Problem A. Article

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

In the year 2048 sociologists got interested in the structure of the community of competitive programmers and as the result of their studies made the following observations. The news about the success of any contestant passes from any of his acquaintances to all his acquaintances (probably through other people). Two persons do not communicate unless they know each other and when they do, they only discuss people they both know, but not themselves. Additionally, looking at the acquaintances of any person, there are no triples among them where all three would be pairwise unacquainted.

Now the community decided to pool their funds to pay the enormous license fee for an article describing a new algorithm that can multiply square matrices in $\Theta(N^{2.32768})$ time (where N is the size of the matrices). Unsurprisingly, the license agreement strictly forbids making copies of the article. Everyone wants to read the article, of course, but no one is willing to pass it on to someone they don't know. Also, they are busy people and don't want to manage the article after reading it. The only exception is Vasya: he's the holder of the article; he bought the copy with the pooled money and he will keep it after everyone has read it. But even he is only willing to collect it from the last reader for archiving, but not to run around carrying it from one reader to another.

Find the maximal number of people who can read the article under the above conditions and the order in which they should pass the article among themselves for that to happen.

All programmers are numbered by positive integer numbers and Vasya's number is 1.

Input

The first line contains two integers N and M ($3 \le N \le 1000$, $0 \le M \le 10\,000$, $M \le \frac{N(N-1)}{2}$), the number of programmers and number of pairs of acquainted programmers. Each of the next M lines contains exactly two integers p_1 and p_2 ($1 \le p_1, p_2 \le N$, $p_1 \ne p2$), numbers of acquainted programmers.

Output

In the first line output integer K, the number of programmers who can read the article. In the next line output K different numbers, the order in which they can read it. This sequence should start with 1, should not contain any number more than once, any two consecutive numbers should correspond to acquainted people and the last programmer should be acquainted with Vasya.

Examples

article.in	article.out

3 3
1 2
1 3
2 3
	3
	1 3 2

4 5
1 2
1 3
1 4
2 3
2 4
	4
	1 3 2 4

Problem B. Airlines – 2

Input file:	`avia2.in`
Output file:	`avia2.out`
Time limit:	1 second
Memory limit:	256 megabytes

Oh! Bytelandavia, once the sole airline in Byteland, has gone bankrupt and now several new companies are after their contracts. Byteland's Civil Aviation Office (CAO) wants to minimize their number, but without hurting the established timetables.

There are N airports in Byteland, and so far there have been direct flights in both directions between all pairs of them. The officials of the CAO want to preserve this situation, but for each pair of airports they want all flights between them be operated by a single company.

The second requirement of the CAO is harder to understand, but it is mandatory none the less (they are not pushing their papers for nothing)...

So, the requirement is that a traveler is not allowed to make a round trip (returning to his starting airport) using only planes of a single company if he makes intermediate stops in an even number of airports. For example, if he has to fly from New Vasiuky to Old Moscow, then to Vovinburg and then return to New Vasiuky, he will have to use at least two different companies. However, if he would also pass through New Bobruisk, there would be no such requirement.

Determine the minimal number of airline companies needed to fulfill the requirements of the CAO. Find also one possible division of the flights between the new companies.

Input

The only line of input contains N, the number of airports in Byteland ($2 \le N \le 500$).

Output

The first line of output should contain K, the number of airline companies. $N - 1$ lines should follow. The i^{th} of these lines ($1 \le i \le N-1$) should contain $N - i$ integers. The j^{th} integer of the i^{th} line should denote the airline company that operates flights between the airports i and $i+j$. Both the companies and the airports are numbered starting from 1.

Examples

avia2.in	avia2.out
2	1 1

3	2
	1 2
	1

Note

The first task in this series was used in the Minsk quarter-final of the ACM ICPC in year 2007.

Problem C. Blots on Paper

Input file:	`blot.in`
Output file:	`blot.out`
Time limit:	1 second
Memory limit:	256 megabytes

Some ink was spilled on a grid paper measuring $M \times N$ cells. Each cell of the paper is now considered either pained or clean. Two painted cells belong to the same blot if there is a path from one of the cells to the other that goes through painted cells only and on each step moves from one cell to another only horizontally or vertically.

