Problem A. Kendama Challenge

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	1024 megabytes

Naniwazu-kun is going to take on the traditional kendama challenge in a New Year's Eve music program. If at least K people succeed consecutively, the record will be broken. To maximize the probability of success as much as possible, he decided to challenge it with a team of N carefully selected players.

The N players will attempt the kendama in order from the 1-st to the N-th. The probability that the i-th player $(1 \le i \le N)$ succeeds is $\frac{A_i}{B_i}$, and each player's success or failure is independent.

Find the probability that there exists at least one segment where K or more consecutive players succeed, modulo 998244353.

Input

In the first line, integers N, K are given separated by a space. $(1 \le K \le N \le 2 \times 10^5)$

In the following N lines, the i-th line contains integers A_i, B_i separated by a space. $(1 \le A_i \le B_i \le 998244352)$

Output

Output the probability modulo 998244353 in a single line.

Examples

standard input	standard output
2 2 1 1 1 2	499122177
5 4 1 1 1 1 1 1 1 1 1 10000	1
5 3 3 14 1 59 2 65 3 58 9 79	62790646

Note

For the first example:

Let S denote the event that player i succeeds, and F denote the event that player i fails.

The possible patterns and their probabilities are as follows:

- SS : $1 \times \frac{1}{2} = \frac{1}{2}$
- SF : $1 \times \frac{1}{2} = \frac{1}{2}$

Only SS contains a segment where at least 2 players succeed consecutively, and its probability is $\frac{1}{2}$.

Since $499122177 \times 2 \equiv 1 \pmod{998244353}$, output 499122177.

For the second example:

Even if Naniwazu-kun, who performs last, is not very skilled, it is fine as long as the helpers are reliable.

Definition of probability modulo 998244353

It can be proven that the required probability is always a rational number. Under the constraints of this problem, when the probability is expressed as a reduced fraction $\frac{y}{x}$, it is guaranteed that x is not divisible by 998244353.

In this case, there exists a unique integer z with $0 \leq z \leq 998244352$ such that $xz \equiv y \pmod{998244353}$. Output this z.

Problem B. Cat Cut

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	1024 megabytes

You are given N strings $S_1, \ldots, S_N$, each consisting of lowercase English letters and having length M.

Initially, let $X = S_1$, and perform the following operation $N - 1$ times.

In the i-th operation, let Y be the string formed by concatenating X and S_{i+1} in this order. Then, choose any contiguous substring of Y of length M, and replace X with that substring.

Output the lexicographically smallest string that can possibly be the final value of X.

Input

The first line contains integers N, M in this order. $(2 \leq N \leq 2000, 1 \leq M \leq 2000)$

Each of the following N lines contains a string S_i of length M, consisting of lowercase English letters.

Output

Print the answer.

Examples

standard input	standard output
2 3 cat cut	atc
2 1 a b	a
3 8 jastaway tatesoto soryuusi	asoryuus

Note

In the first example, initially $X =$ `cat`.

In the first operation, $Y =$ `catcut`. If we choose the contiguous substring from the 2-nd to the 4-th character, then $X =$ `atc`, which is the lexicographically smallest possible.

Problem C. Partition AND/OR Aggregation

Input file:	`standard input`
Output file:	`standard output`
Time limit:	5 seconds
Memory limit:	1024 megabytes

You are given a sequence of positive integers $(A_1, \ldots, A_N)$ of length N. Consider partitioning this sequence A into M contiguous non-empty subsequences $B_1, B_2, \ldots, B_M$.

For a subsequence $B = (A_L, \ldots, A_R)$, define its **score** by

$$S(B) = \frac{A_L \text{ and } A_{L+1} \text{ and } \cdots \text{ and } A_R}{A_L \text{ or } A_{L+1} \text{ or } \cdots \text{ or } A_R}$$

where $x \text{ and } y$ and $x \text{ or } y$ denote the bitwise AND and bitwise OR of x and y, respectively.

Once a partition is fixed, we obtain M values $S(B_1), \ldots, S(B_M)$ as the scores of the subsequences. Sort these values in **descending** order, and define the score of the partition as the K-th value in that order. Considering all possible partitions, find the maximum and minimum possible values of the partition score.

Input

The first line contains integers N, M, K in this order. $(1 \le K \le M \le N \le 10^5)$

The second line contains N integers $A_1, \ldots, A_N$ in this order. $(1 \le A_i < 2^{30}\ (1 \le i \le N))$

Output

Print 2 lines.

Let the maximum and minimum values be $\frac{p}{q}$, $\frac{r}{s}$, respectively, where $p, r \ge 0$, $q, s \ge 1$, $\gcd(p, q) = \gcd(r, s) = 1$. Print p, q on the first line, and r, s on the second line, separated by spaces, in this order.

Examples

standard input	standard output
5 3 3 6 5 7 3 2	2 3 1 7
5 1 1 3 1 4 1 5	0 1 0 1
9 5 3 998 244 353 469 762 49 754 974 721	1 1 208 1023

Note

For the first example, if we choose $B_1 = (6)$, $B_2 = (5, 7)$, $B_3 = (3, 2)$, then $S(B_1) = 1$, $S(B_2) = \frac{5}{7}$, $S(B_3) = \frac{2}{3}$. When these are sorted in descending order, the 3-rd value is $\frac{2}{3}$.

Also, if we choose $B_1 = (6)$, $B_2 = (5, 7, 3)$, $B_3 = (2)$, then $S(B_1) = 1$, $S(B_2) = \frac{1}{7}, S(B_3) = 1$. When these are sorted in descending order, the 3-rd value is $\frac{1}{7}$.

The bitwise AND $x \text{ and } y$ and bitwise OR $x \text{ or } y$ of non-negative integers x, y are defined as follows.

