Problem A. Tests

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

It is soon the exam time for entrants at Berland State University, in short BerSU. As usual, for every subject, instead of passing an entry exam, one can present the results of the centralized test on that subject. Knowing that, Peter had passed the tests for all of the M subjects in advance.

The selection committees of each of the N BerSU faculties have published the rules for automatic conversion of test results into exam grades, as well as the sum of the grades necessary to enter study for each of the S specialties.

Peter is not very confident in his test results, so he made an application for every possible specialty of each of the BerSU faculties. Now, he is curious and wants to know which of his applications were accepted.

Input

In the first line of input file there are two integers N and M ($1 \le N \le 100$, $1 \le M \le 100$) — the number of faculties and the number of subjects for which Peter passed the tests, respectively. Next line contains M integers B_1, B_2, ..., B_M ($0 \le B_i \le 100$); here, B_i is Peter's score for i-th subject. After that come N blocks for description of the faculties.

The first line of a block contains the name of the faculty; it can contain just uppercase and lowercase Latin letters and spaces and is no longer than 100 characters. The second line contains K — the number of exams one should pass in order to enter study for any specialty on that faculty, and after that K integers — the numbers of the subjects at these exams. The third line contains K triplets of numbers $X_1 < Y_1 < Z_1$ $X_2 < Y_2 < Z_2$ ... $X_K < Y_K < Z_K$; here, X_i, Y_i and Z_i have the following meaning. Suppose Peter has a score of T for the respective subject. If $T < X_i$, Peter gets a "2" for it; if the $X_i \le T < Y_i$, he gets a "3"; if $Y_i \le T < Z_i$, a "4"; and finally, if $Z_i \le T$, Peter gets a "5".

After these three lines of a block, there comes a line with a single integer S on it ($1 \le S \le 100$). Next are S sub-blocks describing specialties. Each of the blocks consists of two lines. First of them contains the name of the specialty. Second line contains a single integer — the sum of grades necessary to pass the exam. In order for Peter's application to be accepted, he should have the sum of grades no less than this number.

The names of faculties and specialties can contain just uppercase and lowercase Latin letters and spaces, can't start or end with a space and have length between 1 and 100 characters, inclusive.

Output

For each Peter's application that got accepted, output the names of the faculty and the specialty on a single line, separated by a single space. Lines can be output in any order.

If all Peter's applications were rejected, output the word "`Army`" on the first line of the input file.

Examples

tests.in
2 3 98 87 90 Computer Science and IT 3 1 2 3 90 95 99 80 90 100 91 93 95 2 Computer Security 15 Applied Math and Informatics 14 Mathematics and Mechanics 3 1 2 3 70 80 90 80 85 90 65 76 90 2 Applied Math and Informatics 13 Applied Math in Agriculture 5
tests.out
Mathematics and Mechanics Applied Math and Informatics Mathematics and Mechanics Applied Math in Agriculture

tests.in
1 3 98 87 90 Computer Science and IT 3 1 2 3 90 95 99 80 90 100 91 93 95 1 Computer Security 15
tests.out
Army

Problem B. Game

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

Vasya loves his new game which is played on an infinite rectangular grid where K cells are initially black, all other cells are white. The move of the game is to find three black cells which are vertices of some rectangle with sides parallel to coordinate axis such that the fourth vertex of the rectangle is white. In this case you need to paint the fourth vertex black. Vasya asks you to write a program which calculates the number of black cells in the end of the game, i.e. when no more moves can be made.

Input

The first line contains an integer K ($0 \le K \le 2 \cdot 10^5$). The next K lines contain two integers each — coordinates of black cells X_i and Y_i ($-10^9 \le X_i, Y_i \le 10^9$).

Output

Output the answer to the task.

Example

game.in	game.out
3 1 1 1 2 2 2	4
5 0 0 1 0 0 1 1 2 2 1	9

Problem C. Rifleman

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

Petya is a rifleman. He is located in the leftmost bottom cell of the rectangular field $N \times M$, all other cells of the field are occupied by enemies. Petya and his enemies are points and they are located in the centers of the corresponding cells. Petya has a power supersecret weapon Ber-9875 which destroys all enemies along a straight line. Your task is to calculate the minimum number of shots to do this.

Input

The first line of the input contains two integers N and M ($1 \le N, M \le 10^6$).

Output

Output file must contain the minimum number of shots.

