Problem A. Balance

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	512 mebibytes

We say that a matrix A of size $N \times N$ is *balanced* if $A[i][j] + A[i+1][j+1] = A[i+1][j] + A[i][j+1]$ for all $1 \le i, j \le N - 1$.

You are given a matrix A of size $N \times N$. Your task is to output another matrix B of equal size such that B is balanced and $B[i][j] \ge A[i][j]$ for all $1 \le i, j \le N$. Furthermore, your B must have the minimum possible sum of entry values.

Input

The first line of input contains an integer N, the number of rows and columns of the matrix ($1 \le N \le 50$).

Each of the following N lines contains N integers. Together they describe the matrix A. It is guaranteed that $0 \le A[i][j] \le 35\,000$ for all $1 \le i, j \le N$.

Output

On the first line, print the sum of the values of the balanced matrix B you found. On the next N lines, print the balanced matrix in the same format as given in the input.

Any output matrix that meets the constraints described in the statement will be accepted. The values of the output matrix are not constrained in any way (specifically, they may exceed the value $35\,000$).

Example

standard input	standard output
4 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1	16 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

Problem C. Gravity

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	256 mebibytes

You are given an $N \times M$ binary matrix. Cell (i, j) contains the character "`.`" if it is free and the character "`#`" if the cell is occupied by a piece. Each maximal 4-connected component of "`#`" characters forms an indivisible piece. During the process described below, the pieces do not merge or split in any way. All cells that are part of the same piece will move in exactly the same way.

The pieces start falling towards the floor (the last row of the matrix) with equal speeds. The pieces move down without any rotations. Every second, all pieces try to move down one row. If this motion results in a piece crossing the lower boundary of the matrix, that piece stops in place instead. Similarly, if this motion results in a piece overlapping with another piece (note that this can only happen if the latter piece is not moving), then the former piece also stops in place. In other words, pieces stop falling when they hit the floor or when they hit another piece.

Output the final state of the pieces after all of them stop falling.

Input

The first line of input contains two integers N and M ($1 \le N, M \le 2000$).

The next N lines describe the matrix. Each of them contains M characters which are either "`.`" or "`#`". The characters denoting cells on the same line are **not** separated by any whitespace.

Output

Print the resulting matrix after all pieces have finished falling. The matrix must be printed in the same format as given in the input, except for the line containing the matrix dimensions.

Example

standard input	standard output
10 10 ....######.. ..#....#.. ..#.#..#.. ..#..#.#.. ..#....#.. ..######.. ....#....#.. ....#..	....######.. ..#....#.. ..#....#.. ..#....#.. ..#.##.#.. ..######.. ....#.. ..#....#..

Problem D. Infinite Pattern Matching

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 mebibytes

Consider the infinite binary string I formed by concatenating the binary representations of all the strictly positive integers in increasing order: $I =$ "$11011100\ldots$".

You are given a binary string A. Your task is to find the smallest integer L such that A is a suffix of $I[1 \ldots L]$.

Input

The only line of input contains the binary string A, $1 \le |A| \le 55$.

Output

Print a single line with a single integer: the number L.

Examples

standard input	standard output
11	2
011011	42

Problem I. Taxi

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	256 mebibytes

You are given an undirected tree on N vertices. Each edge has a length, which is a strictly positive integer. M taxis and M customers will appear in this tree, each taxi and each customer appearing in exactly one node. It is possible that a node will contain multiple taxis and/or multiple customers.

A taxi-app will match customers with taxis. These days, the customer must pay for the distance that the taxi travels just to pick them up. The taxi-app is mischievously greedy, so it will match customers with taxis such that the total distance travelled by taxis to their respective customers is as high as possible. Note that each taxi gets assigned to exactly one customer and each customer is assigned to exactly one taxi.

There are N^{2M} different ways in which the taxis and customers may appear in the tree. For each of these ways, we can find the total distance travelled by taxis according to the mischievously greedy matching picked by the taxi-app. Your task is to add all these distances together and compute this sum modulo $10^9 + 7$.

Input

The first line contains two integers, N and M ($1 \leq N, M \leq 2500$).

Each of the next $N-1$ lines contains three integers: x, y and l. This means that there exists an undirected edge between nodes x and y of length l ($1 \leq l \leq 10\,000$). It is guaranteed that the given edges form a tree.

Output

Print a single line containing a single integer: the required sum modulo $10^9 + 7$.

