

Problem A. Lucky Controller

Input file: standard input
Output file: standard output
Time limit: 2 seconds
Memory limit: 64 Mebibytes

Egor works as a controller in the bus. Each day he is given a pack of tickets which he then sells. Recently he has become interested about how many tickets in the pack are lucky. He thinks that the more tickets are lucky the luckier day he will have. Now he wants to find out how lucky for him the next day is going to be. Each ticket number consist of n digits. The ticket is considered to be lucky if the sum of the first $n/2$ digits equals to the sum of the last $n/2$ digits. Egor knows that the numbers in the pack that he will be given can start with equal probability from any number in the interval from a to b inclusively. The pack holds k tickets. The numbers of the tickets are consecutive. Help Egor to find an expected quantity of lucky tickets in the pack.

Input

Input consists of a single line with three integers a , b and k ($0 \leq a \leq b < 10^{12}$, $1 \leq k \leq 10^5$). Integers a , b and $b+k$ consist of same amount of digits, and this amount equals to the amount of digits in the number of each ticket. They may start with zeroes. The amount of digits in a and b is always even.

Output

Output the expected quantity of lucky tickets in the pack in the form of irreducible fraction. In case the result is an integer, no slash should appear in the output.

Example

standard input	standard output
0123 4567 150	6519/635
10 10 20	2
4000 4999 11	103/125

Problem B. Bukazoids

Input file: standard input
Output file: standard output
Time limit: 2 seconds
Memory limit: 64 Mebibytes

Secret laboratory of Fatown has developed a new gluttonous robot which moves on stripe consisting of $n + 1$ cells. Cells are numbered from 0 to n . The robot is located at cell with number 0; each other cell contains several bukazoids which gluttonous robot regales oneself with. The robot can do m single jumps (to adjacent cell) and k double jumps (over one cell). Additionally, $m + 2k = n$. All jumps are jumps forward. To feed gluttonous robot you need to write a program which finds sequence of jumps with highest number of bukazoids on a way.

Input

The first line at the input contains 3 integers: n ($1 \leq n \leq 100$), m ($0 \leq m \leq 100$), k ($0 \leq k \leq 100$). The second line contains n integers — number of bukazoids (up to 100) in corresponding cells of the stripe.

Output

The first line at the output should contain highest number of bukazoids found. The second line should contain $m + k + 1$ integers — numbers of cells visited by the robot, starting from cell with number 0.

Example

standard input	standard output
5 1 2	13
5 2 7 3 1	0 1 3 5

Problem C. Collisions

Input file: **standard input**
Output file: **standard output**
Time limit: **2 seconds**
Memory limit: **64 Mebibytes**

Identical small balls are located on a straight line and can move along this line only. Each ball moves with a constant velocity, but velocities of different balls may be different. When two balls meet, a perfectly elastic collision occurs. It's a common-known physical fact, that when two equal-mass physical bodies A and B collide perfectly elastically, they swap their velocities, i. e. new A 's velocity is old B 's one, and new B 's is old A 's. Your task is to write a program to find the total number of collisions.

Input

The first line at input contains the number of balls N ($3 \leq N \leq 2 \cdot 10^5$). Each of the following N lines contains 2 space-separated integers — the starting coordinate and the velocity of corresponding ball. All start coordinates are in range $-10^{11} < x_i < 10^{11}$, all velocities are in range $-10^8 < v_i < 10^8$. All start coordinates are different. It's guaranteed that each collision involves exactly two balls (none involves three or more balls together).

Output

Your program should output exactly one integer in a single line — the total number of collisions (or 987654321987654321 if the number is infinite).

Example

standard input	standard output
3 -5 3 0 -1 7 -2	3

Problem D. Dancers

Input file: **standard input**
Output file: **standard output**
Time limit: **2 seconds**
Memory limit: **64 Mebibytes**

Alex missed the ballroom dance competition that he wanted to attend. So now he wants to know the pairs of dancers whose dancing he missed. He had several photos from the competition, so he chose one where all dancers are clearly visible and wrote down the coordinates of all N dancers (N is even). Then Alex determined the pairs of dancers by the following algorithm: from not yet paired dancers he chooses two closest (to each other) dancers and assumes that they dance together as a pair. Should he find several pairs of dancers with the same minimum distance between dancers, he chooses lexicographically smallest pair (Alex enumerated dancers by integer numbers from 1 to N , dancers are ordered inside a pair, one with lower number goes first). You are asked to help Alex to determine pairs of dancers.

