Problem A. Code number

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

Problem L in the problem set for the $ICL'13$ tournament turned out to be extraordinarily difficult. The long contest hours did not bring any of the teams to a correct solution, but one of the participants came up with a sneaky plan. He found out that the solution to the task is hidden in a safe, which can be opened using a certain sequence of numbers. The following is known about the sequence:

1. All elements in the sequence are divisors of a certain number N.
2. None of these elements can be divided by one of the others.
3. The sequence has the maximum possible length.
4. Of all the sequences that satisfy the previous three requirements, the correct one is the smallest lexicographically.

Open the safe.

Input

A single integer N ($1 \le N \le 2^{63} - 1$). It is guaranteed that the number N does not have prime divisors greater than 100.

Output

In the first line, output a single number — the quantity of numbers in the sequence. The second line should contain the numbers themselves, separated by spaces.

Example

standard input	standard output
12	2 2 3

Problem B. Puck game

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

For the Puck game rectangular field $N \times M$ is used. Initially the puck of radius R is placed at left side of the field, so, its center is located at the point $(R, N/2)$. Goal of the game is to pass the puck to final position, where its center will be located at point $(M - R, N/2)$, and minimal length of curve, drawed by center of the puck while its movement, is called *the result of game*.

Recently, Federation of Puck game added k walls, parallel to left side of the field. Each wall may have several doorways, width of each doorway is $2 \cdot R$. It is guaranteed that final position still can be reached from starting point.

Given coordinates of all walls and doorways, calculate the result of game.

Input

First line of the input file contains two integers N and M ($10 \le N, M \le 5000$) — dimensions of playing field. Second line contains two integers R and K, where R ($1 \le R \le 10$) is the radius of the puck and K ($0 \le K \le 100$) is number of walls. Next K lines contain descriptions of walls in next format: first integer T ($1 \le T \le 40$) is number of doorways in current wall, second integer W is x-coordinate of the wall ($R \le W \le M - R$), and next T integers are coordinates of upper point of doorways, sorted in descending order. Walls are listed from left to right, distance between any two walls is not less than $2 \cdot R$.

Output

Print one real number with accuracy up to 10^{-6} — length of the shortest way, passed by the puck.

Example

standard input	standard output

`10 10`
`1 1`
`1 3 6`
	`8.000000`

Problem C. Trainings

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	256 mebibytes

At the University of Illusions, Teleport Magic and Occlumence there are n students, patricipating in ACM ICPC ($n = 3^k$). The coach of the university's teams wants to run a series of team training events with the following rules:

- a team always has three participants
- any two students should play together on the same team exactly one time.

Before making a training schedule, coach must create a list of teams that meets the requirements above. Since the coach is busy right now preparing the contest for future students, you have been asked to make the list of teams.

Input

A single number n ($n < 1000, n = 3^k, k > 0$) — the number of students, patricipating in ACM ICPC.

Output

In the first line, print p — the total number of teams participating in the training series. In the following p lines, output three numbers in each — the numbers of the students participating in this team. Students are numbered from 1 to n.

Examples

standard input	standard output
3	1 1 2 3

Problem D. Ideal sets

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

Of all the arts, the most essential is the art of perfect balance.
*„History of Novogireevka“, volume*1.

In the markets of the city of Novogireevka, all the sellers weigh their goods using a set of weights, each of which weighs an integer number of grams, and the total mass of all the weights combined equals N grams. This is called a *perfect* set if any goods with a mass equal to a integer number of grams from 1 to N can be balanced by a certain quantity of weights from this set, and in addition, if there is only one way to do this. The goods are always put in the left cup on the scales, and the weights are put on the right side. Two methods of weighing that differ only by replacing some of the weights with other weights of the same mass are considered to be identical. For example, for $N = 5$ there are three sets like this: $(1, 1, 1, 1, 1)$, $(1, 1, 3)$ and $(1, 2, 2)$.

Find all the perfect sets that have the smallest number of weights with a total mass of N grams.

