Problem A. Audioactive Sequence

Input file:	`standard input`
Output file:	`standard output`
Time limit:	4 seconds
Memory limit:	256 megabytes

John loved numbers. One day he was bored, so he decided to write N of them. The first number John wrote was $a_1 = A$. Each following number a_i, for $2 \le i \le N$, would be obtained from the previous one as follows. John started counting out loud how many times each digit appeared in a_{i-1}, starting from digit zero until nine. He recorded the count of every digit that did show up, by writing it next to a_{i-1}, thus forming a_i. For example, when $A = 1$ and $N = 6$, John's sequence was

$$1, 11, 112, 11221, 1122132, 1122132331,$$

because, for example, there are 2 ones and 1 two in $a_3 = 112$, so that $a_4 = 11221$.

Given positive integers A, M, N, your task is to output the remainder when the last number written by John is divided by M.

Input

The only line of input contains three positive integers A, M, N with $A, M \le 10^{18}$ and $N \le 10^5$.

Output

Output a single positive integer, the answer to the problem.

Examples

standard input	standard output
`1 12345 4`	`11221`
`110 12345 2`	`11012`
`113 12345 2`	`11321`
`11111111110 999999999999999999 3`	`11111111110110212`

Note

In the last test case, the sequence is given by

$$11111111110, 11111111110110, 11111111110110212,$$

because there is 1 zero and 10 ones in a_1, and 2 zeros and 12 ones in a_2.

Problem B. Best Teammates

Input file:	`standard input`
Output file:	`standard output`
Time limit:	4 seconds
Memory limit:	256 megabytes

You recently got interested in competing in some team programming contest, but with everything being virtual, you realized you don't know anyone in college anymore. There were N other students interested in participating, with problem solving skills of $s_1, \ldots, s_N$. Of course, you don't know that, since you didn't really talk to them. In fact, you don't even know what they look like. All you know is that your problem solving skill is S, so you decided to send an email to two random interested students and they happily agreed to form a team. The skill of a team is defined as the product of the skills of each team member.

After the contest, you talked to everyone and found out their skills. Now you're left wondering: what was the expected value of the skill of the team you formed?

Input

The first line of input contains two space-separated positive integers $N, S \leq 1000$ with $N \geq 2$ and S, the number of others interested in the contest and your skill level.

The second line of input contains N space-separated positive integers $s_1, \ldots, s_N \leq 1000$, the skill level of the other students.

Output

Print a single decimal number, the expected value of the skill of your team. Your answer will be considered correct if it differs by at most 10^{-4} from the correct answer.

Examples

standard input	standard output
2 1 2 3	6.00000
3 3 3 3 9	63.00000

Note

In the first test case, there is only one possible team of skill $1 \times 2 \times 3 = 6$.

In the second test case, there are three possible teams of skills $3 \times 3 \times 3, 3 \times 3 \times 9$ and $3 \times 3 \times 9$. The expected skill of a team picked at random is $(27 + 81 + 81)/3 = 63$.

Problem C. Constellation

Input file:	`standard input`
Output file:	`standard output`
Time limit:	4 seconds
Memory limit:	256 megabytes

It had been almost a year since the two of you met—incredible how time flies by. The winter gust, so often cold and unforgiving, tonight felt crisp if not rejuvenating. Moonlight glimmered on her dress as she stood patiently at the corner of some numbered street with some numbered avenue, but those numbers did not matter. All that mattered to you was remembering that list: the constellation of N stars.

It wasn't the same, and the night went slowly. Of course, you had an ace up your sleeve. She had to like stars; how could a budding astronomer not? And there it was, a blessing from the skies: not a single cloud obscured the view. You were ready for your spectacle. It was obvious where to start: at the Initial Nova, the first star of your list. You recalled a property of the constellation: every star was within a distance of R (the *radius*) of exactly two other stars in the constellation, with the exception of the Initial Nova and the Final Nova, which were within a distance of R of exactly one other star.

All you had to do to impress her was to point at the sky and identify the stars. But your list was too scrambled. You started to stammer, and before long, she was bored: "That one? That's just Scorpio."

