

Problem A. Abstract Picture

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

Famous abstract painter Va Sya plans to start new painting. It will be composed as square with grid $n \times n$, where each unit square is painted by some color.

Va Sya already defined the colors for some unit squares. Color of other squares does not matter for him.

For this work Va Sya is planning use the continuous technics: he paints whole row or whole column in some color. Moreover, each row and each column must be painted exactly once, so each unit square will be painted twice and its final color will be the last of two used colors.

Help Va Sya to find appropriate sequence of paints.

Input

First line of the input contains one integer n — length of the painting side in units ($1 \leq n \leq 3000$).

Each of the next n lines contains n characters. If i -th character in j -th line equals to '?', it means that color of i -th cell in j -th row of painting does not matter. Otherwise it contains lowercase English letter from 'a' to 'z' inclusively, which represents the color of corresponding cell (it is well known that Va Sya uses only 26 colors).

Output

Print $2n$ lines, i -th of those lines contains description of i -th paint in the following format:

«h y c » — row y is painted with color c ;

«v x c » — column x is painted with color c .

Rows are numbered sequentially upside down, columns are numbered sequentially leftside right, so upper left corner is on intersection of row 1 and column 1. Each row and each column must be mentioned in the output exactly once.

You may assume that there exists at least one solution for the given input. If there are several correct solutions, print any of them.

Example

standard input	standard output
3	h 1 p
ac?	h 3 q
ab?	v 2 c
?cz	h 2 b
	v 1 a
	v 3 z

Problem B. Battle Mage

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

Battle mage Va Sya bought a silver plate with form of convex polygon with n vertices to practice his spells.

Then he glued the plate on the thin glass of same form and shape, put this plate as a target and started his practice. Today Va Sya practices a spell which makes a perfect circular hole in the plate itself, but keeps the glass untouched. Note that circles may overlap.

After Va Sya ended up his practice session, he noticed, that all circles from the spells lied inside the plate (but may touch it's edge).

Then he decided to hang the whole construction (glass with remaining parts of silver plate) on the door using selected vertex, so it can rotate freely around this vertex until it is stable.

Find the final coordinates of vertices of the plate, if glass is so thin that its weight must be considered as zero.

Input

First line of the input contains three integers n , c and v ($1 \leq n \leq 700$, $1 \leq c \leq 2000$, $1 \leq v \leq n$) — number of vertices in the polygon, number of spells casted by Va Sya and 1-based number of the vertice he used to hang the remaining plate.

i -th of next n lines contains two integers x_i and y_i — coordinates of the i -th vertice ($-10^4 \leq x_i, y_i \leq 10^4$). Vertices are listed in counterclockwise order.

Each of next c lines contains three integers cx_i , cy_i and r_i — coordinates of center and radius of one circle. It is guaranteed that all circles are non-degenerated and lie inside the polygon (possibly touches it's edge).

Output

Print n lines, i -th of then containing x and y — final coordinates of i -th vertice with absolute error 10^{-5} or less. Vertices must be listed in same order as in the input file.

Example

standard input	standard output
4 1 3	2.253456 -1.176442
0 0	4.714949 0.538507
3 0	3.000000 3.000000
3 3	0.538507 1.285051
0 3	
1 2 1	

Problem C. Constant Ratio

Input file: `standard input`
Output file: `standard output`
Time limit: 1 second
Memory limit: 256 megabytes

Given an integer n , find out number of ways to represent it as the sum of two or more integers a_i with the next property: ratio a_i/a_{i-1} is the same positive integer for all possible $i > 1$.

Input

Input consists of one integer n ($1 \leq n \leq 10^5$).

Output

Print one integer — number of representations.

Examples

standard input	standard output
1	0
5	2
567	21

Note

In the first sample no such representation exists.

In the second sample there exist two representations:

- 1 1 1 1 1, then $q = 1$.
- 1 4, then $q = 4$.

Problem D. `dir -C`

Input file: `standard input`
Output file: `standard output`
Time limit: 1 second
Memory limit: 256 megabytes

Famous Berland coder and IT manager Linus Gates announced his next proprietary open-source system “Winux 10.04 LTS”

In this system command “`dir -C`” prints list of all files in the current catalog in multicolumn mode.

Lets define the multicolumn mode for number of lines l . Assume that filenames are already sorted lexicographically.

