Problem A. First Occurence

Input file:	`first.in`
Output file:	`first.out`
Time limit:	2 seconds
Memory limit:	64 megabytes

Vasya is constructing a sequence consisting of digits 0, 1 and 2. He's written a single digit 0 and now he repeats the following steps: he takes the written sequence and copies it two times more, each time replacing 0 → 1, 1 → 2, 2 → 0 so from a sequence of n digits he obtains a $3n$-digit one. For example, the first two his steps are $0 \to 012 \to 012120201$. His goal is to find the first occurence of substring S in this sequence.

Write a program that will find the first occurence of substring S in the sequence being constructed to let Vasya have some sleep...

Input

The input file contains only one line consisting of digits 0, 1 and 2. The length of the line does not exceed 255 characters. The input line cannot be empty.

Output

Write to the output file the first position in this sequence where the substring S occurs. If the substring S does not occur in the Vasya's sequence, output -1.

Example

`first.in`	`first.out`
12	2

Problem B. Convex Subsequence

Input file:	`convex.in`
Output file:	`convex.out`
Time limit:	2 seconds
Memory limit:	64 megabytes

The sequence $a_1, \ldots, a_n$ is called *convex* if for any given $1 < i < n$ the inequality $a_i \le (a_{i-1} + a_{i+1})/2$ holds. If you mark the points on Cartesian plane with $x = i$ and $y = a_i$ and then connect $(1, a_1)$ with $(2, a_2)$, $(2, a_2)$ with $(3, a_3)$, ..., $(n-1, a_{n-1})$ with (n, a_n) and (n, a_n) with $(1, a_1)$, you will get a (possibly degenerate) convex polygon situated downwards of last drawn line. Of course a sequence containing less than three elements is declared to be convex.

You are given a sequence of integer numbers $b_1, b_2, \ldots, b_n$. Your task is to find its convex subsequence $b_{k_1}, b_{k_2}, \ldots, b_{k_l}$ where $k_1 < k_2 < \cdots < k_l$ and l is maximal possible.

Input

The first line of the input file contains the number of test cases K ($1 \le K \le 10$). The rest of file consists of K blocks containing two lines each. Only one integer number N ($1 \le N \le 200$) is written in the first line of each block. The second line of the block contains N integer numbers b_i separated by spaces. All b_i do not exceed 10000 by an absolute value.

Output

Your program is to produce two lines for each block. The first line contains one integer number l — the length of the maximal convex subsequence in the input sequence. The second line contains l numbers separated by a single space — the indices k_1, k_2, ..., k_l such that the sequence b_{k_1}, b_{k_2}, ..., b_{k_l} will be convex. Indices are to be displayed in ascending order.

Separate output for different blocks with a single blank line.

Example

`convex.in`	`convex.out`

2
3
2 3 5
4
2 3 6 8
	3
	1 2 3
	
	3
	1 2 4

Problem C. The King and the Rook

Input file:	`kingrook.in`
Output file:	`kingrook.out`
Time limit:	2 seconds
Memory limit:	64 megabytes

You are given a small chessboard 6 × 6. There are three pieces on it: a White King, a White Rook and a Black King. Your task is to calculate the minimal number of moves required for white side to checkmate the Black King or determine that it is impossible or that the position is incorrect.

Input

The only line of the input file contains three coordinates of fields — coordinates of White King, White Rook and Black King respectively, followed by an identifier of the side to move (`W` for white or `B` for black).

Output

Write to the output file total number of moves (for both sides) required for white to win the game. If the black king is already checkmated, output 0. If the input position is incorrect (two kings are adjacent or it is white move and the check is declared for black), output -1. If the game will end in a draw (e.g. black are stalemated), output -2.

