 : We have pairs $(s_2, s_6), (s_3, s_5)$, 1 of these pairs are identical.

Removing s_2 : We have pairs $(s_1, s_6), (s_3, s_5)$, 1 of these pairs are identical.

Removing s_3 : We have pairs $(s_1, s_6), (s_2, s_5)$, 2 of these pairs are identical.

Removing s_4 : We have pairs $(s_1, s_6), (s_2, s_5)$, 2 of these pairs are identical.

Removing s_5 : We have pairs $(s_1, s_6), (s_2, s_4)$, 1 of these pairs are identical.

Removing s_6 : We have pairs $(s_1, s_5), (s_2, s_4)$, 1 of these pairs are identical.

Thus, only for 2 of the characters is the number of valid pairs $\lfloor \frac{5}{2} \rfloor = 2$, thus the answer is 2.

I. Independent Inversions

4 seconds, 1024 megabytes

Along Seabreak Path, there are $N = 3^K$ rows of flowers, with each row containing exactly K flowers. In the i -th row ($0 \leq i \leq N - 1$), the color of the flowers is denoted by the ternary representation of i , with 0 representing red, 1 representing green, and 2 representing blue.

E.g., if $K = 4$ and $i = 38 = 1102_3$. Then, the flowers in the 38th (0-indexed) row are green, green, red, and blue, in that order.

Shaymin likes each row of flowers a different amount, so they assign a beauty value B_i to each row, where B is a permutation from 0 to $N - 1$.

Two rows i and j are considered independent if they do not have any identical flowers in the same column.

E.g., row $i = 3 = 0010_3$ and row $j = 38 = 1102_3$ are independent, while row $i = 3 = 0010_3$ and row $j = 22 = 0211_3$ are not independent because the first flowers are both red, and the third flowers are both green.

Two rows i and j are considered an independent inversion if:

- $i < j$
- $B_i > B_j$
- Rows i and j are independent.

Help Shaymin count the number of independent inversions in Seabreak Path!

Input

The first line of the input contains two integers N and K ($3 \leq N \leq 177\,147, 1 \leq K \leq 11$), it is guaranteed $N = 3^K$.

The second line of the input contains N integers, where the i -th (0-indexed) integer denotes B_i . It is guaranteed that B_i is a permutation from 0 to $N - 1$.

Output

Output a single integer denoting the number of independent inversions in Seabreak Path.

input
9 2 5 0 1 2 3 4 6 7 8
output
2

The 2 indepedent inversions are (0, 4) and (0, 5). Note that while (0, 1), (0, 2), and (0, 3) are inversions, they are not independent.

J. Job Interview

2 seconds, 1024 megabytes

Recently, a new company, Hyper Flex Technologies, decided to sponsor Prof. Halim's CS3233 course and, as a result, received a large influx of applicants.

An interview was held to evaluate candidates for leadership abilities. Candidates enter the room one by one, in increasing order of skill level A_i , and form a line in the room.

When a new candidate arrives, they must partition the existing candidates into contiguous groups. The rightmost (newest) candidate picks their own group first. Then, the next rightmost ungrouped candidate becomes the new leader and picks their group, continuing until all candidates are grouped.

Each group contributes a conflict factor to the total score:

conflict factor = (max A - min A) \times (group size) + C

where C is a constant, known as the chaos index. The new candidate's score, S_i , is the sum of all conflict factors across groups, and their goal is to minimize their score.

The interviewers maintain a list of potential candidates in front of the interview room. After the i -th candidate is scored, they compute:

$B_i = ((S_i + K_i) \bmod i) + 1$

and modify the list accordingly. Specifically, the B_i -th candidate (1-indexed) is either added to or removed from the list: if the candidate is already on the list, they are removed; otherwise, they are added. (No one really knows how the interviewers judge candidates. But this is the blackbox pattern observed.)