- We split list of filenames into several continuous blocks such as all blocks except for maybe last one consist of l filenames, and last block consists of no more than l filenames, then blocks are printed as columns.
- Width of each column w_i is defined as maximal length of the filename in appropriate block.
- Columns are separated by $1 \times l$ column of spaces.
- So, width of the output is calculated as $\sum(w_i + 1) - 1$, i.e. sum of widths of each column plus number of columns minus one.

Example of multi-column output:

```
a      accd e t
aba    b    f wtrt
abacaba db   k
```

In the example above width of output is equal to 19.

“`dir -C`” command selects minimal l , such that width of the output does not exceed width of screen w .

Given information about filename lengths and width of screen, calculate number of lines l printed by “`dir -C`” command.

Input

First line of the input contains two integers n and w — number of files in the list and width of screen ($1 \leq n \leq 10^5$, $1 \leq w \leq 10^9$).

Second line contains n integers f_i — lengths of filenames. i -th of those integers represents length of i -th filename in the lexicographically ordered list ($1 \leq f_i \leq w$).

Output

Print one integer — number of lines l , printed by “`dir -C`” command.

Examples

standard input	standard output
11 20 1 3 7 4 1 2 1 1 1 1 4	3

Problem E. Extreme Permutations

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

Permutation of length n is called the sequence of n integers, containing each of integers between 1 and n exactly once. For example, $(3, 4, 5, 1, 2)$ and $(1, 2)$ are permutations, $(1, 4, 3)$ and $(2, 1, 3, 2)$ are not.

Lets call the permutation extreme, if for any two neighbor numbers in the permutation difference between them is not less than minimum of those 2 numbers. For example, permutation $(3, 1, 2, 4)$ is extreme, because $|3 - 1| \geq \min(3, 1)$, $|1 - 2| \geq \min(1, 2)$ and $|2 - 4| \geq \min(2, 4)$.

Given an odd n , calculate number of the extreme permutations of length n where positions of some integers are fixed.

Input

First line of the input contains one integer n — length of permutation ($1 \leq n \leq 27$, $n = 2k + 1$ for some integer k).

Second line contains n integers p_1, p_2, \dots, p_n ($0 \leq p_i \leq n$). If p_i is equal to 0, then i -th position is not fixed, otherwise on i -th position must stay p_i . You may assume that if $p_i > 0$ and $p_j > 0$ for $1 \leq i, j \leq n$, $i \neq j$, then $p_i \neq p_j$.

Output

Print one integer — number of the extreme permutations where all fixed elements are on their positions.

Examples

standard input	standard output
5 0 0 0 0 0	4
5 0 1 0 0 5	1

Problem F. Find the Length

Input file: **standard input**
Output file: **standard output**
Time limit: 4 seconds (5 seconds for Java)
Memory limit: 256 megabytes

For each vertex of given undirected weighted graph calculate the length of shortest simple cycle, which contains this vertex.

Input

First line of the input contains one integer n — number of vertices in the graph ($1 \leq n \leq 300$).

Each of n lines contain n integers, i -th integer in the i -th column is equal to 0 for any i . If for $i \neq j$ j -th integer in i -th line a_{ij} is equal to -1 , then vertices i and j are not connected, otherwise they are connected by the edge of weight a_{ij} ($1 \leq a_{ij} \leq 10^6$).

You may assume that graph does not contain self-loops and $a_{ij} = a_{ji}$ for any $1 \leq i, j \leq n$.

Output

Print n integers one per line. i -th of those integers must be length of the shortest simple cycle, containing i -th vertex. If no simple cycles contain i -th vertex, print -1 at corresponding line.

Examples

standard input	standard output
4	11
0 9 1 1	11
9 0 -1 1	-1
1 -1 0 -1	11
1 1 -1 0	

Problem G. Game “Minesweeper”

Input file: `standard input`
Output file: `standard output`
Time limit: 1 second
Memory limit: 256 megabytes

This is interactive problem.

Rules of the Minesweeper game are simple: given a grid 16×16 . Each cell in the grid either is empty or contains a mine. Goal of the game is to label all mines with the flags and open all empty cells. If player opens empty cell, it contains one integer — number of mines in neighbor cells. Two cells are neighbors, if their borders have at least one common point (so each internal cell have 8 neighbors). If a cell, containing an integer 0 is opened (by player or automatically), all neighbors of this cell are opened automatically.

