1961. Cantonese Dialect

Time limit: 1.0 second
Memory limit: 64 MB

After some thought on where to spend a holiday Vova decided to travel around South China and visit Guangzhou, Shenzhen, Macau and Hong Kong.

Vova heard that South China speaks the Cantonese dialect of the Chinese language. So before setting off he learned several simple phrases in Cantonese. During his first walk around the Guangzhou center Vova said hello to n passers-by and m of them responded. Vova concluded that the other $(n - m)$ passers-by, obviously, spoke the Mandarin dialect of the Chinese language.

After Vova finished his walk, he decided to evaluate M, the number of Guangzhou citizens who speak Cantonese. Wikipedia states that the city's population is N people. Help Vova to find such M, which maximizes the probability that exactly m out of n random passers-by speak Cantonese.

Input

The single input line contains integers n, m and N ($1 \le n \le N \le 10^8$; $0 \le m \le n$). Each of the n passers-by was a Guangzhou citizen and met Vova exactly once during the walk.

Output

Print the required M. If there are multiple values of M maximizing the probability, print the largest of them.

Sample

input	output
`10 1 200`	`20`

Problem Author: Alexander Ipatov
Problem Source: Open Ural FU Personal Contest 2013

1578. Mammoth Hunt

Time limit: 1.0 second
Memory limit: 64 MB

The very last mammoth runs away from a group of primeval hunters. The hunters are fierce, hungry and are armed with bludgeons and stone axes. In order to escape from his pursuers, the mammoth tries to foul the trail. Its path is a polyline (not necessarily simple). Besides, all the pairs of adjacent segments of the polyline form acute angles (an angle of 0 degrees is also considered acute).

After the mammoth vanished, it turned out that it had made exactly N turns while running away. The points where the mammoth turned, as well as the points where the pursuit started and where the pursuit ended, are known. You are to determine one if the possible paths of the mammoth.

Input

The first line contains a positive integer N not exceeding 2000. The next $N + 2$ lines contain the coordinates of polyline vertices (this polyline is the mammoth's path). All the coordinates are integers between −2000 and 2000. No two vertices of the polyline coincide.

Output

Output NO if no polyline with vertices in the given points can be the mammoth's path. Otherwise, output YES in the first line, and the mammoth's path in the second line. The path is to be output as a sequence of point numbers (the points are numbered from 1 to $N + 2$ in the same order they were given in the input). If there are many solutions, output any of them.

Samples

input	output
2 0 0 1 1 1 2 2 0	YES 1 3 4 2
2 0 0 0 1 0 2 0 3	YES 1 3 2 4

Problem Author: Alexander Ipatov
Problem Source: USU Junior Contest, October 2007

1892. Morning in Koltsovo

Time limit: 1.0 second
Memory limit: 64 MB

Air traffic controllers at Koltsovo airport will long remember the Sunday morning of October 30. At dawn, the planes carrying participants of the Eastern Subregional Contest back home will start taxiing onto the runway one after another. At the same time, newest airplanes of the Oceanic Airlines company will be landing one after another to take part in the exhibition of the company fleet, which will be held in Koltsovo. The airport has only one operating runway, so managing this flow of airplanes might be a difficult problem for the air traffic controllers.

When all the passengers are aboard a plane, it taxis away from the terminal and queues for takeoff. When a plane is arriving to the airport, it queues for landing. A plane starts takeoff or landing as soon as it receives the corresponding clearance from the air traffic controllers. A plane is not cleared for takeoff as long as there are planes queued for landing regardless of whether they are cleared for landing or not. If a plane queues for landing at the same moment when another plane is ready for takeoff, then the latter plane will not be cleared for takeoff until the former plane lands.

For safety reasons, a certain time must pass between takeoffs and landings. After a plane starts takeoff, at least t_1 seconds must pass before another plane is cleared for takeoff and at least t_2 seconds must pass before another plane is cleared for landing. After a plane starts landing, at least t_3 seconds must pass before another plane is cleared for takeoff and at least t_4 seconds must pass before another plane is cleared for landing.

For each plane you know the exact time when this plane queues for takeoff or landing. Use these data to calculate the times when the planes will be cleared for takeoff or landing, respectively.

Input

The first line contains the integers t_1, t_2, t_3, and t_4 ($30 \le t_i \le 500$; $\max(t_1, t_4) \le \min(t_2, t_3)$). In the second line you are given the number of departing planes n ($1 \le n \le 50$). The following n lines contain the times when each plane queues for takeoff. In the next line you are given the number of arriving planes m ($1 \le m \le 50$). Then there are the times when each of them queues for landing. All the times are in the format `hh:mm:ss` and are in the range from 6 a.m. to 10 a.m. The times in each list are pairwise different and are given in ascending order.

Output

In the first n lines output the times when the departing planes will be cleared for takeoff. In the following m lines output the times when the arriving planes will be cleared for landing. The times should be output in the format `hh:mm:ss`. The planes should be described in the same order in which they are given in the input.