Example

rifleman.in	rifleman.out
3 4	7
2 10	11

Problem D. Subway

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

You need to create the plan of building subway in the city of S***. The subway must consist of some circle lines, each consisting of three or more stations, up to ten stations. Each circle line must be linked as a loop: first station is linked with second, second with third and so on. The circle lines themselves are linked as a chain: changing from one circle line to another is possible only at one station for each consecutive pair of circle lines in that chain. Other pairs of circle lines have no intersections. Any non-degenerate segment of subway is allowed to be contained only by one circle line.

Also it is needed to add radial lines, which start on some stations of circle lines and go to the distant parts of city. Unfortunately, due to lack of funds, radial line is allowed to consist only of two stations, one of them connects the radial line to circle line, and the other station is the terminal station. Terminal station cannot lie on more than one line. Any *transfer station*, which connects any two lines, could not connect more than two lines. No two radial lines are allowed to have common stations.

It is required to build exactly N stations and M segments between stations. Your task is to find the plan.

Input

First line contains two integers N and M ($1 \le N, M \le 100\,000$).

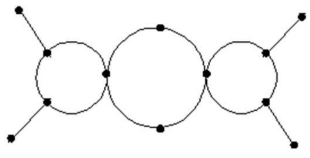
Output

First line must contain the number of circle lines K. K lines follow, each describing one circle line. Circle line is described by the number of stations on it, then the station numbers themselves in traversing order. The next line contains R — the number of radial lines. R last lines contain pairs describing radial lines. All stations are numbered from 1 to N. Your plan must consist of exactly N stations and M connecting segments. If there is no solution, output "`No solution`" without quotes.

Example

subway.in	subway.out
12 14	3 3 1 2 3 3 4 5 6 4 3 7 4 8 4 1 9 2 10 11 5 12 6
4 3	No solution

Picture describing the first sample



Problem E. Tea Party

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

Alice organizes a tea party for her classmates. There are K guests on this party and N types of tea of two kinds: black and green. During each party round, all guests who are currently present on the party get one bag of the same type of tea. After each party round, one of the guests leaves the party because he/she needs to meet somebody elsewhere. Alice needs to minimize the total cost of tea bags spent, but she also wants to make it so that all served tea types must be different, and any three consecutive rounds of the party are not using tea of the same kind. Help her.

Input

The first line of the input contains two integers K and N ($1 \le K, N \le 1000$). N lines follow, each for one tea type, containing two integers c_i and s_i each ($1 \le c_i \le 100\,000$), where c_i is the cost of one bag of i-th type, and s_i is zero for green tea and one for black tea.

Output

Output any possible optimal party plan — K different numbers between 1 and N, each for one type of tea. If it is impossible to organize the party, output "`Impossible`" without quotes.

Example

tea.in	tea.out
3 4 1 0 2 0 4 1 3 1	1 2 4

Problem F. Carlsson vs. Winnie-the-Pooh

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

Carlsson and Winnie the Pooh are eating one pizza, which is a circle of radius R. They divided the pizza into some pieces by N straight lines. They eat pizza in a following way. Any person takes a piece and eats it, then takes the next piece, eats it and so on. They eat pieces with the same speed. The time of eating a piece is proportional to its area. If at any moment of time they are both trying to take a piece simultaneously, Carlsson does his selection first. Your goal is to write the program, which determines the amounts of pizza eaten by each side, if it is known that both sides try to maximize their total amount of eaten pizza.

Input

The first line of the input contains N ($1 \le N \le 4$) and R ($1 \le R \le 100$). The center of the pizza is located at $(0, 0)$. Each of the next N lines describes one line with three integers a, b and c ($-1000 \le a, b, c \le 1000$). These numbers correspond to equation $ax + by + c = 0$. It is guaranteed that each line intersects the edge of pizza in exactly two points.

Output

Output the area of pizza eaten by Carlsson and by Winnie the Pooh. The testing program checks whether the answer is within 10^{-4} of the right answer.

Example

vs.in	vs.out
1 100 1 0 0	15707.9632679490 15707.9632679490

Problem G. Save Vasya

Input file:	`vasya.in`
Output file:	`vasya.out`
Time limit:	3 seconds
Memory limit:	64 megabytes

Vasya has huge problems. He needs to take polynomial $ax + b$, then calculate its k-th power and evaluate the sum of its coefficients. Your task is to help him.

Input

The first line of the input contains three integers a, b and k ($1 \le a, b \le 100$, $1 \le k \le 20$).