Input

The first line of input contains two space-separated positive integers $N, R^2 \leq 10^5$, the number of stars and the squared radius of the constellation.

The next N lines of input contain three space-separated integers x_i, y_i, z_i each of absolute value at most 10^5, the coordinates of the stars in the order of your list. It is guaranteed that these correspond to a valid constellation of radius R, but the list may be out of order.

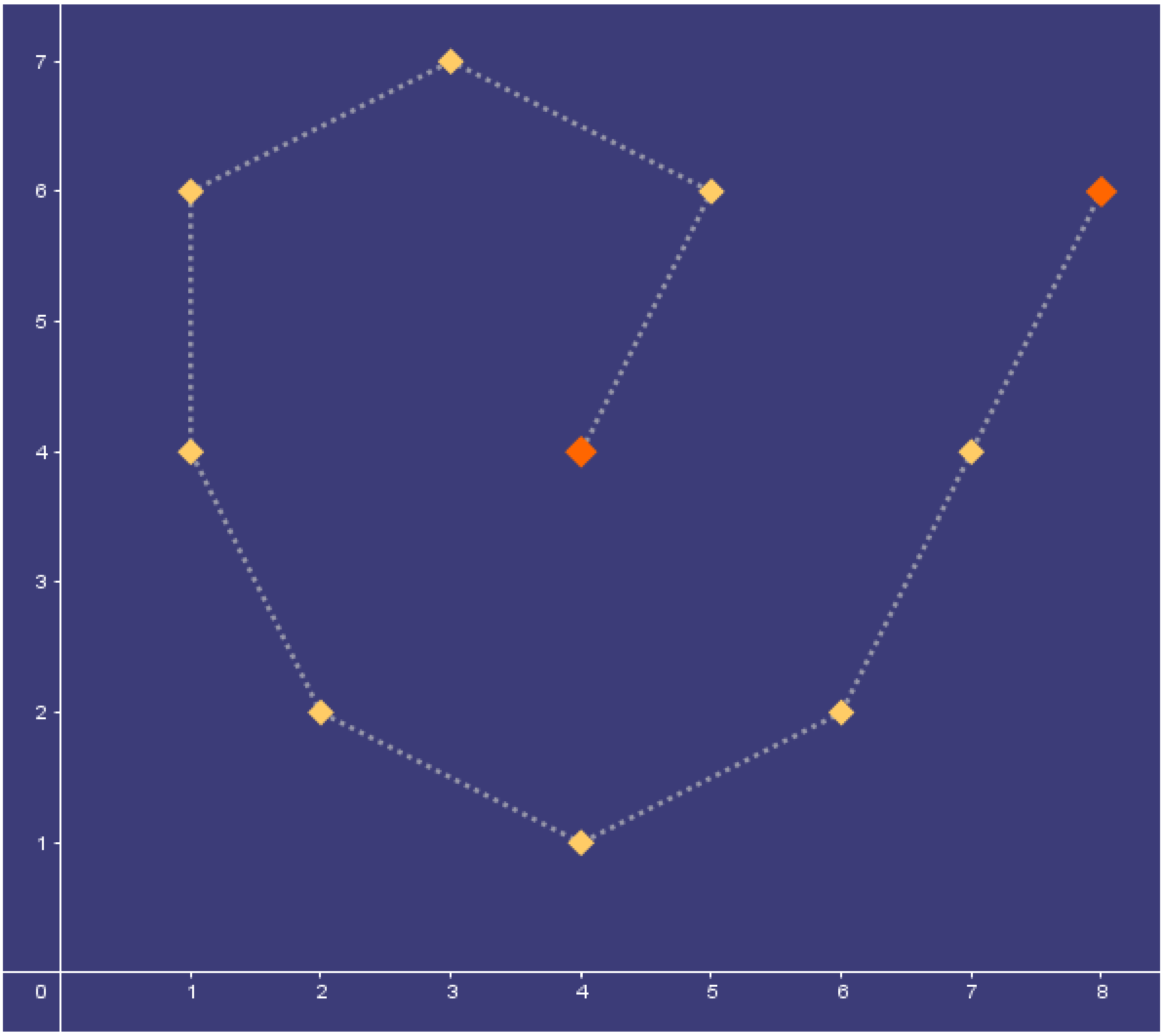
Output

Output a single line starting with 0 (the Initial Nova) followed by $N - 1$ positive integers denoting the indices, as they appear on your list, of the constellation's path: a sequence of stars, each within distance R of its neighbors.