Player can do the following operations:

1. Single click on unopened cell to open this cell. If this cell contains the mine, game is lost. Otherwise cell is opened.
2. Double click on the opened cell which contains an non-zero integer. If number of labels in the neighbor cells is equal to this integer, all non-labeled neighbors are opened. If labeling was wrong, game is lost.
3. Toggle a label on the unopened cell. If cell was not labeled, it is labeled as containing the mine. If cell was labeled, label is removed.
4. Finish the game.

If after the end of game all empty cells are opened and all cells with mines are labeled, player wins, otherwise game is lost.

You are given coordinates of cell, which does not contain a mine. Moreover, it's guaranteed that it's possible to win the game without any nondeterministic situation (where guessing is needed). Your goal is to win.

Input

At the beginning of interaction your program must read two integers — row r and column c of the safe cell ($1 \leq r, c \leq 16$).

Then on each your request you receive next information:

First line of the answer contains one integer N — number of cells with changed status. Each of next N lines contain two integers r_i and c_i — row and column of the cell, and s_i — status of the cell: digit between 0 and 8, if the cell is empty, '*', if the cell was labeled or '-', if label from cell was removed.

Output

If you want to finish the game, print ("4") and end of the line. Otherwise print three space-separated integers: type of operation, row and column (1-based) of the cell to which it is applied. Do not forget to add end of the line character and flush the output buffer.

Example

standard input	standard output
1 1	1 1 1
6	
1 1 0	
1 2 0	
2 1 1	
2 3 1	
1 3 0	
2 2 2	3 3 2
1	
3 2 *	3 3 2
1	
3 2 -	3 3 1
1	
3 1 *	2 2 1
1	
3 2 2	3 3 3
1	
3 3 *	4

Note

Sample in the statement contains example of the interaction protocol for board 3×3 .

Problem H. Hierarchy

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

The hierarchical structure of the new office of Galactical Ministry of Bureacracy is very complicated. Office has 10^9 departments, numbered by sequential integers from 1 to 10^9 .

Initially all departments are empty. Then two types of the events may happen:

1. Arrival: new person is moved from old offices into some department. For this person are known his name and birthday. Assume this event is k -th arrival event for this department. Then person's local id is set to k .
2. Departure: selected person in some department with local id k is moved out this department to the old offices.

The head of each department is defined by the following way:

- If some person is strictly older, than than all other people who are currently working in this department, he became the head of the department.
- In case of tie (two or more oldest employees), the one with least local id in this department became the head of the department.

The head of office is defined by the following way:

- If some person who currently works in new office is strictly older, than all other people who currently are working in the new office, this person is the head of the office.
- In case of tie person, who is working in the department with least number, have priority.
- In case of tie person with least local id on this department have priority.

You are asked to write a program which after each event prints name of the head of office and head of department, which is related to this event.

Input

First line of the input contains one integer n — number of events ($1 \leq n \leq 10^5$). Then n events follow.

First line of the event description contains one integer t — type of event ($|t| = 1$).

If $t = 1$, then new person is moved to the new office. Then second line contains one integer D — number of department, where new person is coming ($1 \leq D \leq 10^9$), next line contains non-empty string composed of no more than 10 lowercase English letters — name of the person, and next line contains his date of birth in Unified Decimal Galaxy Time format $dd:mm:yyyy$, where dd is for day, mm — for month and $yyyy$ — for year, $00 \leq dd, mm \leq 99$, $0001 \leq yyyy \leq 9999$. You may assume that all names in the input file are pairwise distinct.

If $t = -1$, then some person is moved back to the old offices. In this case second line contains two integers D and k — number of department and local id of person who is moved ($1 \leq D \leq 10^9$). It is guaranteed that k does not exceed total number of people who were moved to this department at the moment of event and that all k in the requests with $t = -1$ are pairwise different.

Output

After each event print two space-separated strings: name of the head of the new office and name of head of department, which is related to this event. If new office or department contains no employees, printf "Vacant" instead.

Example

standard input	standard output
8	rab rab
1	rab tor
10	tor Vacant
rab	tur tur
01:01:0001	tor Vacant
1	Vacant Vacant
1000000000	bor bor
tor	bor bor
02:01:0001	
-1	
10 1	
1	
10	
tur	
01:01:0001	
-1	
10 2	
-1	
1000000000 1	
1	
5	
bor	
99:99:9999	
1	
5	
rot	
99:99:9999	

Note

Lets check what happened in the sample.