- In the binary representation of $x \text{ and } y$, the digit at the 2^k $(k \ge 0)$ place is 1 if and only if the digits at the 2^k place in the binary representations of both x and y are 1; otherwise, it is 0.
- In the binary representation of $x \text{ or } y$, the digit at the 2^k $(k \ge 0)$ place is 1 if and only if at least one of the digits at the 2^k place in the binary representations of x and y is 1; otherwise, it is 0.

For example, $3 \text{ and } 5 = 1$, $3 \text{ or } 5 = 7$ (in binary, $011 \text{ and } 101 = 1$, $011 \text{ or } 101 = 111$).

Problem D. Campaign Speech

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	1024 megabytes

You are given a simple N-gon on the xy-plane all of whose edges are parallel to either the x-axis or the y-axis (that is, a polygon with no self-intersections and no holes).

There are M speech locations on the boundary of this polygon.

When movement is allowed only along the boundary of this polygon, find the minimum total travel distance required to visit all speech locations exactly once.

The starting point and ending point of the movement may be chosen arbitrarily on the boundary of the polygon.

Input

The first line contains the number of vertices N of the polygon and the number of speech locations M. $(4 \le N \le 10^5, 1 \le M \le 10^5)$

The $(i + 1)$-th line contains the coordinates (x_i, y_i) of the i-th vertex of the simple N-gon. $(1 \le i \le N, |x_i|, |y_i| \le 10^5)$ The edges of the polygon connect the i-th vertex and the $(i+1)$-th vertex. In other words, exactly one of $x_i = x_{i+1}$ or $y_i = y_{i+1}$ holds. (The $(N+1)$-th vertex denotes the 1-st vertex.) No vertex lies on an edge. (That is, there is no vertex with an angle of 180 degrees.)

The $(j + 1 + N)$-th line contains the coordinates (p_j, q_j) of the j-th speech location. $(1 \le j \le M, |p_j|, |q_j| \le 10^5)$ These points are all distinct and lie on the boundary of the given polygon. (This includes vertices.)

All input values are integers.

Output

Print the answer.

Examples

standard input	standard output
4 3 0 0 3 0 3 2 0 2 0 0 3 0 3 2	5
6 4 0 0 3 0 3 1 2 1 2 2 0 2 3 0 0 2 2 1 1 2	5
10 10 -12 -11 -12 9 3 9 3 1 -3 1 -3 -1 8 -1 8 -10 -5 -10 -5 -11 3 1 3 7 7 -1 -3 0 -4 -10 -12 8 -10 -11 -3 9 -12 -10 8 -7	74

Note

In the first example, there are points on three vertices of a rectangle, and visiting them in the order $(0,0) \to (3,0) \to (3,2)$ gives the shortest total travel distance, which is $3+2=5$.