Examples

<table>
<tr><th>standard input</th><th>standard output</th></tr>
<tr><td><pre>10 5
4 4 0
7 4 0
4 1 0
2 2 0
8 6 0
5 6 0
3 7 0
1 4 0
1 6 0
6 2 0</pre></td><td><pre>0 5 6 8 7 3 2 9 1 4</pre></td></tr>
<tr><td><pre>5 1
0 0 0
1 1 1
0 1 0
0 1 1
1 1 2</pre></td><td><pre>0 2 3 1 4</pre></td></tr>
</table>

Note



Constellation for the first test case.

Problem D. Dudivorce

Input file:	standard input
Output file:	standard output
Time limit:	4 seconds
Memory limit:	256 megabytes

When Dudina first laid eyes on Dudino, she knew they were in for a ride. The two were starting their career as flight attendants at Dudu Airlines, a reputable company servicing N airports with M bidirectional direct flights connecting airports.

Alone they were strong, together unstoppable. Dudina and Dudino climbed the corporate ladder and ten years later became the two owners of Dudu Airlines. To their many employees they were a power couple, a symbol that it was indeed possible to maintain a relationship with someone you work with.

Alas, the fairy tale was all but a façade. Their initial outburst of passion notwithstanding, love swiftly dudwindled, and through layovers, layouts and layoffs, it was progressively replaced by a cold cohabitation. Their conjugal disputes seeped into management, and both knew what to do: divorce was inevitable.

The most difficult part of the divorce was not emotional. It was clear the company had to be divided into Dudina Airlines, whose color would be red, and Dudino Airlines, whose color would be blue. Since some airports are more important than others, they decided that a fair settlement would be to minimize the unfairness of the divorce, defined as the maximal difference of the number of red and blue flights at some airport. They hired you to help with their breakup.

Input

The first line of input consists of two positive integers $N, M \leq 10^6$ with $N \geq 2$. The next M lines each contain two distinct positive integers a_i b_i denoting that there is a bidirectional flight $a_i \leftrightarrow b_i$. There will only be at most one such bidirectional flight for every pair of airports. It is guaranteed that one can fly from any airport to any other via a sequence of flights.

Output

The first line of output should consist of a single nonnegative integer denoting the minimum unfairness of the divorce. The second line of input should consist of M space-separated characters c_1 c_2 $\ldots$ c_M with $c_i \in \{R, B\}$ denoting the color of the i-th flight as it appears in the input. If there are many fair divorce settlements you may output any of them.

Examples

standard input	standard output
3 3 1 2 2 3 3 1	2 B R B
4 4 1 2 2 3 3 1 1 4	1 B R B R
4 4 1 2 2 3 3 4 4 1	0 R B R B

Problem E. Escalator

Input file:	`standard input`
Output file:	`standard output`
Time limit:	4 seconds
Memory limit:	256 megabytes

Esmeralda is a renowned architect, and recently she was hired to design a twenty-first century royal palace. Its expensive structure made of precious stones contains a long hallway with an irregular ceiling divided into N sections, so that the i-th section extends from the $i-1$-th meter until the i-th meter from the entrance, and extends a height of h_i from the floor.

She must choose carefully an interval of the hallway where she can place an escalator made of amethyst. An escalator of size k has height 1 meter in its (leftmost) starting location, then increases in height by one as it grows to the right, up to a height of k meters. Naturally, an escalator can only fit in the hallway if the height of the ceiling is at least that of the escalator.

Now Esmeralda wonders, for each interval she considers, what is the largest size of an escalator that may fit completely inside that interval.