First event: in department 10 arrived new employee *rab*. He got local id 1 in this department, became the head of department and head of office.

Second event: in office 1000000000 arrived new employee *tor*. He got local id 1 in this department, became the head of department, but because *rab*'s birthday is 01 : 01 : 0001, and *tor*'s only 02 : 01 : 0001, *tor* is younger and does not became the head of office.

Third event: from department 10 removed employee with local id 1, i.e. *rab*. Department is empty, so we are printing "Vacant". In other departments only *tor* is working, so he became the head of office.

Fourth event: in department 10 arrived new employee *tur*. He got local id 2 in this department, became the head of department (because department was empty) and became the new head of the office, because his birthday is 01 : 01 : 0001, so he is older, than *tor*.

Fifth event: from department 10 removed employee with local id 2, i.e. *tur*. Department is empty, so we are printing "Vacant". In other departments only *tor* is working, so he became new head of office.

Sixth event: from department 1000000000 removed employee with local id 1, i.e. *tor*. Department is empty,

so we are printing “**Vacant**”. Office is empty, so we are printing “**Vacant**”.

Seventh event: in department 5 arrived new employee *bor*. He got local id 1 in this department, became the head of department and head of office.

Eight event: in department 5 arrived new employee *rot*. He got local id 2 in this department. His birthday is the as the *bor*’s, but his local id is greater, so *bor* keeps his position as head of the department. Similarly, *rot* have same birthday as *bor*, they are working in same department, but his local id is greater, so *bor* keeps his position as head of the office.

Problem I. Interactive Casino

Input file: `standard input`
Output file: `standard output`
Time limit: 1 second
Memory limit: 256 megabytes

This is interactive problem.

In the Interactive Casino game “Binary Roulette” is very popular. Here are the rules of the game.

- Initially the player has 160 tokens.
- In one turn player can bet any positive integer amount of tokens, as long as it does not exceed number of tokens player currently has.
- Inside of slot machine an integer x between 0 and $2^{20} - 1$ is generated. If sum of bits of this integer is odd, then player wins (i.e. returns his bet and additional k tokens), otherwise the player loses his bet. Note that player does not know value of x .
- If the player has zero tokens, game is lost.
- If the player makes more than 200 bets, game is lost.
- If the player at some time has 200 tokens, he won the game.

You found on the Algoleaks site that each next integer on the slot machine is generated using the formula $x_i = 487237 * x_{i-1} + 1011807 \bmod 2^{20}$. Source of x_1 , unluckily, on this site is not revealed.

Your goal is to ensure the victory.

Input

Your program will receive on the input one integer — number of tokens You currently have or -1 in case when game is over by some reason.

Output

If you received -1 , immediately exit your program with code 0 (otherwise you may get the random verdict from the system). Otherwise, if you received integer $T > 0$, print one integer between 1 and T , inclusively — your next bet. Dont forget to print end-of-line character and **flush** the output.

Example

standard input	standard output
160	5
155	10
165	15
180	20
-1	

Problem J. Judgement

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

The jury of Berland regional olympiad in informatics does not trust to contest management systems, so the Berland regional programming contest is judged by the next way. There are n judges in the jury, each judge have authority a_i . When jury receives a diskette with the new solution, each jury member reads its source and votes “OK” or “WA”. If after the voting ends the sum of authorities of jury members, who think that solution is correct, is equal or greater to p , then solution is accepted, otherwise it is rejected.

Several jury members think, that current system is too complicated for them and proposed some changes: each jury member will have new authority b_i , and limit is changed to q ; then, in their opinion, calculations will be easier, and nothing more changes.

You are hired by the Department of Education of Berland region to check if the new system is equivalent to old one, i.e. that at any possible distribution of votes final verdict with the new and with the old parameters will be the same. If the systems differ, output an example of voting, when verdicts in both systems are different.

Input

First line of the input contains one integer n ($1 \leq n \leq 100$) — number of judges in the jury.

Second line contains $n + 1$ integers p, a_1, a_2, \dots, a_n ($1 \leq p, a_i \leq 10^6$) — the current acceptance limit and the current values of authorities of jury members, respectively.

Third line contains $n + 1$ integers q, b_1, b_2, \dots, b_n ($1 \leq q, b_i \leq 10^6$) — the new acceptance limit and the new values of authorities of jury members, respectively.