Input

The first line of input contains even integer N ($2 \leq N \leq 300$). Each i -th line of the next N lines contains two integers — x and y coordinates of point, representing i -th dancer. All points are distinct. All coordinates are less than 10^8 by absolute value.

Output

You should output $N/2$ lines. Each line must contain numbers of dancers in the corresponding pair. The first number in a line should be less than the second. Lines must be sorted in the lexicographically ascending order.

Examples

standard input	standard output
6 0 2 3 2 0 0 1 0 0 -2 2 -2	1 2 3 4 5 6
4 0 0 1 1 0 1 1 0	1 3 2 4

Problem E. Embassy

Input file: **standard input**
Output file: **standard output**
Time limit: 2 seconds
Memory limit: 64 Mebibytes

A queue of TopCoders from all regions of Ukraine to US Embassy in Kyiv never gets empty. It is especially long near TopCoder Open visa deadline. Embassy official talks to each coder during one time unit. Some coders, due to long time in queue, are late to their trains and have to change their tickets. Newly formed Olexiy's TopCoder Fund recovers ticket change fee and therefore aims at reduction of total recovery amount. Let N be the quantity of coders in the queue; i — a sequential number assigned to each coder when entering the embassy; d_i — departure time of the i -th coder's train (measured in time units), w_i — ticket change fee for the i -th coder. It is guaranteed that a coder is still not late visiting US Embassy at time d_i . Your task is to put coders into the queue in the order that minimizes total recovery amount to be paid by Olexiy's TopCoder Fund.

Input

Input contains the number of coders N in the first line, followed by N lines containing d_i and w_i in the $(i + 1)$ -th line, separated with single space. All the numbers are positive integers and not greater than 30000.

Output

Output must contain the optimal queue as N lines, with i -th line containing number of coder standing at i -th position. If several optimal orderings exist you need to output any of them.

Example

standard input	standard output
7	2
3 40	1
2 60	5
6 10	6
1 30	3
4 70	4
4 50	7
4 20	

Problem F. Dictionary of Obscene Words

Input file: **standard input**
Output file: **standard output**
Time limit: **2 seconds**
Memory limit: **64 Mebibytes**

Given a dictionary of obscene words S_1, S_2, \dots, S_n and text T , find if text contains one of obscene words as subsequence. If it does, find smallest prefix of T that contains such subsequence.

Input

First line of input contains one integer n — number of words in dictionary. Following n lines contain words from dictionary, one per line. Each word consists of ASCII characters with codes from 32 to 127, inclusive. Next line contains text T , consisting of the same set of characters. Total length of all words in dictionary doesn't exceed 100 KiB (100×2^{10} bytes). Total size of input file doesn't exceed 1 MiB (2^{20} bytes).

Output

Output 'NO' if there is no obscene subsequence in the text. Otherwise output 'YES $\langle X \rangle$ ', where X is the length of smallest prefix of T that contains some obscene subsequence.

Examples

standard input	standard output
2 hello world abracadabra	NO
2 hello world zzzheluuuulottt	YES 12

Problem G. Greatest Product

Input file: **standard input**
Output file: **standard output**
Time limit: **2 seconds**
Memory limit: **64 Mebibytes**

September 10th is forester day. On this occasion Magic forest dwellers decided to make a party. Kuzma beaver is assigned to take care about intellectual games. Kuzma has invented at his opinion quite interesting game. Rules of the game are simple. Master announces positive integer N . Player needs to calculate product of digits for each positive integer up to N and report the greatest product. All answers should be known beforehand to allow interactive communication during a game. This causes problems since N could be relatively big ($1 \leq N \leq 2 \cdot 10^9$). Kuzma is not good using computers; he asks you to write a program which having positive integer N will find correct answer.

Input

Input file contains one integer N

Output

Output file should contain greatest product for given N .

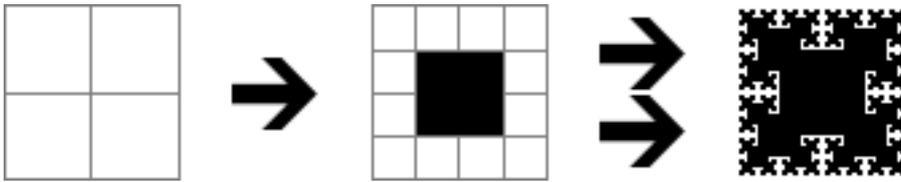
Examples

standard input	standard output
1	1
101090000	43046721
28994	10368
4876	2268
2789	1008