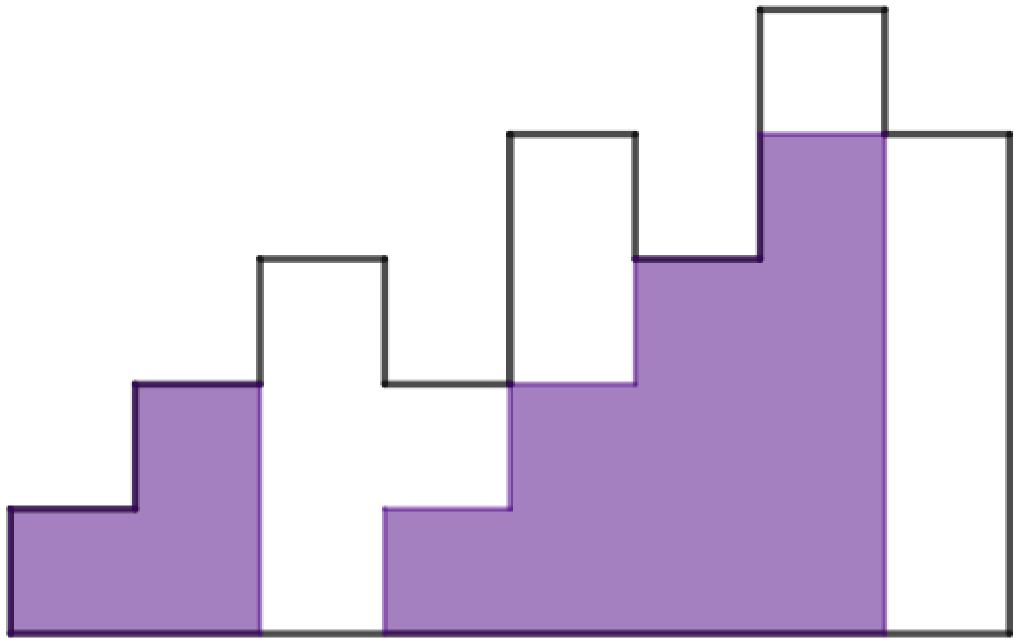


Image for the first test case. Note that the second escalator could have been built starting one unit to the right as well, but it would still have a size of 4.

Input

The first line of input contains two space-separated positive integers $N, I \leq 10^5$, denoting the number of sections in the hallway and the number of section intervals Esmeralda considers.

The second line of input contains N space-separated integers $h_1, \ldots, h_N \leq 10^9$ denoting the height of the i-th section of the hallway.

The following I lines of input contain two space-separated positive integers $\ell_i \leq r_i \leq N$, denoting that Esmeralda is considering building an escalator somewhere within the range $[\ell_i, r_i]$. Note that the answer to each interval does not affect other intervals.

Output

Output I lines, each with a single positive integer containing the largest size of an escalator that may fit in the i-th section interval.

Example

standard input	standard output
8 4 1 2 3 2 4 3 5 4 1 2 3 8 2 4 4 4	2 4 2 1

Problem F. Filling Braids

Input file:	`standard input`
Output file:	`standard output`
Time limit:	4 seconds
Memory limit:	256 megabytes

Dudina's father, once a great painter, was hit hard by the pandemic. He was having a lot of trouble getting his creative juices flowing, and often spent many hours staring at a blank canvas.

One day, he was particularly stumped, so much that he fell asleep and overslept in his atelier. Dudina, noting the dismal state of her father, tried to help. She brought a large collection of N very long braids, and started gluing them to her father's wide blank canvas.

Once father awoke from a slumber, he was in awe. Such was the marvel created by his daughter, and he could not help but color with paint her majestic arrangement. He colored purple every point in the canvas with the property that every ray (half-line) starting at that point intersects some braid. Now he wonders how much paint he will need. Each squared unit of canvas takes one milliliter of paint to fill.

Input

The first line of input contains a positive integer $N \leq 1000$, the number of braids Dudina used.

Each of the following N lines contains three space-separated integers a_i, b_i, c_i of absolute value no greater than 10^4 denoting that the i-th braid can be represented by a line with equation $a_i x + b_i y + c_i = 0$ (here it is guaranteed $a_i^2 + b_i^2 \neq 0$). You may assume that each braid is infinitely long, that no two braids will correspond to the same line, and that no three braids intersect at a single point.