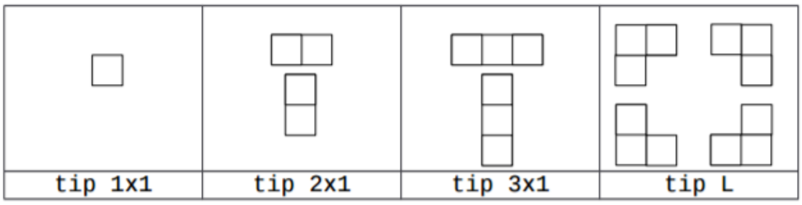
Example

standard input	standard output
5 2 4 5 9805 3 4 2001 2 3 6438 1 3 3790	10784056

Problem J. Tris

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 mebibytes

You are given some domino-like pieces. The following types of pieces are possible:



Note that there are only four types, and you may rotate and reflect any piece for further use. You want to place all the pieces in a matrix of size at most 800×800 so that you get a single non-self-touching cycle. Formally, this means:

- All pieces must fully fit in the matrix and be aligned with the grid.
- No two pieces may overlap.
- If a certain matrix cell is occupied by a piece, then exactly two of its four neighbours must also be occupied.
- All occupied cells are connected. In other words, you can travel from any occupied cell to any other occupied cell by only moving to adjacent occupied cells.
- The "interior" of the cycle must be a single 4-connected area.

Input

The input consists of a single line containing four integers: the number of pieces of each type (in the order they are shown in the image). It is guaranteed that each number is at least 2 and at most 100, and that at least one valid answer exists.

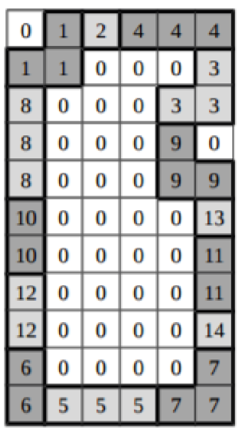
Output

The first line of output must contain two integers N and M ($N, M \leq 800$) denoting the number of rows and columns in your matrix. The next lines must describe the matrix in the following format:

- The matrix must contain integers between 0 and the total number of pieces, inclusive.
- Cells occupied by the same piece must have the same value.
- Cells occupied by different pieces must have different values.
- Cells that are not occupied by any piece must have the value 0.

If there are several valid answers, print any one of them.

Example

standard input	standard output	picture
3 4 3 4	11 6 0 1 2 4 4 4 1 1 0 0 0 3 8 0 0 0 3 3 8 0 0 0 9 0 8 0 0 0 9 9 10 0 0 0 0 13 10 0 0 0 0 11 12 0 0 0 0 11 12 0 0 0 0 14 6 0 0 0 0 7 6 5 5 5 7 7	

Problem M. Bus

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	256 Mebibytes

You need to solve problems like this one:

A bus leaves the bus depot with 4 passengers aboard. At the first stop, 3 more passengers get on. At the next stop, 1 passenger gets off and 2 get on. How many passengers are now on the bus?

This problem presents you with a number of scenarios of this nature. All you have to do is state how many passengers are on board the bus at the end of the scenario.

Input

Input file contains one scenario and begins with a route number (up to 5 numbers or letters with no spaces) and Z — the size of the bus (maximum number of passengers, $10 \leq Z \leq 100$) separated by a space. The 2nd line of the scenario gives P — the initial number of passengers on the bus ($0 \leq P \leq Z$). The 3rd line of the scenario gives S, the number of stops that are to be considered ($1 \leq S \leq 100$). There then follow S lines each containing 2 integers separated by spaces. The first number represents the number of passengers getting off at the stop, the second the number of passengers waiting to board. If the number of passengers waiting to get on is greater than the number of available seats, after the alighting passengers have got off, then the excess passengers are left behind. Health and Safety regulations prohibit the carrying of standing passengers.

Output

Output file should show the route number exactly as input, followed by a space, followed by the number of passengers aboard the bus at the end of the scenario.

Example

standard input	standard output
28HZC 26 4 2 0 3 1 2	28HZC 8

Problem N. Ancestry

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	256 Mebibytes

When some person A_0 descends from some person B_0 living centuries ago, there is not necessarily a unique line of ancestry. Far relatives may get married without even knowing that they are relatives. In such a case, it would be interesting to find the line of ancestry from A_0 to B_0 containing as few women as possible.

You are asked to find this line of ancestry, from a database with ancestor information. In particular, you are asked for the smallest number of women on a line of ancestry from A_0 to B_0.

For the sake of simplicity, the database does not contain real names, but numbers between 1 and a certain maximum N to identify individuals.

Input

Input file has the following format:

- One line with one integer N , satisfying $2 \le N \le 10^5$: the number of persons in the database.
- One line with two different integers A_0 and B_0 , satisfying $1 \le A0, B0 \le N$: the ID numbers of the two persons we are interested in.
- N lines, each with two integers f and m, satisfying $0 \le f, m \le N$. The integers on line i are the ID numbers of the father (f) and the mother (m) of the person with ID number i. An ID number f = 0 (or m = 0) indicates that the father (or mother) of person i is not known.

Integers on the same line are separated by single spaces. The integers f and m on the N lines are gender-consistent: no person is registered both as father and as mother (of the same or different persons). We do not assume age limits; for example, a man and a woman who are ten generations apart, may have a child together. Of course, the ancestor information in the input does not describe cycles.