Example

`kingrook.in`	`kingrook.out`
c6 f4 a5 B	2
c6 f4 b5 W	-1

Problem D. n-Dimensional Farm

Input file:	`farm.in`
Output file:	`farm.out`
Time limit:	2 seconds
Memory limit:	64 megabytes

Of course you know that n-dimensional Kittens live in the n-dimensional space and build true n-dimensional towns. In the town named Transborough their houses are located only in points with integer coordinates in range from 1 to n. When they planned to build this town, they had decided that all the points with at least two equal coordinates are bad for building a house, so now only points with different coordinates are occupied and there are no more points free for building a house now because someone lives at any good point.

Recently the local authorities have built a milk farm near the town. All Kittens visit this farm every day because they like milk very much. Of course any Kitten spends some time walking from home to the farm and back.

The Kittens are arguing who's the luckiest Kitten who lives at the minimal distance to the farm. Help them to establish the truth!

Input

The first line of the input file contains one integer number n ($1 \le n \le 64$). The second line is filled with n numbers separated by spaces. These numbers are coordinates of the milk farm. All coordinates are integers in range from -10000 to 10000.

Output

Write to the output file exactly n numbers separated by a single space — the coordinates of the Kitten's house nearest to the milk farm. If there is a tie, output any of the optimal answers.

Example

farm.in	farm.out
2 0 1	1 2

Problem E. La cucaracha

Input file:	cucarach.in
Output file:	cucarach.out
Time limit:	2 seconds
Memory limit:	64 megabytes

At every midnight in the flat of scientist Vasya a horror starts. Hundreds. . . oh no! thousands of cockroaches go out of every hole to his dinner table destroying all crumbs and leavings!

Vasya hates cockroaches. He thought for a very long time and now he keeps in his hands a supertrap which attracts all cockroaches in its very huge coverage area after activation.

He plans to activate this trap tonight. But there is one problem. This very effective trap with very huge coverage area consumes very large amounts of energy. So Vasya plans to minimize the time this trap will work. He collected the data about all points the cockroaches enjoy. Also he noticed that all the cockroaches move only on the lines of his checked table-cloth with a constant speed (we can suppose this speed is equal to 1, so the cockroach located at one of the junctions can move in one time unit to any vertically or horizontally adjacent junction). Vasya has decided to activate his trap in one of the junctions. When the trap is activated, all cockroaches will move to the junction containing it as fast as they can so in any unit of time the cockroach will move to the adjacent junction maximally decreasing its distance to the trap (if there is a tie, the cockroach may select any of the tying junctions).

Write a program for Vasya that will select a junction minimizing the amount of time needed to destroy all cockroaches.

Of course your program will assume the table-cloth to be the Cartesian plane and the junctions to be the points with integer coordinates.

Input

The first line of the input file contains the number of cockroaches' meeting points N ($1 \le N \le 10000$). The next N lines contain x and y coordinates of the points (integer numbers not greater than 10^9 by their absolute values).

Output

You have to write only two integer numbers — x and y coordinates of a junction that minimizes the trap work time. If there is more than one solution, output any of them.

Example

cucarach.in	cucarach.out

2
1 1
3 3
	2 2

Problem F. Windows

Input file:	windows.in
Output file:	windows.out
Time limit:	2 seconds
Memory limit:	64 megabytes

n rectangular windows have appeared on the screen. Your task is to find the area covered simultaneously by all windows displayed.

Input

The first line of the input file contains one integer number n ($1 \le n \le 30000$). The next n lines contain four integer numbers each: x_1 y_1 x_2 y_2, where (x_1, y_1) are coordinates of the upper-left corner of the window, and (x_2, y_2) are ones for the down-right. All coordinates do not exceed 5000 by an absolute value. Note that the y coordinates on the display grow downwards.

Output

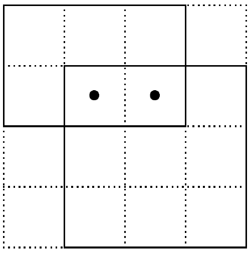
Write to the output file a single integer number — the area covered by all the windows displayed.