Output

Output a floating point number representing the volume of paint in milliliters required by Dudina's father. Your answer will be considered correct if it differs by at most 10^{-4} of the correct answer.

Examples

standard input	standard output
3 1 0 0 0 1 0 1 1 -1	0.50000000
4 1 0 0 0 1 0 0 1 -1 1 0 -1	1.00000000
4 1 0 0 0 1 0 1 1 -2 1 -1 1	2.50000000

Note

The next page contains pictures of the paintings.

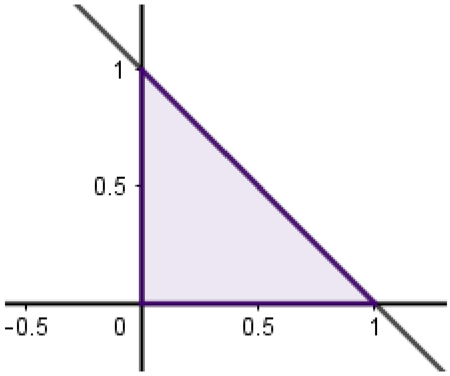


Image for first sample test.

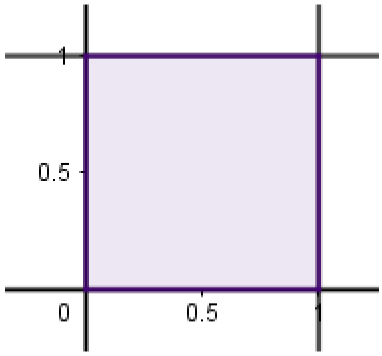


Image for second sample test.

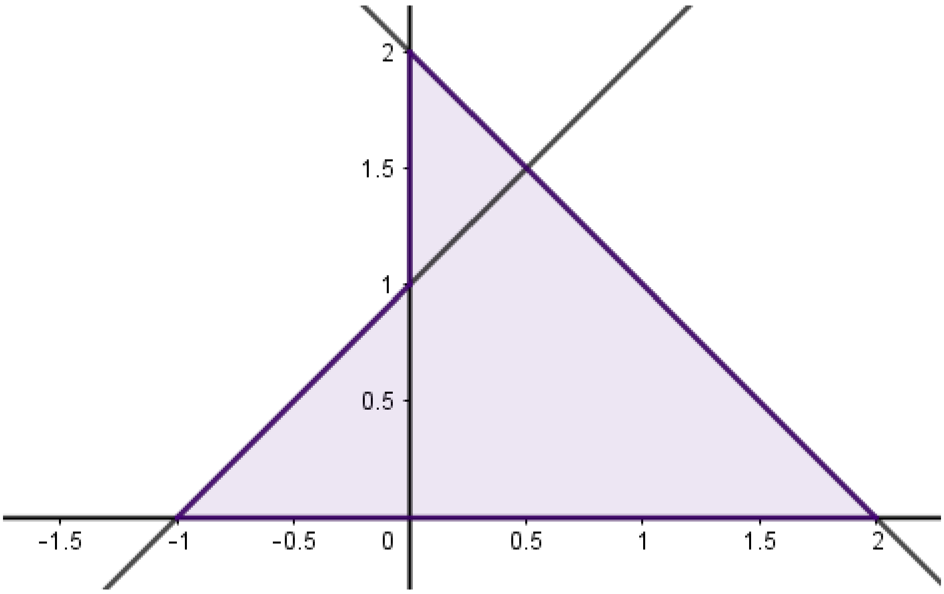


Image for third sample test.

Problem G. Game of Cards

Input file:	`standard input`
Output file:	`standard output`
Time limit:	4 seconds
Memory limit:	256 megabytes

"No the game never ends when your whole world depends
On the turn of a friendly card"

— 'The Turn Of A Friendly Card', Alan Parsons & Eric Woolfson

You challenge your arch-rival Kindred for a sparring match to settle your disputes. Alas, they did not own a gi, so you had to settle for a game of cards, No-Gi-Oh.