Output

Output file should contain a single line, containing the smallest number of women on a line of ancestry from A_0 to B_0 (not counting A_0 and B_0 if either of them is a woman) if B_0 is an ancestor of A_0 according to the database, or containing the string `'no ancestor'` otherwise.

Example

<table>
<tr><th>standard input</th><th>standard output</th></tr>
<tr><td>23
1 8
2 0
3 15
9 4
5 10
11 6
0 7
8 0
0 0
0 0
0 0
12 14
13 0
0 0
0 0
16 23
17 20
0 18
19 0
12 7
21 0
22 0
8 0
0 0</td><td>2</td></tr>
<tr><td>9
2 9
5 2
3 4
0 0
0 0
6 7
0 0
8 9
0 0
0 0</td><td>no ancestor</td></tr>
</table>

Problem O. Olympiad

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	256 Mebibytes

Ministry of Education of Byteland announced that Byteland IOI team selection this year will take place a little differently. The process will start by coach announcing a number, k, and then have all the students stand in a circle. Coach will then, starting with the first student in the circle, move around the circle removing every k-th student until the circle is empty. The last four students removed from the circle will form the team. You decide that if you want to make the team, you will have to act fast and ensure you stand in one of the four selected positions in the circle. Your task is to write a program that, given the number of students in the circle and the number k, will determine the last four positions to be selected by coach.

Input

The input consists of a single line consisting of two integers, N and K, separated by a single space ($4 \le N \le 10^8,\ 1 \le K \le 10^8$). N represents the number of students in the circle, while K represents how often coach removes a student.

Output

Output should consist of a single line with four integers, each separated by a single space. These integers are the last four positions in the circle to be removed by coach. These positions should be listed in the order in which coach removes them.

Example

standard input	standard output
9 3	5 2 7 1
11 7	6 1 4 5

Problem P. Anagrams

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	256 Mebibytes

Two words are called ¡¡anagrams¿¿ if they contain the same letters but in a different order. For example `word` and `drow` are anagrams, but `word` and `worm` are not. In this problem you will be supplied with a set of words. All you have to do is to pick out the word with the most anagrams and report how many anagrams it has.

Input

Input begins with a number n, which is the number of words in the set ($0 \le n \le 1000$). Set contains n words, each on a single line. All words consist of lower case letters only. No word will contain more than 6 letters. It is guaranteed that set will contain at least one word that has an anagram in the list.

Output

Output consists of one word — the word with the most anagrams within the set, followed by a space, followed by the number of anagrams in set (excluding word itself). The word displayed will be the first occurrence in the input file of the anagram. If more than one word has the same highest number of anagrams, display only the one that comes first in the input file.

Example

standard input	standard output
6 nat cat act out tac ant	cat 2
6 worm word galo drown goal drow	word 1

Problem Q. Banking

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	256 Mebibytes

The bankers of Somebank have collected day-to-day data of the daily profits (and losses) of the shares they hold. Based on these numbers they decide to calculate which days would have been the most profitable to buy and sell their shares, so they can compare that with their actual gain.

Your task is to write a program that computes the optimal ¡¡run¿¿ in the sequence of numbers, the subsequence that maximizes the profit. The subsequence you compute is represented by the 1-based indexes of the first and last numbers in the subsequence. We are asking for exactly one date to buy and one date to sell shares since otherwise the solution would be simple: keep your shares on dates that the profit is non-negative.

Input

Input file has the following format:

- One line with a single integer N, $(1 \le N \le 10^6)$ — the length of the sequence to follow.
- One line with N integers p_i $(-1000 \le p_i \le 1000)$ — the profit (or loss) on date i.

The integers are separated by single spaces. At least one of the integers is positive.

Output

Output file contains a single line with two integers k and l $(1 \le k \le l \le N)$, such that the sum of the k-th until the k-th integer is maximized, boundaries included. When k and l can be selected by different ways, choose minimal k and minimal l.

Example

standard input	standard output
11 -3 1 -1 2 3 1 -1 2 -3 -5 7	2 8
9 1 -2 3 -1 -1 3 -2 2 -4	3 6

Problem R. Dimensions

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	256 Mebibytes

Teacher has given his pupils a table of widths, heights, lengths and volumes of some cuboid. Each row on the table has one of the 4 values missing; the pupils — and your program — have to work out the missing value and write it in the table such that the values on each line represent the width, height, length and volume of one cuboid.

Input

Input is a series of lines, each containing 4 integers: w, h, l and v representing the width, height, length and volume of a cuboid in that order. The integers are separated by a single space. $0 < l, w, h < 100, 0 < v < 10^5$. In each row, one of the values has been replaced by a zero. The final row contains 0 0 0 0 and should not be processed.

Output

Output is the same series of lines but with the zero in each line replaced by the correct value for length, width, height or volume as appropriate. It is guaranteed that the new value is always an integer.

Example

standard input	standard output
1 0 2 6 5 5 5 0 0 2 2 80 8 0 9 576 0 0 0 0	1 3 2 6 5 5 5 125 20 2 2 80 8 9 9 576