Sample

input	output
60 60 60 60 2 07:01:00 07:01:30 2 07:00:00 07:02:10	07:01:00 07:02:00 07:00:00 07:03:00

1547. Password Search

Time limit: 1.0 second
Memory limit: 64 MB

After his trip to Japan, Vova has forgotten his password to Timus Online Judge. Fortunately, students of the Ural State University have access to a powerful multiprocessor computer MVS-1000, and Vova can be allowed to use *M* processors for solving complex mathematical problems. Vova wants to use the supercomputer for a simple search of passwords. He remembers that his password is no longer than *N* symbols and consists of lowercase Latin letters. First he wants to check all words of length 1 in the lexicographic order (that is, **a**, **b**, …, **z**), then all words of length 2 in the same order (that is, **aa**, **ab**, …, **zz**), and so on.

In order to use the supercomputer with maximal efficiency, the search must be distributed equally between all processors: the first portion of words is checked by the first processor, the second portion is checked by the second processor, and so on. If it is impossible to distribute the work equally, let the first several processors check one word more than the remaining processors. Vova wants to know the range of words for each processor.

Input

The only line of the input contains the integers *N* and *M* ($1 \le N, M \le 50$). It is guaranteed that the number of words to be checked is no less than the number of processors.

Output

Output *M* lines. Each line must contain the range of words that will be checked by the corresponding processor. See the required format in the sample.

Sample

input	output
5 4	a-fssst fsssu-mmmmn mmmmo-tgggg tgggh-zzzzz

Problem Author: Vladimir Yakovlev
Problem Source: The 11th Urals Collegiate Programing Championship, Ekaterinburg, April 21, 2007

1512. Zinium

Time limit: 1.0 second
Memory limit: 64 MB

Background

Zinium is a miracle of our world - and probably of all the worlds. Found at the close of the XIX-th century during archeological dig on the territory of modern Albania, Zinium belonged to Caesar himself and was lost in his last campaign.

Problem

The artifact is a chessboard of **N*N** cells in size. A cell in its left-bottom corner has coordinates (1, 1) and a cell in its right-top corner has coordinates (**N**, **N**). The legend says if one could place **N** queens onto the board in such a way, that none of them attacks the other, released energy of Zinium will change the world beyond recognition. Rivers will turn back, the sky will fall onto the earth, people will learn to call things by their proper names... Petr the Gorgeous himself will probably participate in the Fourth Challenge of Timus Top Coders... Or will not. So, it is high time to check it.

Input

The only line contains the integer number **N** (4 ≤ **N** ≤ 100000).

Output

For each queen you should output the coordinates of its cell on a separate line. The coordinates should be separated by single spaces. The queens may be listed in any order. If the problem has several solutions, you should output any of them.

Sample

input	output
8	1 5 2 1 3 8 4 4 5 2 6 7 7 3 8 6

Notes

Two queens attack themselves, if they are placed in the cells positioned on the same horizontal, vertical or diagonal line.

Problem Author: Ilya Grebnov, Nikita Rybak, Dmitry Kovalioff
Problem Source: Timus Top Coders: Third Challenge

1558. Periodical Numbers

Time limit: 1.0 second
Memory limit: 64 MB

Little Tom likes amusing mathematical tasks a lot. After studying ordinary periodical numbers he wondered, what if period will be before decimal point. Generally speaking, such a "number" will have infinite number of digits before decimal point, and it will not be even a number, but it is possible to apply some operations to them. But after trying to sum up this numbers for a while, he found this task a bit complicated, even when numbers have periods of the same length and don't have unperiodical part. So he decided to write a program that will solve this problem. But he is not very good at programming, so asked you to help him and write it.

Periodical numbers can be written in form $(a_1a_2\ldots a_k)b_1b_2\ldots b_m$ = $\ldots a_1a_2\ldots a_k\ a_1a_2\ldots a_k\ a_1a_2\ldots a_k\ b_1b_2\ldots b_m$, where a_i and b_j are digits. The summation process starts from the less significant digit and going on like in addition of normal numbers, but never finishes. Your task is to sum up two periodical infinite numbers.

Input

Input has two lines containing two infinite periodical numbers each. It is guaranteed that the given numbers will not have unperiodical part (i.e. will be given in form "$(a_1a_2\ \ldots\ a_k)$") and the given periods of numbers will be the same length not greater than nine.

Output

The output must have one line containing desired number itself. It must be printed in the representation with minimal period length. Among such representations the one having the least unperiodical part's length must be chosen.

Sample

input	output
(234) (342)	(576)

Problem Source: Novosibirsk SU Contest. Petrozavodsk training camp, September 2007

2123. Knapsack

Time limit: 4.0 second

Memory limit: 256 MB

There are n types of weights. The mass of one weight of type i+1 is not less than the mass of two weights of type i. You have exactly 2 weights of each type.

Count the number of ways to select some weights with total mass equal to W. Two ways are different if for some i, the number of selected weights of type i is different.

Input

In the first line of input, there are two integers n and W: the number of types and the desired total mass ($1 \le n \le 60$, $0 \le W \le 4 \cdot 10^{18}$).

