Problem A. Natives

Time limit: 1 second
Memory limit: 512 megabytes

Captain Cook and his team were captured by island natives. Not to be eaten, the adventurers must give some treasures to natives. It turned out that the captain has n treasures.

Chieftain of the natives agrees to let the captain and his team go, if they give him at least half of his treasures. All treasures look pretty to him, so the chieftain agrees to get any treasures, he only wants to get at least half of them.

But actually each treasure has its own value known to Cook. The value of the i-th treasure is a_i. Help the captain to decide which treasures he should give to the chieftain so that the total value of the treasures he keeps to himself is maximum possible.

Input

The first line of input contains integer n ($2 \le n \le 1000$).

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 1000$).

Output

Output one integer — the maximum possible total value of treasures that the captain can keep for himself after he gives at least half of his treasures to the natives.

Example

standard input	standard output
6 2 4 1 3 3 5	12

Note

In the given example Cook can give treasures with values 1, 2 and 3 to the chieftain keeping treasures with values 3, 4 and 5 to himself. Their total value is $3 + 4 + 5 = 12$.

Problem B. Balanced Illumination

Time limit: 1 second
Memory limit: 512 megabytes

Saint Bitsburg government is preparing a technical requirement for New Year city decoration.

The governor thinks that there should be a garland of n lights on the main square. The lights will turn on and off and entertain the residents of Saint Bitsburg.

The chief designer decided that the garland would change its appearance each second. Every light in the garland can be in two states: on and off. Each second exactly one light will change its state from on to off, or from off to on. Also the chief designer wants all combination of lights in the garland to repeat with a period 2^n seconds. During the period of 2^n seconds all 2^n possible lights combinations in the garland have to be presented.

The city's chief engineer, however, noted that frequently turning lights on and off would cause their malfunction. To minimize the chance of the lights malfunction it is required for every light to be turned on and off approximately the same number of times.

So, the final technical requirement for you — Chief Programmer of the Government Department of Information Technology — is here.

- You need to make a plan of 2^n combinations of lights $a_0, a_1, \ldots, a_{2^n-1}$, where a_k is a line of n zeros and ones, $a_k[i] = 1$ means, that the light i in the combination a_k is on, $a_k[i] = 0$ means, that the light i in the combination a_k is off.
- All combinations in the plan have to be distinct.
- This plan will be launched in a cycle, each second the next combination is presented on the garland, in the t-th second the combination $a_{t \bmod 2^n}$ is presented.
- Adjacent combinations have to differ in exactly one light's state. Combination a_{2^n-1} and a_0 also have to differ in exactly one light's state.
- Let us denote by c_i the number of state changes of the light i during a complete cycle, including the final change from a_{2^n-1} to a_0. Then for any $i \neq j$ values c_i and c_j have to differ by no more than 2.

Get to work!

Input

Input contains one integer n ($1 \leq n \leq 17$).

Output

Output 2^n lines of n characters — sequence of combinations in the plan. It is guaranteed that a plan satisfying all requirements exists.

Examples

standard input	standard output
3	000 010 011 111 110 100 101 001
4	0000 0010 1010 1011 0011 0111 0110 0100 0101 0001 1001 1101 1111 1110 1100 1000

Note

In the first sample test $c_1 = c_2 = 2$, $c_3 = 4$.

In the second sample test $c_1 = c_2 = c_3 = c_4 = 4$.

Problem C. How Many Strings Are Less

Time limit: 2 seconds
Memory limit: 512 megabytes

You are given a set D of n strings and a string s. You need to find the number of strings in the set D that are lexicographically less than s.

The given string s is modified q times. Each modification is defined by a pair of an integer k_i and a character c_i. Modification (k_i, c_i) means that all characters of the string s, starting from k_i and up to the end of the string, are replaced by the character c_i.