Input

The first line specifies the quantity of goods to be weighed T $(1 \le T \le 100)$. The second line is a comma-separated list of T positive integers $N_1, N_2, \ldots, N_T$, where N_i is the total mass of all the weights in the i-th set $(1 \le N_i \le 10^5)$.

Output

For each i-th good, output two numbers in the first line: M — the quantity of perfect sets that have the smallest number of weights, and K — the quantity of weights in each of these sets; then, in the next M lines — the perfect sets of K weights with a total mass of N_i grams, listed in order of non-descending mass. Output all the sets in alphabetical order.

Examples

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

Problem E. Mobius band

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

And they sped along the road, driven by the wind, Mobius bands and intricate Lissajous curves. From the collection «Joking Physicists».

Who doesn't know what a Mobius band is? You don't know? It's very simple. Take a rectangular-shaped strip by both ends. Turn the right end of the strip by 180 degrees and glue the ends together. Now you have a Mobius band. Like this one.

The distinctive feature of the Mobius band is the fact that it is a single-sided surface, in contrast to a simple rectangle.

Or, using a scientific definition, the Mobius band is a topological object on which it is possible to get to any point on the surface from any other point on the surface without crossing an edge. This strip was invented by the German mathematician and astronomist August Ferdinand Mobius, a professor at the Leipzig city university (Germany), while observing a housemaid who accidentally put her scarf on her neck on the wrong way. Incidentally, certain scientists believe that our Universe is also built on the principal of the Mobius band. The Mobius band is divided into squares. Some of the squares are impassible, and they are marked with the symbol '`*`'. Squares that have a shared side are considered neighboring squares. From a certain square you can get to any neighboring square if that square is not impassible. You must find the minimal length of the path from the starting square D to the finish square K. The distance between two neighboring squares is considered to be equal to one.

Input

The first line contains two integers N and M ($1 < N \leq 1000, 1 < M \leq 1000$) — the dimensions of the original rectangle $PQRS$.

Each of the next $2 \cdot N$ lines contains M symbols. Allowed symbols are '`.`' (a dot, indicating a normal square), '`*`' (an asterisk, indicating an impassible square), '`D`' (indicates the starting square), and '`K`' (indicates the finish square). The last two symbols only occur once each. The first N lines set one of the sides of the strip; point P corresponds to the lower-left corner, point Q is the lower-right corner, point R is the upper-right corner, and point S is the upper-left corner. The following N lines set the opposite side of the strip, and the strip is oriented so that point P corresponds to the lower-right corner and point S is the upper-right corner.

When constructing the band, point P is glued to point R, and point Q is glued to point S.

It is guaranteed that way between start and finish squares exists.

Output

Output the minimal length of the path between the start and finish squares.

Examples

standard input	standard output
2 5 D.... ***** ***** ..K..	3

Problem F. Cost of the question

Input file:	`standard input`
Output file:	`standard output`
Time limit:	7 seconds
Memory limit:	256 mebibytes

Given an undirected connected graph without loops or multiple edges, made up of n vertices and m edges. Each of the m edges has a weight from 0 to 10^9. Initially, each vertex defines a component that consists of a single vertex — itself.

Operations are allowed that connect two components with an edge, but each such operation has a cost, which is determined in the following way. Assume that a set of vertices A and B are already connected into components. Then the cost of connecting A and B is equal to the maximum weight of all the edges that exist between A and B. In other words, the cost is equal to the maximum of the weights of all the edges u, v such that $u \in A$, $v \in B$. Any two distinct edges have distinct costs.

Find the minimal cost of connecting all the vertices in a single component.

Input

The first line specifies integers n and m $(1 \leq n \leq 5000, 0 \leq m \leq 300000)$. The following m lines describe edges using sets of three integers u v c, where c is the cost of the edge and u and v are the vertices being connected.

Output

In the first line, output two numbers: the cost of the connection and the quantity of edges. In the following lines, output the edges themselves.