In the second line of input, there are n integers a_i: the masses of the weights. It is guaranteed that $1 \le a_1$, $2 \cdot a_i \le a_{i+1}$, and $a_n \le 10^{18}$.

Output

Print a single line containing the answer to the problem.

Sample

input	output
5 100 2 5 10 21 49	3

Problem Author: Alexey Danilyuk

Problem Source: Petrozavodsk Summer 2018. t.me/umnik_team Contest

2080. Wallet

Time limit: 2.0 second
Memory limit: 256 MB

Eugene is not a student struggling to pay his bills anymore. Now he lives in a big city and works for a well-known IT-company. He forgot the times when his wallet contained only his father's credit card with no money on it; now it's full with Eugene's own various credit, discount and membership cards.

Every week Eugene follows the same routine: he visits the same places in the same order and talks with the same people. There is no place for surprises and complications in his life. Finally, after several months in a strange city, he can just perform a pre-determined order of actions and not trouble himself with anything.

However, Eugene does have one problem: sometimes in a store, restaurant or even at a subway station entrance he has to search his wallet for the right card for more than 5 seconds. Eugene hates to waste his precious time on that.

Cards in Eugene's wallet are stored as a stack. Ha wants to be able to just take the top one every time he needs to pay or get a discount for something. After using a card he has no problem with inserting it in any position of the stack.

Knowing an order in which cards are going to be used throughout the week one can easily find a way to assemble them so that the top one will always be the right card to use. Eugene could have done that himself, but he doesn't feel like solving this problem. You do it.

Input

The first line of the input contains two integers n and k ($1 \le n, k \le 10^5$) — total number of cards Eugene has in his wallet and the number of times he uses them during the week. The cards are enumerated with integer numbers from 1 to n.

The second line contains k numbers $a_1, a_2, \ldots, a_k$ ($1 \le a_i \le n$), separated by spaces: cards' numbers in the order they are to be used.

Output

Output k+1 lines.

The first line should contain n card numbers separated by spaces from 1 to n — the initial order of cards in the stack, from top to bottom.

The line number (i + 1) should contain one number — how many cards will be located higher in the stack than the card a_i after it has been used and returned into the wallet.

If there are several correct solutions, output any of them.

Sample

input	output
3 5 3 1 2 2 1	3 1 2 2 1 0

1767. The House of Doctor Dee

Time limit: 0.5 second
Memory limit: 64 MB

Mr. X is very fond of the novel “The House of Doctor Dee”, which is set in London in the 16th century and in the end of the 20th century. Rereading the novel once again, Mr. X decided to draw a map of London with routes of the protagonists Dr. Dee and Matthew Palmer. Dr.Dee, who lived in the 16th century, often traveled from his house to St. Paul's Cathedral, and Matthew, who lived in the 20th century, regularly walked from the National Gallery to the British Museum.

Every time it turned out that the protagonists were at the same point in London, Dr. Dee had visions about what was happening to Matthew at that moment. Such moments were often enough, and Mr. X decided to draw the motion trajectories of Dr. Dee and Matthew in such a way that their common part be as long as possible. However, Mr. X is not very good at the topography of London. He assumes that the city is built by a rectangular scheme—each street stretches through the whole city either from west to east or from north to south. The house of Dr. Dee, St. Paul's Cathedral, the National Gallery, and the British Museum are located exactly at the intersections of two orthogonal streets. Moreover, Mr. X is sure that Dr. Dee and Matthew always took one of the shortest possible routes.

Though London is very big, Mr. X has already drawn its map. He has also marked Dr. Dee's house, St. Paul's Cathedral, the National Gallery, and the British Museum on the map. It now remains to draw the required routes.

Input

The first line contains coordinates of Dr. Dee's house separated with a space. The following lines contain the coordinates of St. Paul's Cathedral, the National Gallery, and the British Museum in the same format. All the coordinates are integers with absolute values not exceeding 10^9.

Output

Output the maximum length of the path that Dr. Dee and Matthew Palmer's routes can have in common.

Sample

input	output
2 2 4 4 5 5 3 3	2

Problem Author: Alex Samsonov
Problem Source: The 14th Urals Collegiate Programing Championship, April 10, 2010

1398. Bishop and Pawn

Time limit: 1.0 second
Memory limit: 64 MB

There are a white bishop and black pawn on a chessboard. Moves are made in accordance with the usual chess rules. White moves first. Black wins if he can promote his pawn to a queen and the white bishop cannot capture the queen by the subsequent move. The game ends in a draw if it’s Black’s turn to move but the pawn cannot move forward. In other cases, White wins. It is required to tell the result of the game if both sides play optimally.

Input

The first and second lines show the positions of the white bishop and black pawn, respectively, by means of the standard chess notation. The rank in which the pawn is initially positioned may have the number from 2 to 7, and the bishop is initially positioned at any square different from the pawn’s square.

Output

Output WHITE if White wins, DRAW in the case of a draw, and BLACK if Black wins.

Sample

input	output
`a1` `c2`	`WHITE`

Problem Author: Magaz Asanov
Problem Source: Practice tour of Urals Championship 2005