Problem H. Fractal Cake

Input file: standard input
 Output file: standard output
 Time limit: 2 seconds
 Memory limit: 64 Mebibytes

Fyodor celebrates his birthday today. Before the guests come he decorates a cake with chocolate cream in a special way. At the beginning cake looks like a square divided into 4 equal white square parts. Fyodor calls fractalization the following sequence of actions. All the little squares of cake get united into groups of 2×2 so that there are no ungrouped fragments. After that each small square is divided into 4 equal squares so that group of 2×2 becomes a group of 4×4 . As the last action 4 squares in the middle of each group are filled with chocolate. Fyodor does not stop at one fractalization and repeats it N times, even when he has to use a microscope. Illustration below shows the initial cake, first fractalization result, and the cake after the fifth fractalization:



Now Fyodor wants to cut pieces of cake with beautiful patterns for guests, but it is difficult to assess beauty of a piece looking at the whole cake. Fyodor wants a program that will quickly show the pattern of rectangular part of the cake.

Input

Input file contains one line with five non-negative integers: N, R_1, R_2, C_1, C_2 . N — the number of fractalization iterations ($N < 20$), R_1 and R_2 — first and last rows, C_1 and C_2 — first and last columns of the part. Following restrictions are also met: $R_1 \leq R_2, C_1 \leq C_2; 0 \leq R_2 - R_1, C_2 - C_1 < 100; 0 \leq R_1, R_2, C_1, C_2 < 2N + 1$.

Output

Output should contain $R_2 - R_1 + 1$ lines each containing $C_2 - C_1 + 1$ characters. Each symbol corresponds to a square and should be 1 in case it's filled with chocolate and 0 otherwise.

Examples

standard input	standard output
1 0 3 0 3	0000 0110 0110 0000
2 0 3 0 3	0000 0110 0111 0011
13 50 55 95 100	101111 100111 100111 101111 101101 100001

Problem I. Hexaroman Numbers

Input file: **standard input**
Output file: **standard output**
Time limit: **2 seconds**
Memory limit: **64 Mebibytes**

Roman numerals are numeral system of ancient Rome based on letters of the alphabet, which are combined to signify the sum (or in some cases, the difference) of their values. This system is decimal but not directly positional, since same digits standing at different positions in usual decimal system are represented by different Roman digits, and one decimal digit could be represented by few Roman.

There are seven Roman numerals associated to the decimal $I = 1$, $V = 5$, $X = 10_{10}$, $L = 50_{10}$, $C = 100_{10}$, $D = 500_{10}$, $M = 1000_{10}$.

Generally, Roman numerals are written in descending order from left to right, and are added sequentially, for example **MMX** (2010) is interpreted as $1000 + 1000 + 10$. Certain combinations employ a subtractive principle, which specifies that where a symbol of smaller value precedes a symbol of larger value, the smaller value is subtracted from the larger value, and the result is added to the total. For example, **MCMXLIV** equals 1944. **I** may precede **V** and **X**, **X** may precede **L** or **C**. The numerals **V**, **L**, and **D** may not be followed by a numeral of greater or equal value.

Subtractive principle has been introduced at medieval ages. Originally it allows only one symbol of smaller value to precede a symbol of larger value. Nowadays this limitation could be skipped to allow shorter notation for some numbers. Also modern computers are much better in hexadecimal numbers and probably it makes sense to introduce «hexadecimal» Roman notation.

Let hexadecimal Roman notation be a notation in which Roman numerals are equivalent to the following numbers: $I = 1$, $V = 8$, $X = 10_{16}$, $L = 80_{16}$, $C = 100_{16}$, $D = 800_{16}$, $M = 1000_{16}$. In some hexadecimal digit can be represented using addition and subtraction, notation with less symbols is used. If both notations produce same number of symbols addition is used. For example number F_{16} is written as IX_{16} , $5C_{16}$ becomes $CCDLXXXV_{16}$.

You need to write a program which can perform computation in hexadecimal Roman notation, namely the following operations: addition, subtraction, and multiplication. All source numbers and results will be integers between 1 and $4FFF_{16}$, inclusively.

Input

The first line at input contains one integer N ($0 < N \leq 100$), the number of test lines. Each line contains test data in format $\langle A \rangle \langle O \rangle \langle B \rangle$ without any spaces. $\langle A \rangle$ and $\langle B \rangle$ are numbers in hexadecimal Roman notation, $\langle O \rangle$ designates operation: +, -, *.

Output

Every line at output should contain hexadecimal Roman notation of calculation result for corresponding test line at the input.