Examples

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

Problem G. Roses

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

Sam loves roses and grows them in his garden. Sam also loves money, so he sells his roses at the nearest market. Before setting off for the market, Sam counted the petals on each rose and decided that he will sell roses with K petals for R rubles. If a rose has less than K petals, each missing petal reduces the price by 1 ruble. If there are more than K petals, Sam will sell this rose for twice the price — for $2R$ rubles. By the end of the day, Sam has sold all his roses and wants to count up his proceeds.

Input

The first line contains three numbers K ($1 \le K \le 100$), R ($K \le R \le 1000$) and N ($1 \le N \le 100$). The second line contains N numbers, not exceeding 100 — the number of petals on each of the N roses.

Output

A single number — Sam's proceeds.

Examples

standard input	standard output
`10 20 3` `10 5 15`	`75`

Problem H. Hare's run through the fields

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

Once, my uncle in Michigan tamed a wild hare. A real hare, with long ears. Then it ran away, and hunters started firing at it. They came running from all over the place. For three days they chased him through the fields. But every time he found a new way to dodge them and ran faster and faster. . .
(S. Japrisot, "Hare's run through the fields".)

. . . So they chased the hare up to my uncle's ranch, which was surrounded by a fence in the shape of a convex polygon $A_1A_2 \ldots A_n$. The hare dashed out from the inside point A of the section and ran in the direction of the border A_1A_2, hoping to hide in the bushes. When he reached the fence A_1A_2, he ran along it for a distance of d_1 and turned in the direction of the fence A_2A_3, where there was another thicket, then ran along it for a distance of d_2, and so on. At the end of his route, the hare ran across to the fence A_nA_1 and ran a distance of d_n along A_nA_1. After this, he went back to his initial point A, where my uncle happily received him in his arms, thus completely befuddling the hunters who were chasing him.

You must calculate the length of the hare's shortest route that starts and finishes at point A.

Input

The first line contains a single positive integer N — the number of vertices on the fence ($3 \le N \le 1000$). In each i-th of the next N lines there are two numbers — the coordinates of the fence's vertices, in order of counter-clockwise traversal. The next line contains N integers — distances $d_1, d_2, \ldots, d_N$, although $0 \le d_i < l_i$ (l_i — the length of the side A_iA_{i+1}, l_N — the length of the side A_NA_1). The next line has two more numbers — the coordinates of point A. Coordinates of all the points — integers not surpassing the absolute value 10 000. All numbers are separated by spaces. It is guaranteed that the shortest route does not go through a vertex of the polygon.

Output

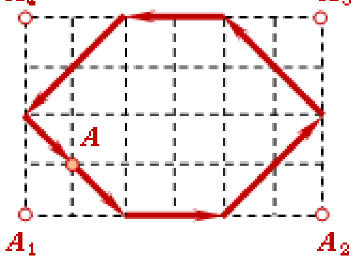
Output the length of the hare's shortest route with accuracy of at least 10^{-5}.

Examples

standard input	standard output
4 0 0 6 0 6 4 0 4 2 0 2 0 1 1	15.313708

Note

The picture shows the hare's shortest route, containing two segments of the length 2, located on the sides A_1A_2 and A_3A_4 of the area $A_1A_2A_3A_4$. Its length is equal to $8\sqrt{2} + 4 = 15.31371\ldots$.


A
A_1
A_2

Problem I. Thimblerigger

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

Beating the thimblerigger is not an easy task! He has n thimbles lined up in a row. The game proceeds like this: first the thimblerigger places a coin under the k-th thimble, then starts to switch certain pairs of thimbles. We know that if thimbles a and b were switched, and the coin was under thimble a, as a result of the switch the coin will now be under thimble b. Luckily, we also know the thimblerigger's strategy. Switches are made using the following algorithm. Consider a certain set of thimbles S:

1. If this set contains two thimbles, they exchange places.
2. Otherwise, consider all the sets of thimbles in alphabetical order that consist of $|S - 1|$ thimbles, where $|S|$ is the number of thimbles in set S, and return to step 1. For example, if S was $\{1, 2, 3, 4\}$, the following sets will be considered in order as the result: $\{1, 2, 3\}$, $\{1, 2, 4\}$, $\{1, 3, 4\}$, $\{2, 3, 4\}$.