Count the number of blots and the area of the largest blot (that is, the number of cells it consists of).

Input

The first input line contains the integers M and N ($1 \le M, N \le 10^5, 1 < M \cdot N \le 10^6$). Each of the following M lines consists of N characters 0 or 1, where 0 denotes a clean and 1 a painted cell. There is at least one painted cell.

Output

The only output line should contain two integers: the number of blots and the area of the largest blot.

Examples

blot.in	blot.out

`6 7`
`1001001`
`1111011`
`1001000`
`1001111`
`0100000`
`0000000`
	`3 13`

`4 4`
`0000`
`0000`
`0010`
`0000`
	`1 1`

Problem D. Compression

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

Vasya is not satisfied with the quality of the spam filter in his mailbox and decided to build his own. He has prepared a list L_0 of N_0 words ($1 \le N_0 \le 5 \cdot 10^4$). Each of the words is at least 1 and at most 40 characters long. In his opinion, any message that contains even one of those words is spam.

The list is quite big and Vasya wants to compress it. More precisely: he wants to replace the list L_0 with a list L_1 of non-empty words such that for every word $w_0 \in L_0$ exists a word $w_1 \in L_1$ where w_1 is a prefix of w_0. Obviously, if the words in L_1 are too short, then too many messages will be falsely qualified as spam. After some thinking, Vasya came up with the following condition: he wants to find the list L_1 so as to minimize the penalty $P = \sum\limits_{w_1 \in L_1} 2^{40-|w_1|}$ where $|w|$ denotes the length of the word w.

Help him solve this problem.

Input

The first input line contains N_0, the number of words in the list L_0. The following N_0 lines contain the words from the list L_0, consisting of lowercase letters only.

Output

The first output line should contain the value of P corresponding to the optimal list L_1. The second line should contain N_1, the number of words in the list L_1. The following lines should contain the words from the list L_1. If there are several optimal solutions, any one of them is acceptable.

Examples

compression.in	compression.out
4 abxcaba abax abay abaz	146028888064 2 aba abxcaba

Problem E. Teach Yourself Pottery

Input file:	`frustum.in`
Output file:	`frustum.out`
Time limit:	1 second
Memory limit:	64 megabytes

Vitya loves educational TV channels. Having seen a program on ancient handicrafts, he decided to apply himself to pottery. As the first task, he built a potter's wheel from an old bucket and acquired a small straight stick for smoothing out the walls of the vessels. He plans to use the kitchen oven for firing his creations...

So far Vitya manages to produce only beakers—vessels having the shape of a truncated cone with the radius of the base equal to the radius of the bottom of the bucket and the length of the side equal to the length of his smoothing stick.

Vitya wants to give his first proper beaker to his mother as her birthday present. He also wants the vessel to be able to contain as much delicious foods as possible and thus he asks you to compute the maximal volume possible with his technology.

Input

The only input line contains two integers L and D ($1 \leq L, D \leq 10^3$), the length of the stick and the diameter of the bucket. Thickness of the walls of the beaker is negligible.

Output

The only output line should contain the desired volume where either absolute or relative deviation from the true value is at most 10^{-9}.

Examples

`frustum.in`	`frustum.out`
`1 1`	`1.7141155472`
`4 5`	`143.2761976593`

Problem F. The Young Networker

Input file:	`lan.in`
Output file:	`lan.out`
Time limit:	1 second
Memory limit:	256 megabytes

The early 2000's. . .

Having just boasted about your extensive experience in designing small-scale home networks, you were asked to take on the job for your apartment block. Now is probably not the time to confess that this is really your first attempt.

So, the network will connect N users. Each of them connects to one of the hubs via a dedicated network cable. The hubs may also be connected to one another with similar cables. Any socket of a hub may be used by either an end user or for a connection to another hub. The final network will have to enable any pair of users to communicate with each other via one or more hubs.

Digging through your piles of old hardware, you found M hubs suitable for the purpose. The i^{th} of those hubs ($1 \le i \le M$) has k_i sockets for network cables.