For example, let the initial string s be "`anatoly`", then the queries $(5, \texttt{o})$, $(3, \texttt{b})$, $(7, \texttt{x})$ change the string as follows:

"`anatoly`" → "`anatooo`" → "`anbbbbb`" → "`anbbbbx`"

After each modification of the string s, you need to output the number of strings of the set D that are lexicographically less than s.

Note

String a is lexicographically less than string b if $a \neq b$ and one of two conditions is satisfied:

- a is a prefix of the string b;
- for some i, the first i characters of the string a are equal to the corresponding characters of the string b, and $a_{i+1} < b_{i+1}$.

Input

The first line contains two integers n and q — the number of strings of the set D and the number of modifications ($1 \leq n, q \leq 10^6$).

The second line contains a string s consisting of no more than 10^6 lowercase Latin letters.

The following n lines contain the strings of the set D. Each string consists of lowercase Latin letters. The total length of the strings in D does not exceed 10^6.

The following q lines contain descriptions of modifications. The description consists of the integer k_i and the lowercase letter of the English alphabet c_i, separated by a space ($1 \leq k_i \leq |s|$).

Output

The first line of output must contain the number of strings of the set D that are lexicographically less than the initial string s.

Then output q lines. In the i-th line, print the answer after the i-th modification.

Examples

standard input	standard output
4 3 anatoly boris anatooo anbbbbu anba 5 o 3 b 7 x	0 0 2 3
5 5 abcde buz ababa build a aba 1 b 3 z 2 u 4 z 1 a	3 3 3 4 4 1

Note

In the first sample test, the string changes as follows:

"anatoly" → "anatooo" → "anbbbbb" → "anbbbbx".

- Initial string "anatoly" is lexicographically less than all the strings of the set, so the answer to the problem is 0.
- After the first modification, the string becomes "anatooo" and there is an equal string in the set, but the answer to the problem is still 0, since it is not less than the current one.
- Then the string becomes "anbbbbb", which is lexicographically greater than "anatooo" and "anba", but less than "anbbbbu" and "boris", so the answer is 2.
- After the last modification, the line will become "anbbbbx", which is lexicographically greater than "anatooo", "anba" and "anbbbbu", the answer is 3.

Problem D. Exam registration

Time limit: 2 seconds
Memory limit: 512 megabytes

Usually everyone associates the New Year with a Christmas tree and a festive table, but students associate the New Year with exams session. It was December 31, and second-year students signed up for the exam.

There are n days when they can take the exam. Each student signed up to take the exam on one of these days. So, on the i-th day a_i students want to take the exam, but the maximum number of students who can take the exam on this day is b_i.

Teachers need all students to have the opportunity to take the exam, so some students may have to be moved to another day. Teachers can choose any number of students and assign each of them to take the exam on any day.

If a student wanted to take the exam on the i-th day, and the teachers eventually moved it to the j-th day, then *dissatisfaction* of this student will be equal to $|i - j|$.

Help the teachers distribute the students so that for all i on the i-th day, no more than b_i students took the exam, and the maximum dissatisfaction among the students was minimal.

Input

The first line of the input contains a single integer n — the number of days when the students can take the exam ($1 \le n \le 10^6$).

The second line of the input contains n integers a_i — the number of students who want to take the exam on the i-th day ($1 \le a_i \le 10^9$).

The third line of the input contains n integers b_i — the maximum number of students who can take the exam on the i-th day ($0 \le b_i \le 10^9$).

Output

Print a single integer — for which minimum k it is possible make the dissatisfaction of any student not to exceed k. If there is no solution, you should print -1.

Examples

standard input	standard output
4 6 14 70 1 70 3 16 5	2
1 2 2	0
1 3 2	-1

Note

Consider the first sample test. Note that 70 students want to take exam on the third day, but only 24 students can take exam on days 2, 3, and 4. So the answer is at least 2.

One of the ways to move the students in the first sample test so that no student is moved more than 2 days from his desired day is the following.