Figure for Sample Input 1

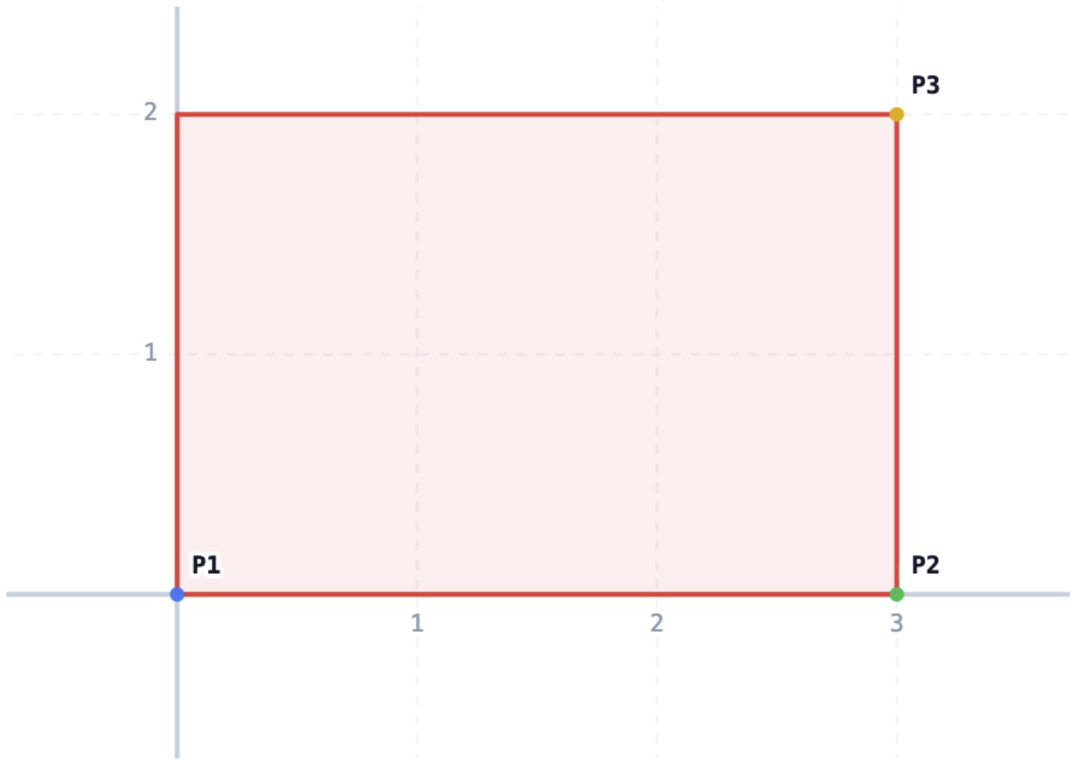


Figure for Sample Input 2

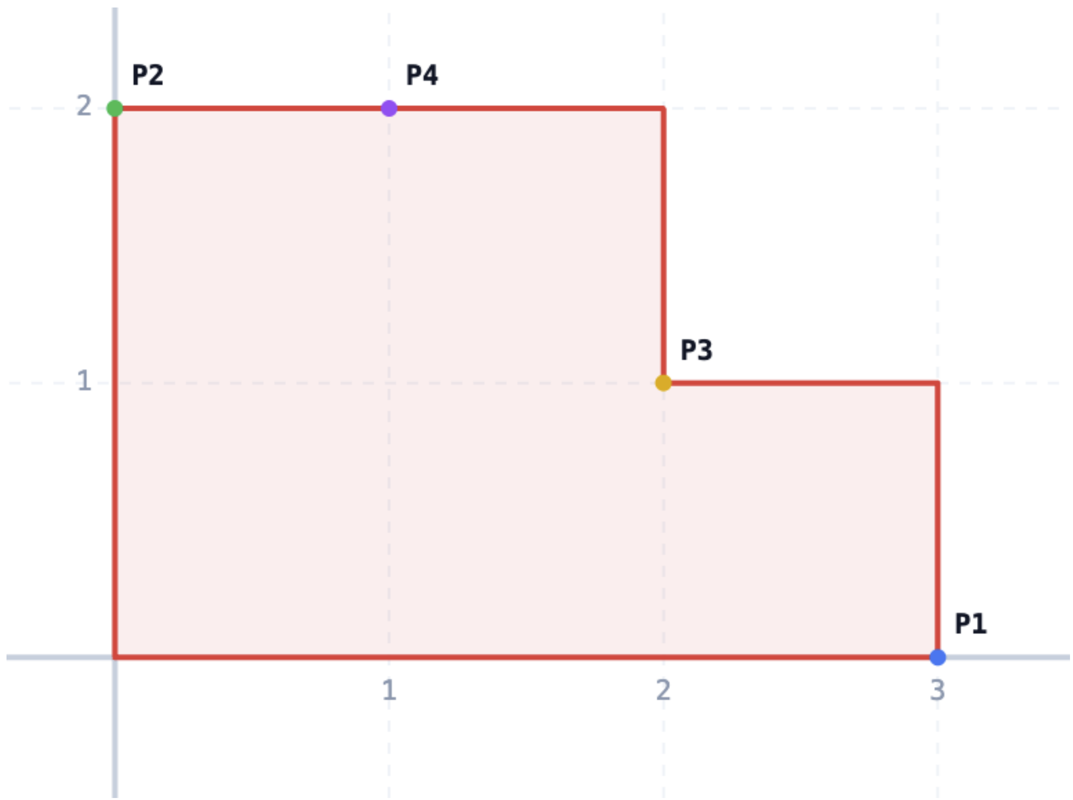
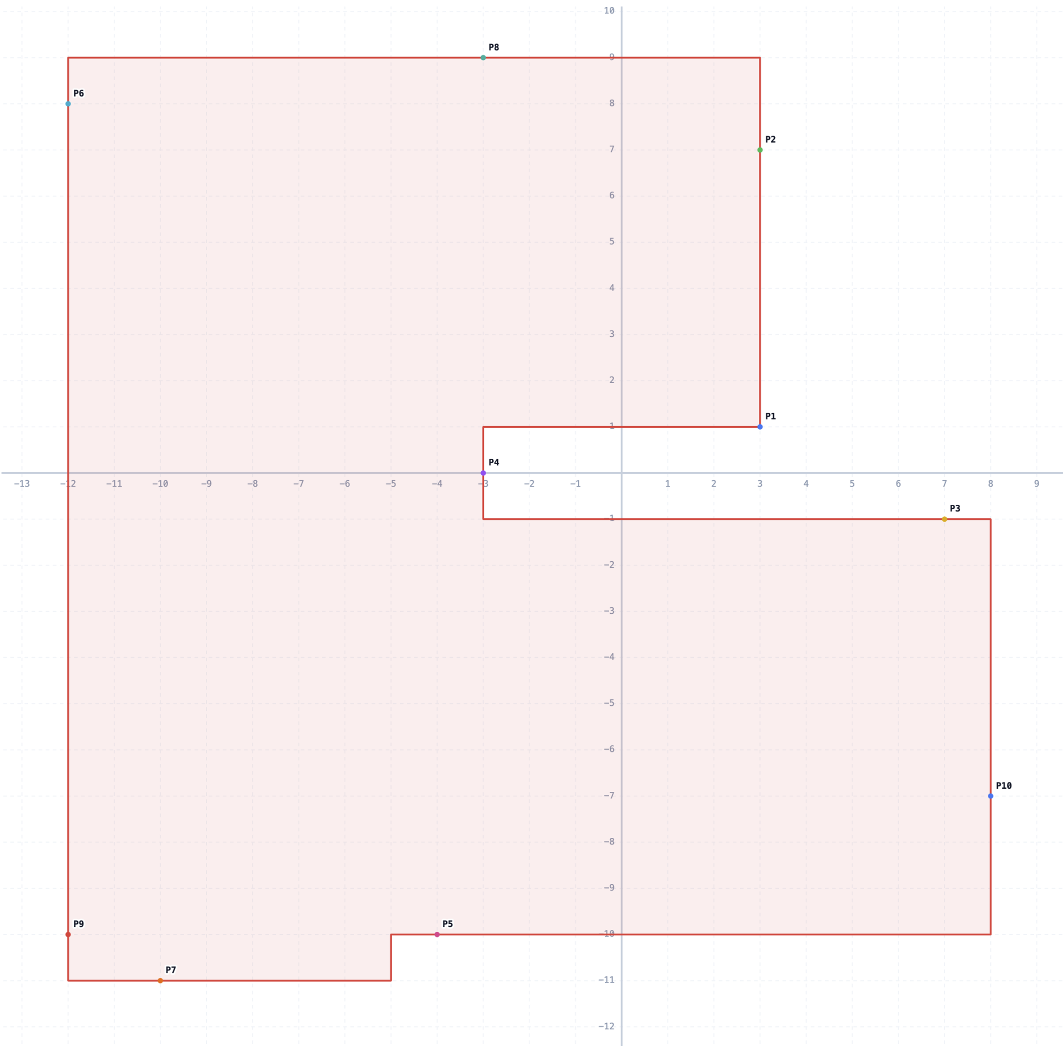