As you have promised that each end user will only have pay for the cable to connect themselves to the nearest hub, you want to design the network using minimal number of your hubs (so you can keep the rest for future projects).

Are you up to the task?

Input

The first input line contains the integers N and M ($2 \le N \le 1000, 1 \le M \le 300$). The second line contains M integers, the values of k_i ($2 \le k_i \le 48$).

Output

If there is no solution, the only line of output should contain the text `Epic fail`.

Otherwise, the first line of output should contain K, the number of hubs used, and the second line K integers, the numbers of the hubs (they are numbered starting from one in the order they are given in the input).

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

Examples

lan.in	lan.out

`10 2`
`48 10`
	`1`
	`1`

`2 5`
`2 2 2 2 2`
	`1`
	`5`

`10 1`
`9`
	`Epic fail`

`10 2`
`5 6`
	`Epic fail`

`20 2`
`11 11`
	`2`
	`2 1`

Problem G. Pattern Matching

Input file:	`pattern.in`
Output file:	`pattern.out`
Time limit:	1 second
Memory limit:	64 megabytes

Pattern is a nonempty string that may contain lowercase letters of the Latin alphabet and the special character "`*`" (star).

We will say that a string T *matches* the pattern P if and only if there is a way to replace the stars in P with (possibly empty) sequences of lowercase letters so that the result equals T. For example, the string `aadbc` matches the pattern `a*b*c`, as we can obtain the string from the pattern by replacing the first star in the pattern with `ad` and the second one with the empty string. On the other hand, the string `abcbcb` does not match this pattern.

Given a non-empty string S, count the number of cyclic shifts of S that match the given pattern P.

Input

The first line of input contains the pattern P (1 to 100 characters long). The second line contains the string S (1 to 100 000 characters long).

Output

The output should consist of a single integer, the number of cyclic shifts of S that match the pattern P.

Examples

pattern.in	pattern.out
aaaa aaaa	4
a*a aaaaaa	6
*a*b*c* abacabadabacaba	15

Problem H. Roman Cities

Input file:	`roman.in`
Output file:	`roman.out`
Time limit:	1 second
Memory limit:	256 megabytes

A well-known software development company has been commissioned by the Archaeological Society. One of the modules has to help the archaeologists to process data about the ruins of buildings they have found during their excavations of ancient cities. Development of this module has been assigned to Vasya.

Vasya, being a seasoned programmer, at once noticed that the module would need a database to contain the descriptions of the ruins and the estimated construction times of the buildings. It would be all fine, but suddenly the manager got the genial idea that since the database describes Roman ruins, the years of construction should be stored in the Roman number system. Now Vasya is wondering how many symbols he needs to set aside for each year number field in the database. According to the functional specification, the software module must be able to handle years from A to B (inclusive). Help Vasya determine the minimal number of characters sufficient for storing any year number in the range from A to B.

Input

The only input line contains the descriptions of the years A and B, separated by the "`-`" sign. A description of a year consists of one to four decimal digits (the number of the year), followed by either "`AD`" (*Anno Domini*, the current era) or "`BC`" (*Before Christ*, before the current era). In both directions the years are numbered starting from 1. It is known that $753\text{BC} \le A \le B \le 2012\text{AD}$.

Output

The output should consist of a single integer, the minimal number of characters that have to be reserved in the database for the year number.

Examples

roman.in	roman.out
`753BC-747BC`	`3`
`1BC-1AD`	`7`
`2000AD-2012AD`	`10`

Note

It is well known that the Roman number system uses the following seven letters for digits: `I` = 1, `V` = 5, `X` = 10, `L` = 50, `C` = 100, `D` = 500, and `M` = 1000. Natural numbers are written by repeating these digits. For correct representation of a larger integer in the Roman system, first the thousands, then the hundreds, then the tens and finally the ones are written. In any of the positions, some digits (`I`, `X`, `C`, `M`) may be repeated, but no more than three times.

If in any position a digit with lower or equal value is to the right of a digit of higher value, the lower value is added to the higher one. Only `I`, `X`, `C`, and `M` may be added and there may be no more than three equal-valued digits in any position. If, however, a digit with lower value is to the left of a digit with higher value, the lower value is subtracted from the higher one. There are only six subtractions allowed: `IV` = 4, `IX` = 9, `XL` = 40, `XC` = 90, `CD` = 400, and `CM` = 900. Any other combinations are not allowed. So, for example, 99 has to be written as `XCIX`, not as `IC`.