Example

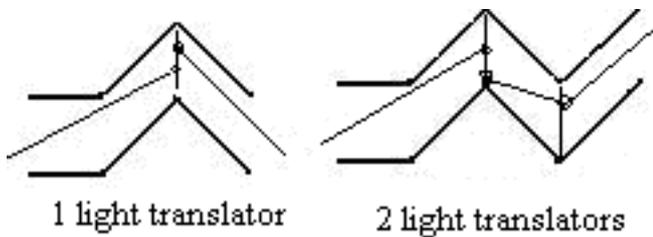
standard input	standard output
3	XXXLV
XIIV+XXXXII	CDXXVIII
XXIIII*XXXIII	CXI
XXIV*IV	

Problem J. Beam in Tunnel (Div 1 Only!)

Input file: standard input
 Output file: standard output
 Time limit: 2 seconds
 Memory limit: 64 Mebibytes

A tunnel with square profile consists of $(n - 1)$ sections. Floor in each section is flat and can go up or down along the section. Coordinates $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$, where $x_1 < x_2 < \dots < x_n$ describe points where tunnel's floor starts, ends, or one section joins other. Ceiling of the tunnel is 1 meter above the floor and corresponding sections join points are at coordinates $(x_i, y_i + 1)$.

Laser beam is directed to the tunnel from its start. To facilitate beam transition through the tunnel light translators could be installed at section boundaries. These translators can retransmit laser emission in necessary direction, moreover retransmitted beam not necessarily starts at the point where coming beam hit the translator.



As soon as light translators could be installed at sections boundaries only, their coordinates can only be x_1, x_2, \dots, x_n .

Your task is to determine the minimal number of light translators necessary to allow beam transition through the tunnel.

Input

The first line at input contains integer N ($2 \leq N \leq 1000$). Following N lines contain 2 numbers each — coordinates x_i and y_i (for both x_i and y_i can be used 'extended' type; $-10000 \leq x_i, y_i \leq 10000$).

Output

The only line at output should contain one integer — the minimal number of light translators necessary.

Examples

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

Problem K. Key

Input file: **standard input**
Output file: **standard output**
Time limit: 2 seconds
Memory limit: 64 Mebibytes

Hackers often have to crack passwords for different data encryption systems. A novice hacker Bill faced such a problem one day. After performing several experiments he noticed regularity in the formation of an encryption key. He knew that a key is an odd integer K , such that K^2 does not divide $(K-1)!$ and its value is between A and B , inclusively ($A \leq K \leq B$). Note, that $(K-1)! = (K-1) \cdot (K-2) \cdot \dots \cdot 2 \cdot 1$.

He was not able to advance further due to his poor mathematics. To help young hacker you have to find all possible values of the key.

Input

Input consists of two integers A and B ($3 \leq A < B \leq 10^{18}$, $(B-A) \leq 100$).

Output

Your program has to print all possible values of key K in ascending order, separated by space. It is guaranteed that there is at least one key in the range.

Examples

standard input	standard output
3 10	3 5 7 9
7 14	7 9 11 13

Problem L. Game on Chessboard (Div 1 Only!)

Input file: standard input
Output file: standard output
Time limit: 2 seconds
Memory limit: 64 Mebibytes

There is a chessboard of the size $M \times N$. K fairy chess pieces called (p, q) -leapers ($p < q$) are placed in some squares on this board. Leaper's move is similar to a regular chess knight's move, with some constraints though.

When (p, q) -leaper moves, it can move p squares horizontally and q squares vertically (only upward), or q squares horizontally (only leftward) and p squares vertically. In other words, the move to q squares must be in a direction where corresponding coordinate decreases. Moving outside of the board is prohibited. However several leapers are allowed to occupy the same square.

Two players are playing the game, alternating moves. Each player in his turn chooses some leaper and moves it according to the rules. The player, who is not able to move any leaper, loses the game. Given a board configuration, determine the winner, assuming both players play optimally.

Input

The first line of input contains 5 integers: M, N, K, p, q ($1 \leq M, N \leq 10^9, 1 \leq K \leq 10^5, 1 \leq p < q \leq 20$). Each of following K lines contains coordinates r_i and c_i of corresponding leaper ($1 \leq r_i \leq M, 1 \leq c_i \leq N$).

Output

The single line of output should contain string 'First', if the first player wins the game under optimal strategy, and 'Second' otherwise.

Examples

standard input	standard output
10 10 2 1 2 3 7 7 3	Second
7 5 3 1 3 2 3 1 5 4 3	First