Figure for Sample Input 3


P1
P2
P3
P4
P5
P6
P7
P8
P9
P10

Problem E. Ball Dumping Golf

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	1024 megabytes

There is one box labeled with each integer from 1 to N. Also, for each integer from 1 to N, there are M balls labeled with that integer.

These NM balls are shuffled and then distributed into the N boxes, with exactly M balls placed into each box.

There are $\frac{(NM)!}{(M!)^N}$ possible ways to place the balls (if all balls are considered distinguishable), and all of these arrangements occur with equal probability.

You will perform operations on these boxes and balls. One operation consists of the following steps.

1. Choose one box and go in front of that box.
2. If there is no ball in that box, terminate the operation.
3. Choose any one ball from that box and discard it outside the box.
4. Finally, go in front of the box whose label matches the label of the ball most recently discarded, and return to step 2.

Define your **score** as the number of operations required until all NM balls have been discarded. You want to **minimize** this score.

Find the **expected value** of the score when you act optimally, modulo 998244353.

Input

The first line contains N and M in this order, separated by spaces. ($1 \le N \le 10^5$, $1 \le M \le 100$)

Output

Print the answer.

Examples

standard input	standard output
2 2	166374060
3 1	831870296
100000 100	402978825

Note

For the first example, the possible ball arrangements and the corresponding optimal ways of operating are as follows.

- Put two balls labeled 1 into box 1, and two balls labeled 2 into box 2. (Probability 1/6)
  - In the first operation, go in front of box 1. From there, take out a ball labeled 1 and go in front of box 1. Then take out another ball labeled 1 and go in front of box 1 again. At this point, box 1 is empty, so terminate the operation.

  - In the second operation, go in front of box 2. From there, take out a ball labeled 2 and go in front of box 2. Then take out another ball labeled 2 and go in front of box 2 again. At this point, box 2 is empty, so terminate the operation.
  - In this case, the minimum achievable score is 2.

- Put one ball labeled 1 and one ball labeled 2 into each of box 1 and box 2. (Probability 2/3)
  - In the first operation, go in front of box 1. From there, take out a ball labeled 1 and go in front of box 1. Then take out a ball labeled 2 and go in front of box 2. Then take out a ball labeled 2 and go in front of box 2. Then take out a ball labeled 1 and go in front of box 1. At this point, box 1 is empty, so terminate the operation.
  - In this case, the minimum achievable score is 1.

- Put two balls labeled 2 into box 1, and two balls labeled 1 into box 2. (Probability 1/6)
  - In the first operation, go in front of box 1. From there, take out a ball labeled 2 and go in front of box 2. Then take out a ball labeled 1 and go in front of box 1. Then take out a ball labeled 2 and go in front of box 2. Then take out a ball labeled 1 and go in front of box 1. At this point, box 1 is empty, so terminate the operation.
  - In this case, the minimum achievable score is 1.

In summary, the minimum score is 2 with probability 1/6, and the minimum score is 1 with probability 5/6, so the expected score overall is 7/6. Therefore, output 166374060, which represents this value modulo 998244353.

Definition of expected value modulo 998244353

It can be proven that the expected value to be found is always a rational number. Also, under the constraints of this problem, if the expected value is written as a reduced fraction $\frac{y}{x}$, it is guaranteed that x is not divisible by 998244353.

In this case, there exists a unique integer z between 0 and 998244352, inclusive, satisfying $xz \equiv y \pmod{998244353}$. Output this z.

Problem F. 1e16 Cities

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	1024 megabytes

There are 10^{16} cities, labeled $1, 2, \ldots, 10^{16}$.

For distinct cities i, j, there is a bidirectional road between city i and city j if and only if $\mathrm{lcm}(i, j) = A \cdot \gcd(i, j) + B$.

Answer Q queries.

In the i-th query, an integer x_i is given. Output the bitwise XOR of the labels of all cities that are reachable from city x_i by traversing zero or more roads.

Input

The first line contains integers A, B in this order, separated by spaces. $(1 \le A, B \le 10^8)$

The second line contains an integer Q. $(1 \le Q \le 10^5)$

Each of the following Q lines contains an integer x_i for the i-th query. $(1 \le x_i \le 10^{16})$

Output

Print Q lines.

On the i-th line $(1 \le i \le Q)$, output the answer to the i-th query.

Examples

standard input	standard output
3 28 4 20 26 7 28	28 26 54 108
81781525 3945925 10 53907475 6160906250298067 3007621769603801 134161450 23675550 4034161385146811 2151358558435 12908151350610 112647860153451 9491287293575	53908389 6160906250298067 3007621769603801 9491260218029 2151369618045 4034161385146811 2151369618045 332851610359999 112647860153451 9491260218029

Note

For the first example, the existence of a road between cities i, j is equivalent to $\mathrm{lcm}(i, j) = 3 \cdot \gcd(i, j) + 28$.

- For the first query, the cities reachable from city 20 are $8, 20$.
- For the second query, the only city reachable from city 26 is 26.