It is also important to remember that the Romans did not use the AD/BC concepts. Instead, they counted the years from the mythical founding of Rome (*Anno Urbis Conditae*—753BC).

Some examples of Roman year numbers:

- 753 BC = 1 AUC = `I`
- 1 BC = 753 AUC = `DCCLIII`
- 1 AD = 754 AUC = `DCCLIV`
- 2012 AD = 2765 AUC = `MMDCCLXV`

Problem I. Sequence Sum

Input file:	`sequence.in`
Output file:	`sequence.out`
Time limit:	1 second
Memory limit:	64 megabytes

Little Kostya has many cards with numbers 1 to N on them. Last time he laid out the cards from 1 to N in increasing order and obtained a huge number. He could not figure out what to do with this number, but his dad, having noticed several strange properties of the number, proposed a task to a contest similar to the one you're now participating in.

Two years have passed. During this time many numbers have been laid out using the cards. However, Kostya does not like to lay out monotonous sequences any more. This is because his mom told him how many students were disappointed that they could not solve the first task about the cards.

As time progresses, the students keep asking for new tasks to solve. Now the dad, having mastered all his creative thoughts, decided that this time the students will have to solve the following: let's consider all integer sequences with lengths up to N and elements in the range from 1 to N; now, let's exclude all the monotonous sequences (as you remember, there already was a task about them two years ago); next, let's view each of the sequences as a number written in base $N + 1$; and finally—what could be easier—let's just sum them all. Of course, Kostya, still a little boy, probably would not be able to do this, but perhaps you can?

P. S. The sequence a_1, a_2, ..., a_k is turned into the number $\overline{a_1 a_2 \ldots a_k}$ in base $N + 1$.

Input

The only line of input consists of the integer N, from 1 to 1 000 000 000.

Output

The only line of output should consist of the sum described above. However, the actual value may be gargantuan, so you're asked to output it modulo 1 000 000 007.

Examples

`sequence.in`	`sequence.out`
`2`	`12`
`3`	`1080`
`4`	`103460`

Note

For $N = 3$ the possible sequences are: 1, 2, 3, 11, 12, 13, 21, 22, 23, 31, 32, 33, 111, 112, 113, 121, 122, 123, 131, 132, 133, 211, 212, 213, 221, 222, 223, 231, 232, 233, 311, 312, 313, 321, 322, 323, 331, 332, 333. Of those, the following are non-monotonous: 11, 22, 33, 111, 112, 113, 121, 122, 131, 132, 133, 211, 212, 213, 221, 222, 223, 231, 232, 233, 311, 312, 313, 322, 323, 331, 332, 333.

Problem J. Order Splitter

Input file:	`splitter.in`
Output file:	`splitter.out`
Time limit:	1 second
Memory limit:	64 megabytes

Electronic stock exchange is a complicated real-time order processing system. Today electronic exchanges operate in the market handling millions of orders every day. You have set up a small company which provides execution services for clients and you're facing a problem of large order splitting between exchanges. It's very expensive to set up a new exchange, hence there is only limited number ($1 \le N \le 30$) of exchanges available to split an order.

An order has a notional (projected amount that has to be executed on the exchanges) $1 \le L \le 10^9$ and a price. In scope of this problem, you only care about the notional. The price at which an order is sent to an exchange will be provided by a client and forwarded to the exchange as is.

The given order has to be split into multiple child orders, which are sent to the exchanges according to the given ratios R_i $(0 \le i < N)$. However, each exchange has restrictions on the step size. Each order notional has to be divisible by a certain step size specified by the exchange. You are given a step size S_i of each exchange, meaning that the exchange i can only accept orders of notional S_i, $2 \cdot S_i$, $3 \cdot S_i$, ...