- $6, 14, 70, 1$ — initial schedule of students;

- $6, 70, 14, 1$ — move all students from the third day to the second day, and all students from the second day to the third day;
- $70, 6, 14, 1$ — move all students from the second day to the first day, and all students from the first day to the second day;
- $70, 6, 11, 4$ — move three students from the third day to the fourth day;
- $70, 3, 14, 4$ — move three students from the second day to the third day;

Note that each student was moved no more than two days from their initial position in the schedule.

Problem E. Fair Robbery

Time limit: 1 second
Memory limit: 512 megabytes

Robin Hood is well-known for stealing from the rich and giving the stolen to the poor. But this time he'll have to resort to one-sided robbery since there are only rich people in the city he's currently staying in. There are n rich people in this city in total, their houses are located in a row along the main street. The person living in the i-th house has exactly a_i money.

Robin Hood has several gang members, so he is going to prepare a plan for the robbery in advance. The plan is described by an integer k and a real number t which mean that houses with numbers $k, k+1, \ldots, n$ will be robbed, and from each of them exactly the fraction of money equal to t will be stolen. In other words, after the plan is executed, the people will have

$$a^{\text{new}} = [a_1, a_2, \ldots, a_{k-1}, (1-t)a_k, (1-t)a_{k+1}, \ldots, (1-t)a_n],$$

money left respectively. And the total amount of money stolen will be equal to

$$b = t \cdot (a_k + a_{k+1} + \ldots + a_n).$$

Let's denote the *unfairness* after the robbery as the value $\max(a^{\text{new}}) - \min(a^{\text{new}})$: the difference between the maximal and the minimal amounts of money people will have after the robbery.

Robin Hood's gang hasn't arrived to the city yet, so he doesn't know how many houses they'll be able to rob successfully. Help him to find out for each k from 1 to n, inclusive, which t between 0 and 1, inclusive, corresponds to the minimum possible unfairness after the robbery with the plan (k, t). If for a fixed k the minimum unfairness can be achieved with several different values of t, you should choose the plan that maximizes the total amount of money stolen.

Input

The first line of input contains a single integer n — the number of people living in the city ($1 \le n \le 2 \cdot 10^5$).

The second line of input contains n space-separated integers a_i — the initial amount of money each townsperson has ($1 \le a_i \le 10^9$).

Output

Print n real numbers t_i ($0 \le t_i \le 1$). For each k between 1 and n the pair (k, t_k) should denote the plan with minimal possible unfairness after the robbery among all plans with such k, and among those — the plan with maximal possible total amount of stolen money.

Your answer is accepted if the absolute or relative error of each printed number compared to the correct answer is not greater than 10^{-9}.