- For the third query, the cities reachable from city 7 are $7, 49$.
- For the fourth query, the cities reachable from city 28 are $28, 112$.

The bitwise XOR $x \oplus y$ of non-negative integers x, y is defined as follows.

- In the binary representation of $x \oplus y$, the digit at the 2^k ($k \geq 0$) place is 1 if and only if exactly one of the digits at the 2^k place in the binary representations of x and y is 1; otherwise, it is 0.

For example, $3 \oplus 5 = 6$ (in binary, $011 \oplus 101 = 110$).

Problem G. The Symbolic Tree

Input file: `standard input`
Output file: `standard output`
Time limit: 2 seconds
Memory limit: 1024 megabytes

Kyoto University has a magnificent symbolic tree as its symbol tree.

Impressed by this symbolic tree, K-kun made a copy of it as a rooted tree with N vertices.

The vertices of the tree are labeled $1, 2, \ldots, N$, the root of the tree is vertex 1, and the i-th edge connects vertices u_i and v_i.

Thinking that merely copying the tree would be uninteresting, K-kun writes a non-negative integer on each vertex of the tree so that the following **conditions** are satisfied.

Conditions

- The integer written on the root is K.
- For each vertex other than the root, the integer written on that vertex is at most the integer written on its parent.

Find the remainder when the number of possible ways to write integers on the vertices of the tree is divided by 998244353.

Two ways of writing integers on the vertices are considered different if and only if there exists some vertex such that the integer written on that vertex is different.

Input

The first line contains integers N, K separated by spaces. ($2 \leq N \leq 3000, 1 \leq K \leq 10^9$)

Each of the following $N-1$ lines contains integers u_i, v_i separated by spaces, representing an edge of the tree.

Output

Output the answer modulo 998244353.

Examples

standard input	standard output
5 1 1 2 1 3 3 4 3 5	10
16 16 15 14 15 11 7 10 14 2 4 6 14 16 5 3 1 5 12 11 5 7 2 9 13 10 5 14 9 6 8 1	623173536

Note

For the first example, the following 10 assignments of integers to vertices $1, 2, 3, 4, 5$ satisfy the conditions.

- $1, 0, 0, 0, 0$
- $1, 0, 1, 0, 0$
- $1, 0, 1, 0, 1$
- $1, 0, 1, 1, 0$
- $1, 0, 1, 1, 1$
- $1, 1, 0, 0, 0$
- $1, 1, 1, 0, 0$
- $1, 1, 1, 0, 1$
- $1, 1, 1, 1, 0$
- $1, 1, 1, 1, 1$

Problem H. How to Validate Such a Program

Input file: `standard input`
Output file: `standard output`
Time limit: 2 seconds
Memory limit: 1024 megabytes

This is an interactive problem.

KUPC-kun wrote a program that solves the following problem.

> You are given a tree $T = (V, E)$ with N vertices.
> For a sequence $\{a_1, \ldots, a_k\}$ of vertices of T of length k, define its score as follows.
>
> - Let $d(u, v)$ be the number of edges on the path between vertices u and v in T. Then the score is $\prod_{i=1}^{k} d(a_i, a_{(i \bmod k)+1})$
>
> A subset S of V is given. For each $1 \le k \le N$, find the following value q_k.
>
> - The sum of the scores over all vertex sequences $\{a_1, \ldots, a_k\}$ of length k whose elements belong to S, taken modulo $2^{61} - 1$

KUPC-kun secretly keeps the information of the tree T in advance, and continues using T as the tree given to the program.

You are trying to recover the information of the leaves using the above program. Letting the number of vertices of the tree be N, you may ask the following question at most N times.

- Choose a subset S of V and ask for the output of the program.

Assuming that KUPC-kun's program is correct, identify all leaves contained in the tree T from the information obtained through the questions.

The judge is not adaptive, and the tree T is fixed before the interaction begins.

Interaction Protocol

First, N is given from standard input. ($2 \le N \le 50$)

For each question, output to standard output in the following format.

? $s_1 s_2 \cdots s_N$

Here, $s_1 s_2 \cdots s_N$ is a string of length N representing the subset S, where $s_i = 1$ if $i \in S$, and $s_i = 0$ if $i \notin S$.

In response, the following is given from standard input.

$q_1\ q_2\ \cdots\ q_N$

Once you have identified all leaves, output your answer in the following format.

! $t_1 t_2 \cdots t_N$

Here, $t_1 t_2 \cdots t_N$ is a string of length N, where $t_i = 1$ if i is a leaf, and $t_i = 0$ if it is not a leaf.

After this output, terminate your program immediately.

Whenever you output something, append a newline at the end and flush standard output.

Example

standard input	standard output
5	
	? 00101
0 8 0 32 0	
	? 11001
0 44 108 968 3960	
	? 10000
0 0 0 0 0	
	? 11111
0 76 348 3336 22200	
	! 11001

Note

For the first example, the edge set of the tree secretly held by the judge is $(1,3),(2,3),(3,4),(4,5)$.

In the first question, $S = \{3, 5\}$.

Note that $d(3,3) = 0, d(3,5) = d(5,3) = 2, d(5,5) = 0$.

For example, there are the following 4 vertex sequences a of length 2 whose elements belong to S.

- If $a = (3,3)$, then the score of this sequence is $d(3,3) \times d(3,3) = 0 \times 0 = 0$
- If $a = (3,5)$, then the score of this sequence is $d(3,5) \times d(5,3) = 2 \times 2 = 4$
- If $a = (5,3)$, then the score of this sequence is $d(5,3) \times d(3,5) = 2 \times 2 = 4$
- If $a = (5,5)$, then the score of this sequence is $d(5,5) \times d(5,5) = 0 \times 0 = 0$

The judge responds with 8 for q_2, which is the remainder when $0 + 4 + 4 + 0$ is divided by $2^{61} - 1$.