Example

windows.in	windows.out

2
0 0 3 2
1 1 4 4
	2

Example Picture



Problem G. A Turing Machine

Input file:	`turing.in`
Output file:	`turing.out`
Time limit:	2 seconds
Memory limit:	64 megabytes

The Turing machines are one of the easiest ways to illustrate the general conception of an algorithm. In this task you have to simulate a simple Turing machine with three cycled tapes with m cells on each.

This machine has n states enumerated with integers from 1 to n. There are some rules defined for each state (maybe zero) depending on symbols currently read from all three tapes. The possible symbols are 0, 1 and `E`.

So each rule has the following form: initial state; three symbols; three symbols to write to each tape; shift offset for first tape; for the second; and for the third; and the next state.

If the machine is in the initial state written in the rule and reads these three symbols from its three tapes by its read/write heads, it replaces them with the symbols specified by the rule, shifts all tapes by the specified offset and goes to the described next state.

If the machine does not have a rule for the current state and observed symbols, it aborts execution.

Also there is a special state, called *finishing state*. If the machine enters this state, the program immediately stops. This state always has number 0.

Input

The first line of the input file contains three numbers: m ($1 \le m \le 5000$), n ($1 \le n \le 200$) and q ($1 \le q \le n$) — initial state for the program. The next three lines contain m symbols each — the initial symbols written on the first, second and third tape respectively. Symbols are listed from the current read/write head position. Next line contains the number of rules K ($1 \le K \le 27n$). Each rule is described in the form specified above. Shift offset is a positive or negative number less than m by an absolute value. Positive number shifts the tape left (i.e. the symbol right to the symbol currently read will be read on this tape after shift). All fields are separated by a single space except groups of three symbols written without spaces between them. States are natural numbers in range $1 \ldots n$. The next state field in the rule may also take a value of 0 (the finishing state).

The machine may not have more than one rule for each state with a given set of symbols. It is guaranteed that the given machine will not perform more than 10^6 operations.

Output

If the machine successfully entered its finishing state, display the word `SUCCESS` in the first line. Otherwise (if the machine entered a state for which no rule can be found) display `ABORTED` in the first line.

The next three lines must contain the symbols written on each three tapes, starting from the symbol currently observed.

Example

turing.in	turing.out
4 2 1 EEEE 0011 11EE 3 1 E01 01E 1 -1 2 2 2 E1E EEE 1 1 1 2 2 111 000 2 3 -1 0	ABORTED E0EE 01EE E1EE

Problem H. The System Console

Input file:	`console.in`
Output file:	`console.out`
Time limit:	2 seconds
Memory limit:	64 megabytes

Andrew is writing the system console for the ACM automatic judging&testing system. Now he's working hard about the subsystem filtering some lines displayed by the system to a console. For example, the technical comitee members responsible for delivering printed jobs and notifications to contestants should see all system output related to printer.

Write the program which will filter all lines containing given substrings from a testing system output.

Input

The first line of the input file contains number of search strings N ($1 \le N \le 100$). The next N lines contain one non-empty search string each. Length of any line does not exceed 80 characters.

The rest of file contains the system output wherefrom the strings are to be filtered. The length of the whole input file is limited to 640 KBytes. The length of any testing system output line given in the input file may not exceed 250 characters. All characters in every string are ASCII characters with codes from 32 to 126. The system output may contain empty lines.

Output

Write to the output file all strings (in order they appear in the console output) containing at least one occurence of one or more given search strings. An occurence is supposed to be case-sensitive (exact symbol-to-symbol match).