Examples

standard input	standard output
3 1 4 2	1.00 0.75 0.50
3 3 2 1	1.00 0.00 0.00

Problem F. Counting Antibodies

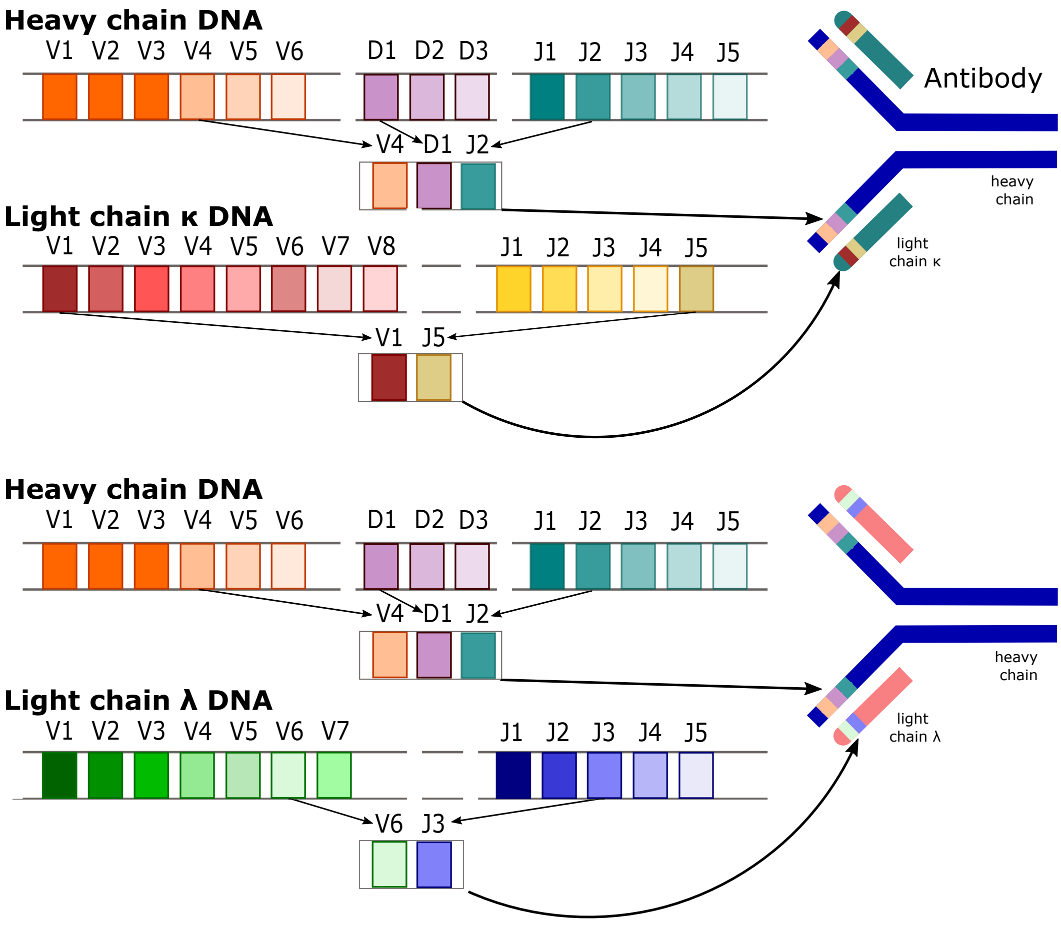
Time limit:	1 second
Memory limit:	512 megabytes

Immunoglobulins also known as antibodies are protein molecules. Antibodies play one of the key roles in the immune reaction — they detect harmful foreign agents — bacteria or viruses — and help to eliminate them. Every foreign molecule binds with unique type of antibody. There are plenty of potential harmful agents, so, there should be a tremendous number of immunoglobulin types to neutralize all threats. Huge required amount of immunoglobulin types make it impossible to encode each type in DNA. Luckily, there is a solution.

Immunoglobulins are produced by immune cells: B-lymphocytes based on DNA information — genes. Immunoglobulin genes are combined from DNA fragments like constructor. Each fragment exists in several variants and is responsible for variable region of immunoglobulin molecule. This process is called somatic recombination.

Immunoglobulin molecule consists of two identical heavy chains and two identical light chains. There are two types of light chains with similar structure — κ and λ. Both of two light chain types have two variable regions — V and J. To form a variable region one gene fragment is selected from multiple variants: V_κ and J_κ variants for V and J regions respectively in κ-light chain, or V_λ and J_λ variants for V and J regions respectively in λ-light chain.

There is only one heavy chain type with three variable regions — V, D and J. To form each of them one gene fragment from V_h, D_h and J_h variants respectively is selected.



You need to count how many possible immunoglobulin molecules can be produced for given values of V_κ, J_κ, V_λ, J_λ, V_h, D_h and J_h.

Input

The first line contains two integers V_κ, J_κ ($1 \le V\kappa, J\kappa \le 1500$) — number of gene fragment variants for κ-light chain V and J variable regions, respectively.

The second line contains two integers V_λ, J_λ ($1 \le V_\lambda, J_\lambda \le 1500$) — number of gene fragment variants for λ-light chain V and J variable regions, respectively.

The third line contains three integers V_h, D_h and J_h ($1 \le V_h, D_h, J_h \le 1000$) — number of gene fragment variants for heavy chain V, D and J variable regions, respectively.