You and Kindred both bring your own decks of $2N - 1$ cards each, each labeled with a positive integer. You two know each other too well, and are aware of each other's cards. In every round, you both play a card face down. The cards are revealed simultaneously, and the player who placed the larger card wins the round. At the end of the game, $2N - 1$ rounds have been played, and the player who has won at least N rounds is declared the victorious No-Gi-Oh champion.

You are planning on coming up with some strategy, but first, you want to know whether either player can guarantee a victory regardless of the other's moves.

Input

The first line of input contains the positive integer $N \leq 10^5$.

The following line contains $2N - 1$ positive integers $a_1, \ldots, a_{2N-1} \leq 10^9$, the labels of your cards.

The following line contains $2N - 1$ positive integers $k_1, \ldots, k_{2N-1} \leq 10^9$, the labels of Kindred's cards.

Since the two of you are arch-rivals, your cards are disjoint, that is $a_i \neq k_j$ for all $1 \leq i, j \leq 2N - 1$.

Output

If you can guarantee victory regardless of how Kindred plays, print `EZCLAP`.

If Kindred can guarantee victory regardless of how you play, print `NERFPLS`.

If neither player can guarantee victory, print `TURNOFAFRIENDLYCARD`.

Examples

standard input	standard output
2 2 3 4 1 10 1	EZCLAP
2 1 1 1 10 200 3000	NERFPLS
3 1 2 10 25 30 5 3 20 35 45	TURNOFAFRIENDLYCARD

Problem H. Husky Pack

Input file:	`standard input`
Output file:	`standard output`
Time limit:	4 seconds
Memory limit:	256 megabytes

Husky commander Dubs assembled a pack of n dogs to protect campus against viral threat. While not really sure what to do, Dubs is certain that some, though not necessarily all, of the dogs need to assemble into platoons of size d: no more, no less. All the dogs that are not assigned to any platoon may frolic around Sylvan Grove until safety is reinstated.

First of all, Dubs needs to determine who will frolic and who will strive. Help a good boy determine the number of ways to make this initial decision so that the active dogs may form platoons later.

Input

The single line of input contains two space-separated integers $2 \le n \le 10^{18}$ and $1 \le d \le 1000$, the number of dogs in the pack and the number of dogs per platoon.

Output

Output a single integer, representing the number of ways Dubs can split the dogs into two groups of active and inactive canines. Dubs will leave the platoon division for later. Since the answer can be very large, output it modulo $10^9 + 7$.

Examples

standard input	standard output
`3 2`	`3`
`2 1`	`3`

Note

In the first sample, Dubs has to make sure the number of active dogs is even. Say the dogs are called Buddy, Daisy and Snoopy. Then Dubs can select as active dogs:

- Buddy and Daisy;
- Buddy and Snoopy;
- Daisy and Snoopy.

In the second sample, there can be platoons of any size. Say the dogs are called Buddy and Daisy. Then Dubs can select as active dogs:

- Buddy;
- Daisy;
- Buddy and Daisy.

Problem I. Inversion Counting

Input file:	`standard input`
Output file:	`standard output`
Time limit:	4 seconds
Memory limit:	256 megabytes

Dudina, the divine daredevil, learned about permutations recently. She learned that given a permutation $\pi : \{1, \ldots, n\} \to \{1, \ldots, n\}$, an *inversion* is a pair of indices $(i, j) \in \{1, \ldots, n\}^2$ so that $i < j$ and $\pi(i) > \pi(j)$. She even found out how to count the number of inversions in any permutation, and she said you can probably find out using your favorite search engine too.

Dudina is an ambitious trickster, though. Unsatisfied, she wants to count the number of inversions of big permutations. She learned that she can generate permutations by picking a prime number p and a parameter $a \in \{1, \ldots, p-1\}$.

She denotes π_a the permutation $\{1, \ldots, p-1\} \to \{1, \ldots, p-1\}$ given by $\pi_a(j) = a \cdot j \mod p$. Here, mod p denotes the remainder when divided by p. For example, when $p = 5$, the permutation π_2 is:

$$\pi_2(1) = 2, \pi_2(2) = 4, \pi_2(3) = 1, \pi_2(4) = 3.$$

Given the prime p and parameter a, help Dudina count the number of inversions in π_a.