Before entering the room, each candidate checks the list and avoids selecting leaders from it. Specifically, a candidate i will not choose a group of the form $[j + 1, j + 2, \dots, i - 1, i]$ if candidate j is on the list. **Since the list is only viewed before entering the room and they do not know the updated list, different candidates may have different sets of people they avoid when choosing their group during the partition process.** (See sample explanation for better understanding.)

You are the last candidate — clearly the best — and you are anxious to know in advance what your score will be.

Input

The first line of input contains two integers N, C ($1 \leq N \leq 500\,000, 0 \leq C \leq 10^9$), the number of candidates and the chaos index.

The second line of input contains N integers, A_i ($0 \leq A_i \leq 10^9, \forall i < N, A_i \leq A_{i+1}$) the skill level of the i -th candidate.

The third line of input contains N integers, K_i ($1 \leq K_i \leq i$).

Output

Output a single integer representing your score.

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

First candidate score: 1. With a group of [1].

$B_1 = 1$, the candidate list becomes {1}, thus candidate 2 cannot form group [2].

Second candidate score: 5. With a group of [1, 2], since candidate 2 cannot choose {1} as leaders and cannot form [1], [2].

$B_2 = 1$, the candidate list becomes {}, thus candidate 3 has no restriction on what group to form.

Third candidate score: 6. With a group of [1, 2], [3]. Note that candidate 2's list remains {1} and thus we cannot form [1], [2], [3]. Since after choosing group [3], candidate 2 will choose their group and it cannot be [1, 2] as mentioned earlier.

$B_3 = 3$, the candidate list becomes {3}, thus candidate 4 cannot form group [4].

Repeating, we will get that the score of candidate 5 is 9 with [1, 2], [3, 4, 5].

K. Kanto To Johto

1 second, 1024 megabytes

Red is a Pokemon trainer who has beaten the Kanto Pokemon League, and wants to get to Johto for new challenges. To get there, he needs to travel by train.

The region has N train stations labelled from 1 to N . Ash starts at station 1 and needs to get to station N to enter Johto. There are M bidirectional train lines, where the i^{th} train line connects stations X_i and Y_i , and multiple train lines may connect the same pair of cities. **It is possible to get from city 1 to city N with the train lines, but it may not be possible to get from city 1 to some other cities with the train lines.**

To use the i^{th} train line, Red needs to pay C_i dollars for a lifetime pass, after which he may use the train line as many times as he wants. He can only purchase the pass for the i^{th} train line when he is at station X_i or Y_i , and he can only purchase it once.

The payment system is online, where Red only needs to make a delayed bulk payment for the passes he bought when he enters Johto (due to immigration policies). Note that when he reaches station N , he does not need to pay yet, and can continue taking trains. He only pays when he exits station N and reaches the immigration counter.

Due to a bug in the system, he will only need to pay for the passes with the K smallest costs (or all of them if he bought fewer than K passes) when he reaches Johto.

Help Red determine the minimum amount of money (in dollars) he needs to spend to get from Kanto to Johto.

Input

The first line of the input contains 3 integers N, M, K ($2 \leq N \leq 10^5$, $1 \leq K \leq M \leq 2 \cdot 10^5$).

The next M lines of the input each contain 3 integers X_i, Y_i, C_i ($1 \leq X_i, Y_i \leq N, X_i \neq Y_i, 1 \leq C_i \leq 10^9$).

It is guaranteed that Red can get from station 1 to station N using the train lines.

Output

Output a single integer, the minimum amount of money (in dollars) Red needs to spend to get from station 1 to station N .

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

input
7 10 5
1 2 4
2 6 3
6 4 2
3 5 7
4 7 3
3 7 1
2 4 10
1 3 12
5 7 4
4 5 4
output
12

input
3 4 2
1 2 1
1 2 1
1 2 2
2 3 2
output
2

In the first sample, note that Red cannot buy the pass for the train line connecting stations 3 and 6, as there is no way to reach those cities by train.