Output

Output one integer — number of immunoglobulin variants that can be produced.

Example

standard input	standard output
40 5 41 6 50 30 6	4014000

Note

In conclusion, we note that somatic recombination is powerful protective mechanism for various bacteria and viruses, but not only one. Other ways to prevent foreign invasion are beyond the focus of this problem.

Problem G. The Math of Sailing

Time limit: 1 second
Memory limit: 512 megabytes

Captain Polycarp has been dreaming about owning a three-mast sailing ship since he was a child. And at last, his dream is about to come true, he has saved enough money and bought a beautiful caracca "Pulcheria". However he overlooked the fact that it was sold without any sails on, so now he has to find them separately.

Caracca should have exactly four sails: one on the back mast, one on the front one, and two on the middle one. And Polycarp, coincidentally, has exactly four pieces of fabric with sizes t_1, t_2, t_3 and t_4. Polycarp can use those fabric pieces as sails in any order, making them smaller beforehand, if needed.

Let's denote the size of the front mast sail as a_1, the sizes of the middle mast sails as a_2 and a_3 and the size of the back mast one as a_4. The *maneuverability* of the ship is calculated as $a_1a_4 + a_2 + a_3$ and its *stability* is calculated as $a_1 + a_4 + a_2a_3$.

For the caracca to walk the sea safely captain has to adjust the sails in such a way that the maneuverability of the ship is equal to its stability. And for maximum comfort, these two values should be as large as possible.

To sum it up, Polycarp has to lower the values of $t_1, \ldots, t_4$ if needed, and then distribute the resulting sizes among the four sails in any order so that the following equality will hold and both of its sides will be maximal possible: $a_1a_4 + a_2 + a_3 = a_1 + a_4 + a_2a_3$.

Help him find a way to make the ship's maneuverability equal to its stability while maximizing both of these equal values.

Input

The first and only line of input contains four integers t_1, t_2, t_3 and t_4 — the sizes of pieces of fabric Polycarp has ($1 \le t_i \le 10^4$).

Output

In the first line of output print p — a permutation of integers from 1 to 4, the i-th number in it should specify from which piece of fabric was the i-th sail created. For example, if the second sail was created by decreasing the size of the fourth piece of fabric then $p_2 = 4$.

In the second line print space-separated values a_1, a_2, a_3 and a_4 — the resulting sizes of the sails ($1 \le a_1, a_2, a_3, a_4 \le 10^4$). These values don't have to be integers.

Your answer is accepted if the absolute error of the resulting maneuverability and stability compared to the correct answer does not exceed $2 \cdot 10^{-6}$ and both of those values differ from each other by no more than 10^{-6}.

Examples

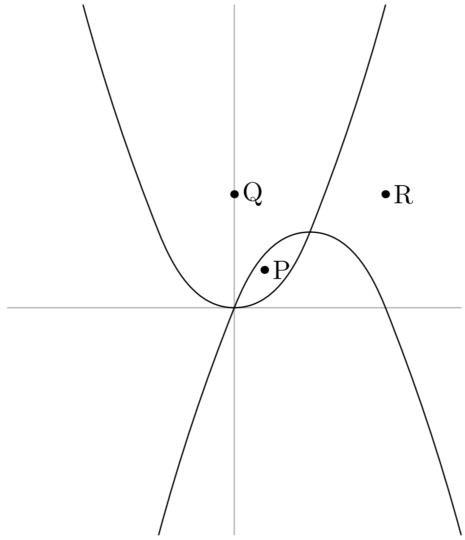
standard input	standard output
1 1 1 1	1 2 3 4 1.0 1.0 1.0 1.0
7 5 3 5	1 2 4 3 7.0 4.0 5.0 3.0
2 3 4 5	1 2 3 4 2.0 2.3333333333 4 5

Problem H. Lots of Parabolas

Time limit: 1 second
Memory limit: 512 megabytes

On a plane there is a set of parabolas given by equations in the form $y = a \cdot x^2 + b \cdot x + c$.