Input

The single line of input contains two space-separated positive integers $p \leq 10^9 + 7$ and $a \leq \min(10^5, p-1)$. It is guaranteed that p is prime.

Output

Output a single positive integer, the number of inversions of π_a.

Examples

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

Note

In the first test case, there are three inversions at indices $(1, 3), (2, 3)$ and $(2, 4)$.

Problem J. Jake Joke

Input file:	`standard input`
Output file:	`standard output`
Time limit:	4 seconds
Memory limit:	256 megabytes

Jake often tells jokes, but they're usually not funny. One day you told him something and he wondered how many subsequences of what you just said are *Jake Jokes* of order k. A Jake Joke of order k is any string which does not contain k consecutive vowels, or k consecutive consonants. Jake believes that the letters a, e, i, o, u, y are vowels, and that every other letter is a consonant. Since the number of Jake Jokes can be astronomically large, output it modulo $10^9 + 7$.

Two subsequences are considered distinct if they are formed by different indices in the original string.

Input

The first line of the input contains a positive integer k with $2 \le k \le 1000$.

The second line of the input contains a string of no more than 1000 lowercase English letters representing what you told Jake.

Output

Output a single positive integer, the answer to the problem.

Examples

standard input	standard output
`2` `jake`	`11`
`3` `ooo`	`6`
`1000` `notnowjake`	`1023`

Note

In the first test case, the 11 subsequences are j, a, k, e, ja, je, ka, ke, jak, ake, jake.

In the second test case, there are 3 subsequences equal to o and 3 subsequences equal to oo.

Problem K. Keep Year

Input file:	`standard input`
Output file:	`standard output`
Time limit:	4 seconds
Memory limit:	256 megabytes

2020 was a leap year because 2020 is a multiple of 4 but not of 100. 2021 is not a leap year, but it is a *keep year*, which means 2021 is a product of two or more consecutive primes: $2021 = 43 \times 47$.

Recall that a positive integer p is *prime* if its only divisors are 1 and p. Two primes numbers p and q are *consecutive* there does not exist a prime number a with $p < a < q$.

Given year y, determine whether it is a keep year.

Input

The single line of input consists of one positive integer $y \le 10^{18}$.

Output

Output `NO` if year y is not a keep year. If year y is a keep year, output a line containing `YES` followed by a line containing an increasing sequence of consecutive prime numbers $p_1\ p_2\ \cdots\ p_\ell$ with $y = p_1 \cdots p_\ell$. It can be proved that the sequence is unique in this case.

Examples

standard input	standard output
2022	NO
2021	YES 43 47
30	YES 2 3 5

Note

2022 is not a keep year because while $2022 = 2\times3\times337$, there exists a prime number 5 so that $3 < 5 < 337$.

Problem L. Lowest Anxiety

Input file:	`standard input`
Output file:	`standard output`
Time limit:	4 seconds
Memory limit:	256 megabytes

It's been a long time since all your friends got together, and now that all of you are vaccinated it's finally time for a great party. N friends (including you) decided to sit in a big circle and catch up.

The issue is that there are always some people who got a lot of stuff done over quarantine. Without proper social distancing procedures, such tryhards may cause a lot of anxiety in the group. More specifically, the anxiety of a sitting arrangement is defined as the maximum value of the absolute difference of how much two people who sit next to each other got done.

An avid programmer, you decided to minimize the anxiety of your group.

Input

The first line of input contains a single positive integer $N \leq 10$, the number of people at the party.

The second line of input contains N integers, each at most 1000, denoting how much the i-th person got done over quarantine.

Output

Output a single positive integer, the minimal anxiety of a sitting arrangement.

Examples

standard input	standard output
2 3 1	2
4 2 4 7 10	6

Note

In the first test case, any seating arrangement has an anxiety of 2.

In the second test case, if the tryhard who got 10 things done over quarantine sits next to the people who got 4 and 7 things done, the anxiety of the group will be $|10 - 4| = 6$.