Because it's not always possible to precisely split the client order notional into child orders, an algorithm has to be written to achieve the best possible allocation that minimizes $D = |L - \sum C_i|$, where the notionals of the child orders allocated to corresponding exchanges are denoted as C_i. Clients prefer that the child order size is not significantly different from $CR_i = L \cdot R_i / \sum R_i$. To meet that requirement, C_i can be obtained only by rounding CR_i either up or down to the nearest step size. However, if CR_i is divisible by S_i, it's required that $C_i = CR_i$. It means that you must send the child order notional according to the given ratio if it can be sent, or otherwise round it either up or down. If rounding down yields 0 notional, then you can choose not to send an order to the exchange.

Input

The first line of each test case contains 2 integers: number of exchanges $1 \le N \le 30$ and the order notional to split $1 \le L \le 10^9$. The second line contains N integers specifying $0 \le R_i \le 100$, $\sum R_i > 0$. The third line contains N integers specifying the step sizes $1 \le S_i \le 10^9$.

Output

For each test case output one line containing the cumulative notional of the child orders $L_r = \sum C_i$, optimized according to the rules described above. If there is more than one optimal L_r, output the smaller value.

Examples

splitter.in	splitter.out

2 250
1 2
100 150
	250

3 1000
25 25 50
34 77 52
	989

3 10
1 1 0
50 50 50
	0

Problem K. The Merry Student Life During the Term...

Input file:	`student.in`
Output file:	`student.out`
Time limit:	2 seconds
Memory limit:	256 megabytes

As Vasya Pupkin has completed no labs since the start of the term, the dean has threatened to expel him. Vasya would really like to avoid that, so, finally, he decided to get serious...

During this term, Vasya is enrolled in N classes, and each of them asks for K_i labs ($1 \le i \le N$). Vasya realizes it's impossible to study all subjects at once, so he has decided to study one of them, then do all the labs in that subject, and only then turn to the next subject...

The labs in one subject can be done in any order, but the lab instructors will assign penalties for lab work that is turned in late. The penalty depends on the time when the work was actually turned in and equals $w_j \cdot t_j$ for every $1 \le j \le T$, where w_j is a coefficient given by the lab instructor, t_j is the time when the work for the lab j is turned in, and $T = \sum_{i=1}^{N} K_i$ is the total number of labs to be done.

The time needed to perform the lab (after studying the subject, of course) is known for each lab and equals p_j for every $1 \le j \le T$.

Help Vasya find the order of tackling the subjects that minimizes the total penalty

$$\sum_{j=1}^{T} w_j \cdot t_j.$$

Assume that all labs are done immediately one after another and that the times to study the subjects are negligible compared to the lab times.

Input

The first input line contains the integer N ($1 \le N \le 500$). The second line contains the N values of K_i ($1 \le K_i \le 100$). The third line contains T integers, the values of p_j ($1 \le j \le T$), where the labs for the first subject are listed first, then the labs for the second subject, and so on. Finally, the fourth line contains the values of w_j in the same order. All the values p_j and w_j are positive integers not exceeding 10 000.

Output

The first output line should contain the minimal total penalty (an integer). The second line should contain a permutation of the integers 1, ..., T that achieves the minimal penalty. If there are several possible solutions, output any one of them.

Examples

student.in	student.out

1
5
1 2 3 4 5
5 4 3 2 1
	70
	1 2 3 4 5

2
2 2
1 1 2 2
1 1 2 2
	23
	1 2 3 4

Problem L. Sum

Input file:	`sum.in`
Output file:	`sum.out`
Time limit:	1 second
Memory limit:	256 megabytes

You are given an integer N and asked to partition it into a sum of natural numbers so that the sum of the inverses of these natural numbers would be equal to one; that is, to find K natural numbers $n_1, \ldots, n_K$ such that

$$\sum_{i=1}^{K} n_i = N \qquad \text{and} \qquad \sum_{i=1}^{K} \frac{1}{n_i} = 1.$$

Input

The only input line contains the integer N ($1 \le N \le 1\,000\,000\,000$).

Output

If the required partition does not exist, the only output line should contain the text `Epic fail`, otherwise the first output line should contain the number of addends and the second line the addends themselves, in any order. If there are several solutions, output any one of them.

Examples

sum.in	sum.out
2	Epic fail
4	2 2 2