Let's consider a point to be located *inside* a parabola if it located above the parabola in case of positive coefficient a, or below the parabola in case of negative a.



On this figure, the point P is located inside both parabolas, the point Q is inside one of them, and the point R is inside none of them.

You need to find any point that is located inside all parabolas. It is guaranteed that such point exists.

Input

The first line contains a single integer n ($1 \le n \le 100\,000$) — the number of parabolas.

Each of the next n lines contains three integers a, b, c ($|a|, |b|, |c| \le 10^9$; $a \ne 0$), describing a parabola $y = a \cdot x^2 + b \cdot x + c$.

Output

Print two real numbers x and y — coordinates of a point located inside all parabolas.

The answer is considered correct if there exists a point at distance at most 10^{-6} from the printed one, which is located strictly inside all parabolas.

Example

standard input	standard output
4 1 2 3 1 -3 -5 -1 3 4 -2 4 6	0.24999999632501932 4.124999990812548

Problem I. Wheel of Fortune

Time limit: 2 seconds
Memory limit: 512 megabytes

Katya's old dream to be in the "Wheel of Fortune" game has come true.

Let's remind the rules of the "Wheel of Fortune" game:

- The goal of the game is to guess a word hidden by the game host. The contestants make turns by naming a letter.
- If the named letter occurs in the hidden word, then the host opens all its occurrences in the word, and the contestant can name a new letter.
- If the contestant names a letter that doesn't occur in the hidden word, the turn of the next contestant starts.

The host of the game hides the word of length L. Katya really wanted to win the game, so she hacked the game editor's computer. She found out that the hidden word will be one of N words.

Katya has the first turn in the game. Help Katya to understand if she can guarantee to win the game. Is it true that Katya will be able to guess any word from the stolen list without loosing the turn?

Input

In the first line there are two integers L and N — the length of the hidden word and the number of stolen words ($1 \le L \le 10^6$, $1 \le N \le 10^5$).

In the next N lines there are different words of length L. They consist of lowercase English letters.

It is guaranteed that the total length of all words does not exceed 10^6.

Output

Output "YES", if Katya is able guarantee herself a win, otherwise output "NO".

Examples

standard input	standard output
5 2 hello world	YES
4 4 idea play game warm	YES
4 2 game name	NO

Note

In the first sample Katya can win, if she first names the letter "l". Seeing the opened letters, she can determine he hidden word.

In the third sample for all Katya's strategies there is a risk of naming a wrong letter and loosing the turn.

Problem J. Yurik and Woodwork Lesson

Time limit: 1 second
Memory limit: 512 megabytes

Today Yurik got up early in the morning with a great joy because, according to the schedule, the first lesson in school is a woodwork lesson. It is the favorite Yurik's lesson because he usually ignores teacher and plays board games with his friends at the back of the classroom in this lesson. But he was disappointed when it turned out that the test would take place today.

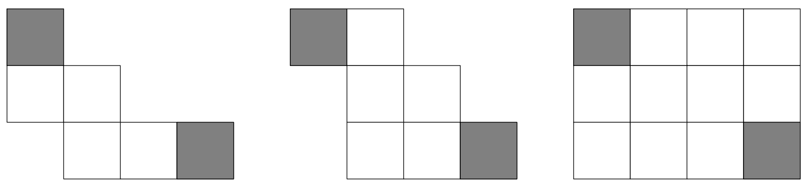
At the beginning of the lesson the teacher handed out wooden boards of size $N \times M$ to each student, including Yurik. Using lines drawn in pencil, the board is divided into N rows and M columns in such way that it consists of $N \cdot M$ square cells of size 1×1.

The test is to cut out some cells from the the board using a jigsaw in such way that the the resulting part of the board is *nice*.