By computing similarly for the other sequence lengths, we obtain the judge's response `0 8 0 32 0`.

The leaves of this tree are vertices $1, 2, 5$, and outputting `! 11001` correctly completes the identification of the leaves.

Problem I. Xor Magic Square

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	1024 megabytes

A good matrix of size N is an $N \times N$ matrix of positive integers such that the total XOR of each row, each column, and both diagonals is 0.

More precisely, an $N \times N$ matrix A is called a good matrix of size N if it satisfies all of the following conditions. Here, $x \oplus y$ denotes the bitwise XOR of x and y, and $\bigoplus_{i=1}^{N} a_i = a_1 \oplus \ldots \oplus a_N$.

- $A_{i,j}$ $(1 \le i, j \le N)$ is a positive integer
- For each $i = 1, 2, \ldots, N$, $\bigoplus_{j=1}^{N} A_{i,j} = 0$
- For each $j = 1, 2, \ldots, N$, $\bigoplus_{i=1}^{N} A_{i,j} = 0$
- $\bigoplus_{i=1}^{N} A_{i,i} = 0$
- $\bigoplus_{i=1}^{N} A_{i,N-i+1} = 0$

A positive integer N is given.

Among all good matrices of size N, output one whose total sum of all elements, $\sum_{1 \le i,j \le N} A_{i,j}$, is minimum.

If no good matrix of size N exists, report that.

Input

The input consists of a single integer N. $(1 \le N \le 2 \times 10^3)$

Output

If no good matrix of size N exists, print -1 on a single line.

If it exists, print the minimum possible total sum of all elements on the first line.

Then print the matrix A in the following N lines, with elements separated by spaces. That is, the $(i+1)$-th line should contain the elements of the i-th row of matrix A, separated by spaces.

If there are multiple solutions, any of them may be printed.

Examples

standard input	standard output

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

`1`	`-1`

Note

For the first example, the total XOR of each row, each column, and both diagonals is 0, so it satisfies the conditions for a good matrix. Also, among all good matrices of size 2, the total sum of all elements cannot be made smaller than 4, so the minimum value is 4.

For the second example, there is no good matrix of size 1, so print -1.

The bitwise XOR $x \oplus y$ of non-negative integers x, y is defined as follows.

- In the binary representation of $x \oplus y$, the digit at the 2^k ($k \geq 0$) place is 1 if and only if exactly one of the digits at the 2^k place in the binary representations of x and y is 1; otherwise, it is 0.

For example, $3 \oplus 5 = 6$ (in binary, $011 \oplus 101 = 110$).

Problem J. Sum of max of iai

Input file:	`standard input`
Output file:	`standard output`
Time limit:	4 seconds
Memory limit:	1024 megabytes

You are given a positive integer N and a prime number P.

For a permutation $(a_1, a_2, \cdots, a_N)$ of $1, 2, \cdots, N$, define its **score** $f(a)$ as follows.

$$f(a) = \max\{ia_i \mid i = 1, 2, \cdots, N\}$$

Find the remainder when the sum of the scores over all permutations is divided by P.

Input

The first line contains N, P in this order, separated by spaces. $(1 \le N \le 10^4, 10^8 \le P < 10^9, P$ is prime$)$

Output

Print the remainder when the sum of the scores over all permutations is divided by P.

Examples

standard input	standard output
`10 100000007`	`77379290`
`1000 998244353`	`168695631`

Note

For the first example, for instance, $f(3, 9, 4, 10, 8, 2, 7, 5, 6, 1) = 54$.

The sum of the scores over all 10! permutations is 277379304, and the remainder when this is divided by the prime $P = 10^8 + 7$ is 77379290, so print 77379290. **Note that for this input, the prime number P is the 9-digit integer $10^8 + 7$.**

Problem K. Square Resistance Value

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

Using only resistors with resistance $1\ [\Omega]$*, construct a resistor with resistance* $\sqrt{D}\ [\Omega]$.

You are given a positive integer D. Construct one **connected undirected graph** satisfying all of the following conditions. Under the constraints of this problem, it can be proven that such a graph always exists.

- The number of vertices N is between 2 and 300, inclusive, and each vertex has a distinct label from 1 to N
- The number of edges M is at most 300, and self-loops and multiple edges are allowed
- The "effective resistance from vertex 1 to vertex N", defined as below, is within an **absolute error of** $\pm 10^{-6}$ **from** $\sqrt{D}$

Let G be a connected undirected graph with n vertices and m edges ($n \geq 2$), and suppose that the j-th edge connects vertices a_j, b_j. Consider assigning a real number V_i ($i = 1, 2, \cdots, n$) to each vertex of graph G, and a real number I_j ($j = 1, 2, \cdots, m$) to each edge, so that all of the following equations are satisfied.

- $I_j = V_{a_j} - V_{b_j} \ \ (j = 1, 2, \cdots, m)$
- $\sum_{b_j = i} I_j - \sum_{a_j = i} I_j = 0 \ \ (i = 2, 3, \cdots, n-1)$
- $\sum_{b_j = n} I_j - \sum_{a_j = n} I_j = 1$

It can be proven that such an assignment always exists, and furthermore that the value of $V_1 - V_n$ is uniquely determined. We define this value as the "effective resistance from vertex 1 to vertex n".

Input

The input consists of a single positive integer D. ($1 \leq D \leq 5000$)

Output

On the first line, print the number of vertices N and the number of edges M of the constructed graph, in this order, separated by spaces.

On each of the following M lines, the i-th line ($i = 1, 2, \cdots, M$) should contain the endpoints of the i-th chosen edge, separated by a space.