To get to city N , Red may buy the pass to travel between stations 1 and 2, travel to station 2, buy the pass to travel between stations 2 and 7, travel to station 7, buy the pass to travel between stations 7 and 5, travel to station 5, buy the pass to travel between stations 5 and 4, travel back to station 7 (note that he does not have to travel to station 4 after buying the pass), then go to immigration.

The passes he buys cost 3, 3, 2, 2 in that order, and he pays for the $K = 2$ cheapest, for a cost of $2 + 2 = 4$. It can be shown this is the minimum possible.

In the second sample, Red may travel $1 \rightarrow 2 \rightarrow 6 \rightarrow 4 \rightarrow 7$, buying the passes between consecutive cities in the route, and pay $4 + 3 + 2 + 3 = 12$ as he only buys 4 passes but $K = 5$. It can be shown this is the minimum possible.

In the third sample, Red may buy all the passes, and he only pays for the $K = 2$ passes between stations 1 and 2 which cost 1 each, for a total cost of $1 + 1 = 2$.

L. Last Goal

1 second, 1024 megabytes

In the 2025 Pokeathlon World Cup, the clock ticks into the 90th minute. The stadium is deafening, the tension is unbearable— Team Pingu is locked in a dead heat with their rivals. The ball is at the feet of their captain, the indomitable Piplup. He eyes the goal, calculating the perfect shot. The crowd holds its breath. He takes aim—AND STRIKES! The ball rockets forward, slicing through the air like a missile. Time slows as it hurtles toward the net... AND IT'S IN! GOOOOOOAL! The fans erupt in wild celebration as Piplup delivers the match-winning strike!

The goal area can be treated as a rectangle with walls on three of its sides. Since Piplup is incredibly strong, the ball will not lose speed upon hitting the sides of the goal but will instead perfectly reflect off the walls. You are tasked with capturing a photo of this incredible match-winning shot. To get the perfect image, you plan to stand just behind the goal. However, you are afraid that the ball might hit your \$100,000 camera. Thus, you must calculate where the ball will impact the back wall of the goal so that you can position yourself as far away as possible while still being directly behind the wall.

You are given the location where Piplup made the shot and the point at which he was aiming. (The aiming point may be behind the goal.) **Note that it is guaranteed that the ball, following a straight line, will go into the goal and eventually hit the back wall.** (As Piplup had secretly used the move Lock-On prior to taking the shot.) Please determine the farthest distance from any point on the back wall to the point where the ball will hit the back wall of the goal. (Please refer to sample explanation for a clearer picture.)

Input

The first four lines of input each contain two integers, X_i, Y_i ($-10^9 \leq X_i, Y_i \leq 10^9$), representing the goal area in counterclockwise order, with $(X_2, Y_2) - (X_3, Y_3)$ being the back wall of the goal.

The next line of input contains four integers, A, B, C, D ($-10^9 \leq A, B, C, D \leq 10^9$), indicating that the ball will start from (A, B) and move towards (C, D) until it reaches the goal.

It is guaranteed the ball is not within the goal, inclusive of the boundary. The path the ball follows is also guaranteed to hit the back of the goal. (Yes we are very nice.)

Output

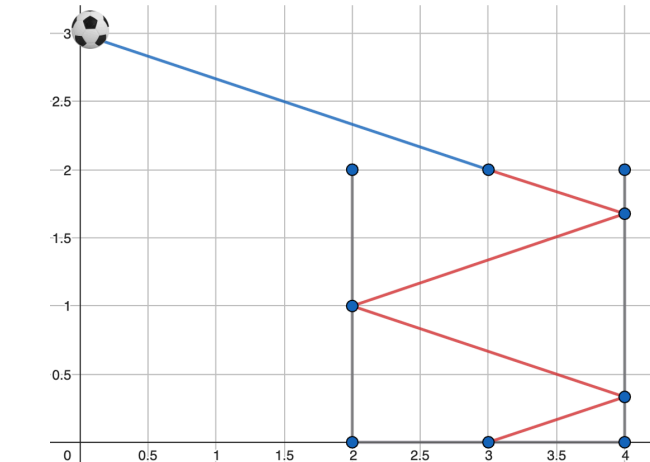
Output a single values representing the farthest distance you can be from point of impact of the ball against the back wall. Given that you are still somewhere on the back wall.