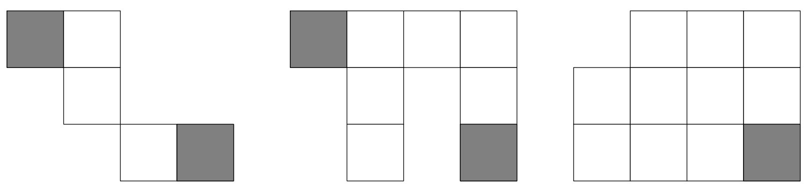
The board is called *nice* if the following five conditions are satisfied:

1. The upper-left cell of the board wasn't cut away.
2. The lower-right cell of the board wasn't cut away.
3. The resulting board is a connected area of cells. It means that every cell can be reached from any other cell using some number of steps. In one step we can move to the adjacent cell to the left, right, top or bottom.
4. For each row of the resulting board the following condition is satisfied: all cells in this row that weren't cut from the board, form a continuous horizontal segment of cells.
5. For each column of the resulting board the following condition is satisfied: all cells in this column that weren't cut from the board, form a continuous vertical segment of cells.

Every board that doesn't satisfy at least one of these conditions is called *ugly*. Images below illustrate examples of nice and ugly boards. The upper-left and lower-right cells are painted grey.



Examples of nice boards of size 3×4



Examples of ugly boards of size 3×4

Since Yurik never listens to his teacher and love mathematics, instead of doing the test, he wonders how many different nice boards can be obtained from the original board by cutting away some, possibly zero, number of cells? Two boards are considered different if sets of cut away cells are different.

Please, help Yurik to answer this question.

Input

The only input line contains two integers N and M — dimensions of the original board ($1 \le N, M \le 10^5$).

Output

Print a single integer — the number of different nice boards that Yurik can obtain by cutting away some cells from the original board.

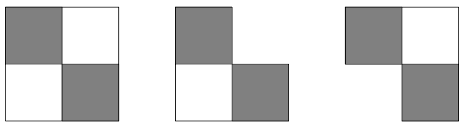
As the answer can be very large, you should print it modulo 998 244 353.

Examples

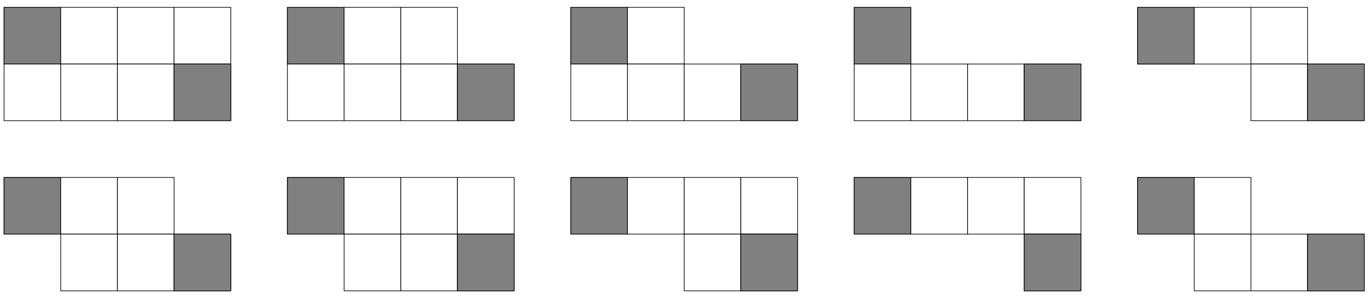
standard input	standard output
2 2	3
2 4	10
100 100	818380736

Note

Images below illustrate the first and the second examples:



All nice boards that can be cut from the original board of size 2×2

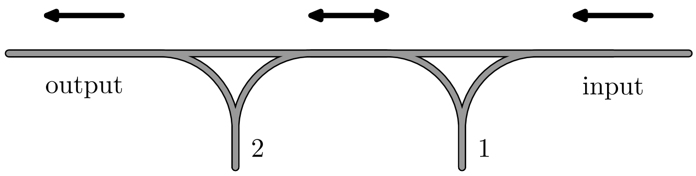


All nice boards that can be cut from the original board of size 2×4

Problem K. Railroad sorting

Time limit: 1 second
Memory limit: 512 megabytes

Arsenii works as an operator at a sorting station, the scheme of which is shown on the image.