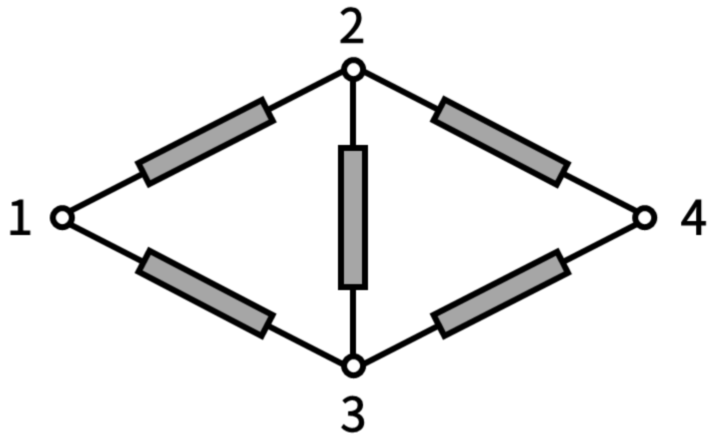
If there are multiple graphs satisfying the conditions, any one of them may be printed.

Example

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

Note

The following is an illustration of the output for the first example.



That the effective resistance from vertex 1 to vertex n is 1 [Ω] can be explained as follows.

- Since all resistors have resistance 1 [Ω], and by symmetry the potentials at vertices 2 and 3 are equal, the resistor between them can be treated as if it does not exist.
- As a result, the circuit reduces to two branches in parallel, each branch consisting of two 1 [Ω] resistors in series.
- The effective resistance of two 1 [Ω] resistors in series is 2 [Ω], and the effective resistance of two 2 [Ω] resistors in parallel is 1 [Ω].

The following output is also considered correct.

```
2 1
1 2
```

Problem L. Make Many KUPC

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	1024 megabytes

There is a string S of length N consisting of uppercase English letters. You may perform the following operation on S any number of times.

- Choose a quadruple of integers (i, j, k, l) such that $1 \le i < j < k < l \le |S|$, $S_i = \texttt{K}$, $S_j = \texttt{U}$, $S_k = \texttt{P}$, and $S_l = \texttt{C}$. Replace all of S_i, S_j, S_k, S_l with `X`, and earn $(i \times j \times k \times l)$ yen.

Find the maximum amount of money that can be earned in total, modulo 998244353.

Input

The first line contains an integer N. ($1 \le N \le 5 \times 10^5$)

The second line contains a string S of length N consisting of uppercase English letters.

Output

Print the answer.

Examples

standard input	standard output
10 KKUPCUCAPC	1164
4 TUNA	0
30 KUCCKCKKPUKUPCUCPUCKPCKKUUPCPK	619704

Note

Note that you are asked for the remainder of the maximum value, not the maximum possible remainder.

For the first example, you can earn 1164 yen by performing the following operations.

- Choose $(i, j, k, l) = (1, 3, 4, 7)$. You earn $1 \times 3 \times 4 \times 7 = 84$ yen. Then $S =$ `XKXXXCUXAPC`.
- Choose $(i, j, k, l) = (2, 6, 9, 10)$. You earn $2 \times 6 \times 9 \times 10 = 1080$ yen. Then $S =$ `XXXXCXXXAXX`.

It can be proven that it is impossible to earn more than 1164 yen, so print 1164.

For the second example, no operation can be performed even once, so print 0.

Problem M. Linked VERSE

Input file: `standard input`
Output file: `standard output`
Time limit: 7 seconds
Memory limit: 1024 megabytes

Helloooo! Can you hear me?

You are given a sequence $A = (A_1, A_2, \ldots, A_N)$ of length N consisting of integers greater than or equal to -1. Using this sequence and a parameter c, perform the following operations.

- Initially, set the variable $x := 0$.
- For $i = 1, 2, \ldots, N$, repeat the following operation.
  - If $A_i = -1$, replace x with $x := \max(0, x - c)$.
  - Otherwise, replace x with $x := x + A_i$.

Answer Q questions of the following form.

- When the parameter $c = C_i$, find the maximum value attained by x over the entire sequence of operations.

Input

The first line contains N, Q in this order. $(1 \le N, Q \le 3 \times 10^5)$

The second line contains N integers A_i. $(-1 \le A_i \le 10^6)$

Each of the following Q lines contains the value of C_i for the i-th question. $(0 \le C_i \le 10^6)$

Output

Print Q lines.

On the i-th of these lines, print the answer to the i-th question.

Example

standard input	standard output
20 11 50 100 50 100 0 200 -1 50 100 -1 200 -1 200 0 200 -1 200 200 -1 200 30 60 90 180 270 360 540 200 400 600 0	1700 1550 1400 950 570 500 500 850 500 500 1850

Note

We explain the first question of the first example input.

- In this question, the parameter is $c = 30$.
- Initially, the variable is $x = 0$.
- Since $A_1 = 50$, x is updated to $x = 0 + 50 = 50$.
- Since $A_2 = 100$, x is updated to $x = 50 + 100 = 150$.
- $\ldots$
- Since $A_6 = 200$, x is updated to $x = 300 + 200 = 500$.
- Since $A_7 = -1$, x is updated to $x = \max(0, 500 - 30) = 470$.
- $\ldots$
- Since $A_{19} = -1$, x is updated to $x = \max(0, 1530 - 30) = 1500$.
- Since $A_{20} = 200$, x is updated to $x = 1500 + 200 = 1700$.

Including the omitted parts, the maximum value attained by x over the entire sequence of operations is 1700.

Problem N. Cellular Component Constellation

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	1024 megabytes

You are given two integers N and M.