Output

If old and new systems are equivalent, print “YES”. Otherwise in the first line of output print “NO”, and in second print example of voting, with different verdicts in old and new systems. Voting is encoded with the string of length n , where i -th character is ‘1’, if i -th judge considered solution correct and ‘0’ otherwise.

If several answers are possible, print any of them.

Examples

standard input	standard output
3 8 4 5 6 2 1 1 1	YES
3 6 4 5 6 2 1 1 1	NO 001

Problem K. Krotek

Input file: `standard input`
Output file: `standard output`
Time limit: 6 seconds
Memory limit: 256 megabytes

Mole Krotek from famous Czech animation movie lives underground on 2D-plane on the depth ε .

Soon Krotek dug N tunnels for his new home. Each tunnel is the segment, given by coordinates of the endpoints. Tunnels can intersect each other only in the endpoint they share. No three endpoints are collinear.

Now Krotek wants to link all tunnels to the one network, connecting some of the existing endpoints by the new tunnels. New tunnels can intersect other new tunnels or the existing ones only in the endpoint they share.

Find out minimal total length of the new tunnels.

Input

First line of the input contains one integer N ($1 \leq N \leq 1000$) — number of existing tunnels.

Each of the next N lines contain four integers (x_1, y_1, x_2, y_2) — coordinates of the endpoints of the one existing tunnel. Coordinates does not exceed 10^4 by absolute value.

You may assume that tunnels does not intersect (except for case when they share a endpoint) and that no three endpoints are collinear.

Output

Print the minimum total length of the additional tunnels with absolute or relative error 10^{-6} or less.

Example

standard input	standard output
2 0 0 3 0 0 1 1 2	1.0000000000

Problem L. Liesbeth and the String

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

Little Liesbeth likes to play with strings. Initially she got a string of length n , consisting of letters 'a' only.

Then Liesbeth performs next operations with the string:

- If the first letter of the string is 'a', then she adds to the end of the string "bc", and after that removes first two letters of the string.
- If the first letter of the string is 'b', then she adds to the end of the string 'a', and after that removes first two letters of the string.
- If the first letter of the string is 'c', then she adds to the end of the string 'aaa' and after that removes first two letters of the string.

Liesbeth stops when she has the string of length 1. For example, if $n = 4$, she needs 6 operations :

aaaa \rightarrow aabc \rightarrow bcbc \rightarrow bca \rightarrow aa \rightarrow bc \rightarrow a

Liesbeth found that for some n number of operations is too big, and its hard to understand if the process is finite. So she asked You to write the program to help her.

Input

First line of the input contains one integer n ($2 \leq n \leq 10^6$) — length of the initial string.

Output

Print one integer — number of operations needed to obtain string of length 1. If process is infinite, print -1 instead.

Examples

standard input	standard output
4	6
3	24

Problem M. Matrix, The

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

...Neo finally understood how The Matrix is working.

The Matrix is the square grid $n \times n$; each cell of the grid contains 0 or 1. Next additional limitations are working for The Matrix:

- Each column contains at least one 1; ones in the column are “continuous”, i.e. no zeros between any two ones in the column can be found.
- Each line number of ones is between a and b , inclusively

The Matrix changes with time following next law: lets define state of The Matrix as sequence of n^2 elements $(a_{1,1}, \dots, a_{1,n}, a_{2,1}, \dots, a_{n,n})$, obtained from The Matrix by writing all its lines one by one. Then all states, corresponding to correct (i.e. conforming with limitations above) instances of The Matrix, are ordered lexicographically and in time t The Matrix have t -th state in the resulting list.

Neo is sure that he can reconstruct the Matrix in any moment of time. Can you do it?

Input

First line of the input contains four integers n , a , b and q ($1 \leq n \leq 10$, $1 \leq a \leq b \leq n$, $1 \leq q \leq 1000$) — dimension of The Matrix, parameters of The Matrix and number of queries, respectively. i -th of the next q lines contains one integer t_i — some moment of time ($1 \leq t_i \leq 10^{18}$).

Output

For i -th request print n lines, each containing n characters — The Matrix in the time t_i . If total number of correct instances of the Matrix is less than t_i , print “No such matrix.” without quotes. Separate answers on different requests with newline. Follow the sample for clarify.

Example

standard input	standard output
3 2 3 5	011
1	011
2	101
16	
34	011
35	011
	110
	101
	111
	101
	111
	111
	111
	No such matrix.