The station has an input track, an output track and two dead ends. The operator can move cars between tracks and dead ends.

If car x is the first car on the input track, this car can be moved to any dead end. Command "1" moves the car to the dead end 1, and command "2" moves car to the dead end 2.

If the car x is the closest car to the exit in one of the dead ends, it can be moved to output track. Command "-1" moves the car from the dead end 1, and command "-2" moves the car from the dead end 2.

Finally, you can move cars between dead ends. If x is the closest car to the exit in one of the dead ends, it can be moved to another dead end. Command "12" moves the car from the dead end 1 to the dead end 2, and the command "21" moves the car from the dead end 2 to the dead end 1.

Please note that cars cannot be returned to a dead end from the output track and cannot be returned from a dead end to the input track. Also, you cannot move the car directly from the input track to the output track, it is required to use a dead end. Both dead ends can contain any number of cars.

A train of n cars arrives on the input track, each car has a unique number from 1 to n.

Arsenii must sort the cars so that they are all on the output track and their numbers from left to right are in ascending order. Help him form a sequence of commands that will help him to achieve this. The number of commands in the sequence must not be exceed $2 \cdot 10^6$.

Input

The first line of input contains integer n — the number of cars ($1 \le n \le 1000$).

The second line contains n different integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le n$) — the numbers on the cars in order from left to right on the input track.

Output

Print the sequence of commands that will cause the cars to be on the output track, and their numbers would be in ascending order. The sequence must contain no more than $2 \cdot 10^6$ command.

If there are multiple correct sequences, you can output any of them.

It is guaranteed that for any input data there is a sequence of commands containing at most $2 \cdot 10^6$ commands.

Example

standard input	standard output
5 3 1 5 2 4	1 1 -1 1 1 -1 12 -1 1 -1 -2

Problem L. Birthday

Time limit: 2 seconds
Memory limit: 512 megabytes

Bogdan received a birthday gift: a board game called "Subsegment sum". This entertaining game consists of n two-sided cards. An integer is written on each side of each card. The cards are arranged in a row on the table and are indexed from 1 to n, left to right. After the arrangement the cards can be turned over, but not swapped.

The player receives tasks, each task is a pair of numbers l and r. After receiving a task, the player places each card with index from l to r, inclusive, some side up. The target is to make the sum of the numbers on the upper sides of cards with index from l to r, inclusive, as large as possible.

Bogdan became bored with achieving maximum sums, so he decided to make the game harder. Now Bogdan selects a number k, and when he solves the task for cards from l to r, inclusive, he places these cards with some side up in such a way that the sum of the numbers on their upper sides was as large as possible, but not divisible by k. If Bogdan is able to solve this task, he denotes the received maximum sum as $f(l, r)$. If he is unable to select sides to make the sum on the upper sides indivisible by k, he considers $f(l, r) = 0$.

After some playing, Bogdan started thinking about the following problem. He wants to calculate the sum of $f(l, r)$ for all possible pairs l and r, in other words, calculate $\sum_{1 \le l \le r \le n} f(l, r)$.

Help Bogdan find this sum. Since the answer can be very large, calculate it by modulo $10^9 + 7$.

Input

The first line contains two integers n and k ($1 \le n \le 5 \cdot 10^5$; $1 \le k \le 10^9$).

Each of the next n lines contains a description of a card on the table: two integers a_i and b_i ($1 \le a_i, b_i \le 10^9$) — the numbers written on two sides of the card with index i.

Output

Output one integer, the answer taken modulo $10^9 + 7$.

Examples

standard input	standard output
3 3 1 2 2 3 3 1	23
5 5 4 1 4 2 2 3 2 4 1 5	130