Output, in the format specified in the Output section, an $N \times N$ grid whose cells are colored either white or black and which satisfies the following conditions. If no such grid exists, print `-1`.

- The sizes of the connected components of white cells appearing in the grid consist of exactly M distinct values
- The sizes of the connected components of black cells appearing in the grid consist of exactly M distinct values

If there are multiple solutions, you may output any of them.

Input

The first line contains integers N, M in this order, separated by spaces. ($2 \le N \le 2000, 1 \le M \le 2000$)

Output

If a grid satisfying the conditions exists, print N lines. On the i-th of these lines ($1 \le i \le N$), print a string s_i of length N as follows.

- If the cell in row i, column j ($1 \le j \le N$) of the constructed grid is colored white, then the j-th character of s_i must be `.`(dot).
- If the cell in row i, column j ($1 \le j \le N$) of the constructed grid is colored black, then the j-th character of s_i must be `#`.

If no grid satisfying the conditions exists, print `-1` on the first line.

Examples

standard input	standard output

`4 2`	`###.`
	`..##`
	`##.#`
	`.##.`

`2 3`	`-1`

`12 7`	`.#..#.#.##.#`
	`.#.#..#.##.#`
	`.##...#.##.#`
	`.#.#..#.##.#`
	`.#..#.##..##`
	`......######`
	`######......`
	`#...##..###.`
	`#.##.#.#....`
	`#...##.#....`
	`#.####.#....`
	`#.####..###.`

Note

Two white cells $c_1,\ c_2$ are said to be connected if one can move from c_1 to c_2 by repeatedly moving to a vertically or horizontally adjacent cell and passing only through white cells.

A set S of white cells is called a connected component if S satisfies the following conditions.

- Any two cells in S are connected.
- No white cell not contained in S is connected to any cell contained in S.

Connected components of black cells are defined similarly.

For each connected component, its size is defined as the number of cells it contains.

Below is an appendix.

Explanation of Sample Output 1

The sizes of the connected components of white cells are the two distinct values 1 and 2. The sizes of the connected components of black cells are also the two distinct values 4 and 6.

Figure for Sample Output 1

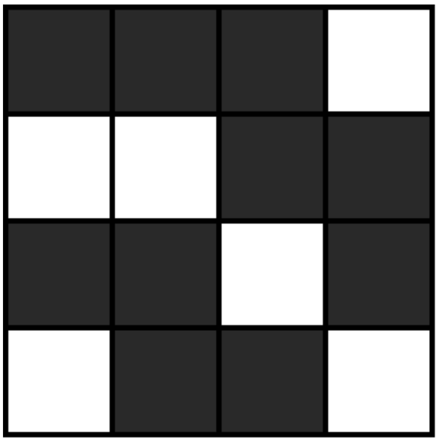
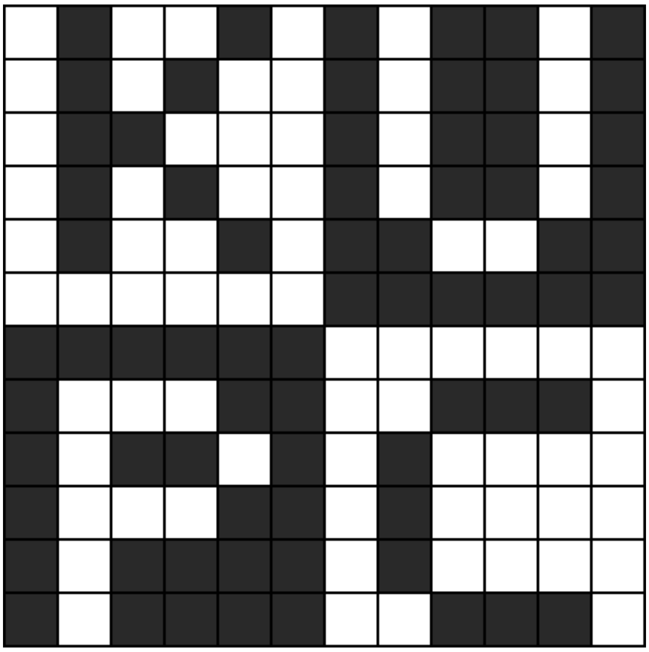


Figure for Sample Output 3



Problem O. Xor Triangle

Input file: `standard input`
Output file: `standard output`
Time limit: 2 seconds
Memory limit: 1024 megabytes

You are given a positive integer N.

Find the number of pairs of integers (a, b) such that $1 \le a, b < 2^N$ and the following condition is satisfied, modulo the prime 998244353.

- There exists a non-degenerate triangle whose side lengths are $a, b, a \oplus b$.

Here, for integers x, y, $x \oplus y$ denotes the bitwise XOR of x and y.

Input

The first line contains an integer N. $(1 \le N \le 10^{18})$

Output

Print the number of pairs of integers (a, b) satisfying the condition, modulo the prime 998244353.

Examples

standard input	standard output
2	0
5	390
10000	851087540

Note

For the second example, for instance, $(a, b) = (13, 24)$ satisfies the condition. Since $a \oplus b = 13 \oplus 24 = 21$, there exists a non-degenerate triangle whose side lengths are $13, 24, 21$.

There are exactly 390 pairs of integers (a, b) satisfying the condition.

The bitwise XOR $x \oplus y$ of non-negative integers x, y is defined as follows.

- In the binary representation of $x \oplus y$, the digit at the 2^k $(k \ge 0)$ place is 1 if and only if exactly one of the digits at the 2^k place in the binary representations of x and y is 1; otherwise, it is 0.

For example, $3 \oplus 5 = 6$ (in binary, $011 \oplus 101 = 110$).