You must predict where the coin will end up after the thimblerigger's tricks.

Input

The first line contains the integers n and k ($2 \le n \le 10^{18}, 1 \le k \le n$).

Output

A single number — the number of the thimble that the coin will be found under as a result of the thimblerigger's actions.

Examples

standard input	standard output
4 2	4
3 1	3

Problem J. Twisted palindrome

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

We are given two strings: A and B. The length of both of these strings is identical and equal to n. Lets call *twisted* a string with characters that are made by concatenating the substring A from the symbol i to the symbol j and the substring B from the symbol j to the symbol k, $1 \le i \le j \le k$.

A twisted palindrome is either a twisted string that is a palindrome, or any palindrome substring from string A or B. You must find a twisted palindrome of the maximum length and print it.

Input

The first line contains the number n ($1 \le n \le 10^5$). The following two lines contain the words A and B, which consist exclusively of capital Latin letters.

Output

In the first line, print a single number — the length of the maximum twisted palindrome. The following two lines must contain the words A and B, in which any symbols that are not included in the maximum twisted palindrome are replaced with the dot symbol '`.`'.

Examples

standard input	standard output
5 ABCDE BAECB	5 .BC.. ..ECB

Problem K. Phenomenal Memory (Division 1 Only!)

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

Mister John Smith has a phenomenal memory. He reads a lot and remembers the texts he has read quite well. Smith recently read a new book consisting of M words. Suddenly, John realized that the text is all too familiar. A book he had read last year, which consisted of N words, surfaced in his mind. Now it only remained to find out how similar these two books actually are.

Input

The first line has an integer N ($1 \le N \le 10^5$). Next there are N strings S_i, consisting of lowercase Latin letters. The total length of the strings S_i does not exceed 10^5. The next line contains one integer M ($1 \le M \le 10^6$). Then M strings T_i follow, consisting of lowercase Latin letters. The total length of the strings T_i does not exceed 10^6.

Output

For each symbol of each string T_i, output the length of the maximum prefix of one of the strings S_j, which matches a substring of the string T_i that starts with this symbol.

Examples

standard input	standard output

`5`
`acbd`
`caba`
`abar`
`book`
`bat`
`2`
`abacaba`
`abarbabooca`
	`3 2 2 4 3 2 1`
	`4 2 1 0 2 2 3 0 0 2 1`

`2`
`icl`
`xiii`
`2`
`xiiiicl`
`icicx`
	`4 1 1 1 3 0 0`
	`2 0 2 0 1`

Problem L. Twins (Division 2 Only!)

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

Lana and Nala enjoy being identical twins. They wear the same outfits every day, including jewelry. They especially love necklaces of colored beads, but they continually run into a problem trying to determine whether two beaded necklaces are identical or not.

Beads come in 26 colors, and the colors are identified by the lower case alphabet. A beaded necklace containing n beads is represented by a string of n lower case alphabetic letters. The necklaces are continuous (no clasps). It is easy to see that the two beaded necklaces "`abcdefg`" and "`abcdefg`" are identical. But it is not quite so obvious that the necklaces "`efgabcd`" and "`bagfedc`" are also identical to "`abcdefg`".

The twins need you to write a program for them that will determine whether or not two beaded necklaces are identical.

Input

The input will begin with an integer C ($1 \leq C \leq 100$) that denotes the number of test cases. The rest of the input will consist of C pairs of necklaces, each on a line by itself. Each necklace contains at least 1 and no more than 99 beads.

Output

The output should display the case number (as shown below) followed by either "`YES`" or "`NO`", indicating whether the necklace pair is identical or not.

Example

standard input	standard output
4 lana nala abcdefg abcdefg abcdefg bagfedc abcdefg opqrst	Case #1: YES Case #2: YES Case #3: YES Case #4: NO