Your answer is considered correct if its absolute or relative error does not exceed 10^{-3} . Formally, let your answer be a , and the jury's answer be b .

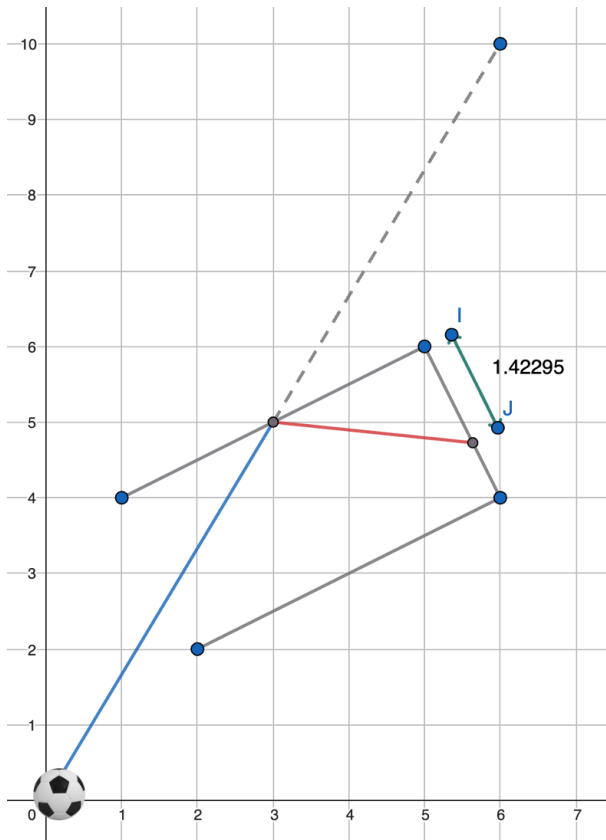
Your answer is accepted if and only if $\frac{|a-b|}{\max(1,b)} \leq 10^{-3}$.

input
2 2
2 0
4 0
4 2
0 3 3 2
output
1

input
2 2
6 4
5 6
1 4
0 0
6 10
output
1.42295234932



For the first sample: The goal is the black rectangle with 3 sides filled in. The blue + red path is the path taken by the ball. You can stand at one of the edge of the back wall, being 1 unit distance away from the point of impact.



For the second sample: The goal is the black rectangle with 3 sides filled in. The blue + red path is the path taken by the ball. Note that the aiming point can be behind the goal, but the ball will hit the wall of the goal first. You can stand at point $(5, 6)$ on the back wall, which is the farthest away from the point of impact, with a distance of 1.42295 units.

M. Miracles can be Created

1 second, 1024 megabytes

Tomori is playing a minigame at a card shop. In the game, there are $N + 1$ cards numbered from 0 to N , with each number appearing on exactly one card. The game works as follows: the player chooses one card, and then the shop rewards the player with a card whose number is equal to the bitwise XOR of the numbers on the N cards that were not chosen.

Since Tomori already owns all the cards numbered from 0 to N , she wants to obtain a card with a number greater than N — that is, a card she doesn't already have.

For example, when $N = 12$, Tomori can choose the card numbered 3 and she will win the card numbered $0 \oplus 1 \oplus 2 \oplus 4 \oplus 0 \oplus \dots \oplus 12 = 15$, a card she doesn't already have.

However, she realizes that for some values of N it is impossible to win such a card. To avoid wasting time and money, she wants to know in advance whether her goal is achievable for a given N . Help her determine that!

Input

The first line of input contains one integer N ($1 \leq N < 2^{30}$).

Output

If it is possible for Tomori to win a new card, print YES. Otherwise, print NO.

input
12
output
YES

input
3

output
NO