Example

Note: some lines of the input and output may be split to fit the column.

console.in
1 Sending to printer 29.09.2002 13:05:02 judge #4 aborted his previous job 29.09.2002 13:05:04 judge #1: response for (2, 1): OK * ok * 29.09.2002 13:05:04 Saving state of contest 290902 29.09.2002 13:05:04 sent message "E(1): Wrong Answer, TestN=2" to 01, channel 11 29.09.2002 13:05:04 Sending to printer notification for 01, limit=2/30 29.09.2002 13:05:08 channel #11: message id 11_2_15706952026812138l for 01 acknowledged 29.09.2002 13:05:15 monitor (#1) connected at channel #37 29.09.2002 13:05:15 monitor #1: sending personal monitor data to 01 29.09.2002 13:05:15 monitor #1 disconnected from channel #37 29.09.2002 13:05:15 Saving state of contest 290902 29.09.2002 13:05:35 monitor (#1) connected at channel #37 29.09.2002 13:05:35 monitor #1: sending personal monitor data to 43 29.09.2002 13:05:35 monitor #1 disconnected from channel #37 29.09.2002 13:05:40 got from console #1: Msg="monitor", ID="" 29.09.2002 13:06:14 print client (#1) connected at channel #37 29.09.2002 13:06:14 Sending to printer file "e.pas" from 01, 473/31/1, limit=3/30 29.09.2002 13:06:14 print client #1: sending job confirmation. 29.09.2002 13:06:14 print client #1 disconnected from channel #37

console.out
29.09.2002 13:05:04 Sending to printer notification for 01, limit=2/30 29.09.2002 13:06:14 Sending to printer file "e.pas" from 01, 473/31/1, limit=3/30

Problem I. Quarks&Quirks

Input file:	quarks.in
Output file:	quarks.out
Time limit:	2 seconds
Memory limit:	64 megabytes

In the physical laboratories of the St. Petersburg State University our scientists have created a system consisting of n nodes connected with K pipes. Then they have launched a particle called "quark" to node i. This system works with two types of particles: "quarks" and "quirks"; in one unit of time the particle flies from one end of pipe to another. There are two types of pipes: the pipes of first kind do not change the flying particle, the rest transform quarks to quirks and vice versa.

They have learned as the result of the experiment that after M units of time there was a quark detected at node j. Calculate the number of different ways the launched quark might move during this experiment.

Input

The first line of the input file contains five integer numbers n, M, i, j and K ($1 \le i, j \le n \le 50$, $1 \le M \le 50$, $1 \le K \le 1250$). The next K lines describe one pipe each. The description of each pipe consists of three numbers u_i v_i t_i, where u_i and v_i are the nodes connected ($1 \le u_i \ne v_i \le n$), and t_i is the type of pipe (1 — does not change the type of particle, 2 — changes).

Output

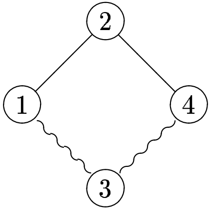
You must write only one integer number W — the number of different ways the launched quark might move.

Example

quarks.in	quarks.out

4 2 1 4 4
1 2 1
2 4 1
1 3 2
4 3 2
	2

Example Picture



Problem J. Isogonal Point

Input file:	isogonal.in
Output file:	isogonal.out
Time limit:	2 seconds
Memory limit:	64 megabytes

Consider a triangle lying on a plane. Someone has marked a point inside this triangle and connected this point by segments with all vertices. Then he's drawn the bisectors of this triangle and symmetrically reflected each of segments relatively to corresponding bisector (drawn from the same vertex of the triangle). It is known that three lines containing the reflected segments intersect at one point. This point is called *isogonal* to the original one.

Your task is to find the isogonal to a given point inside a given triangle.

Input

The first line of the input file contains number of test cases K ($1 \le K \le 10\,000$). Next K lines contain eight integer numbers each: x_1 y_1 x_2 y_2 x_3 y_3 x y where (x_1, y_1), (x_2, y_2) and (x_3, y_3) are coordinates of the triangle vertices (not lying on the same line), and (x, y) are the coordinates of marked point strictly inside the given triangle. All coordinates do not exceed $1\,000$ by an absolute value.

Output

For each test case write a single line with two real numbers rounded accurate to four digits after decimal point — x and y coordinates of the isogonal point.

Example

isogonal.in	isogonal.out
1 0 0 4 0 0 3 2 1	0.3333 0.6667

Example Picture