Output

The first line must contain the sum of coefficients of the resulting polynomial.

Example

vasya.in	vasya.out
1 2 2	9

Problem H. Amplifiers

Input file:	`amplifiers.in`
Output file:	`amplifiers.out`
Time limit:	3 seconds
Memory limit:	64 megabytes

Scientist Shurik needs voltage that is N times more that the standard voltage in the wall outlet for power supply for his time machine. The standard voltage is equal to one Bervolt. Shurik decided to use voltage amplifiers. In the nearby shop he found the amplifiers of two types, the first type creates voltage $2X - 1$ Bervolt from X Bervolt, the second one creates voltage $2X + 1$ Bervolt from X Bervolt. The number of amplifiers in the shop is unlimited. Shurik wants to build a sequence of amplifiers from the outlet to the time machine. Of course he wants to minimize the number of amplifiers. Help him.

Input

A single integer $1 \le N \le 2 \cdot 10^9$.

Output

If it is possible to make such scheme, output in the first line the minimal possible number of amplifiers. The second line in this case is to contain the sequence of amplifiers from the outlet to the time machine. Use number 1 for the first-type amplifiers and number 2 for second-type amplifiers.

If there is no solution, output "`No solution`" (without quotes).

Example

amplifiers.in	amplifiers.out
5	2 2 1

Problem I. Berland All-Round Competitions

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

The Berland has a tradition of all-round competitions. The rules of those competitions are the following. The first stage is swimming some distance, then there are K stages of horse riding. There are N horses located at the start of each of the riding stages (N is the number of participatnts). For each horse, its *ridingness* S_j is known. The more the ridingness is, the faster the horse finishes the stage. For each participant, there are three parameters: swimming skill W_i, riding skill R_i and strength P_i. Participant who reaches the start of some riding stage (which is the finish of the previous stage) takes the horse with maximal ridingness from those who are still at the start of this stage. Each horse can participate only in one stage. If two participants reach the start of any stage simoultaneously, the one with the most strength takes the better horse. All participants have different strengths. Time required to complete the riding stage for i-th participant is equal to $2 \cdot 10^7 - R_i - S_j$, for the swimming stage the time is $2 \cdot 10^7 - W_i$. After the start of the competition (i. e. the start of the swimming stage) there are no pauses and changing or taking the horse requires no time.

Petya knows all characteristics of participants. Also he knows that the jury picks horses in such a way that on i-th stage ($1 \le i \le K$) the ridingness of j-th horse ($1 \le j \le N$) is equal to

$$f(i, j) = 3A_i^2 + 5A_iB_j + 2B_j^2,$$

where all A_i and B_j are also known.

Now Peter wants to know the order of the participants on the finish line. Help him.

Input

The first line of the input contains two integers N and K ($1 \le N \le 10^3$, $1 \le K \le 10^3$). N lines follow, containing three integers each: W_i, R_i and P_i ($1 \le W_i, R_i, P_i \le 10^6$). Then there are two lines: the first is for A, which contains K natural numbers not greater then 10^3, the second is for B, which contains N natural numbers not greater than 10^3.

Output

Output the space-separated transposition of N numbers — the order of participants. If two participants reach the finish simultaneously, they use armwrestling to determine the winner, and the participant with higher strength takes the better place.

Example

competitions.in	competitions.out
4 2 2 5 3 3 1 4 3 4 1 2 2 2 2 3 4 5 6 7	2 3 1 4

Problem J. The Lesson of Physical Culture

Input file:	`lesson.in`
Output file:	`lesson.out`
Time limit:	3 seconds
Memory limit:	64 meagbytes

In one of the schools in Berland, the teacher of physical culture got cold during the morning excercises. After some discussion the pedagogical assembly has made a decision that the next lesson of physical culture will be driven by the teacher of mathematics Gleb Antonovich, who does not care whom, where and when he should talk about mathematics.

After coming to the gym, Gleb Antonovich noticed that the floor is the $N \times M$ grid of unit squares. The first exercise of the lesson was to stand in some cells in such a way that any student is in the center of some cell, there is no more than one student in any cell, and there are no more than two students in any 2×2 square to prevent discomfort. But because of the limited space, it was required that each square 2×2 must contain two students.

Gleb Antonovich is now interested, how many different ways of such standing exist if all students are equivalent. Your task is to find the number.

Input

The first line contains two integers N and M ($2 \le N, M \le 1000$).

Output

Output the number of ways without any leading zeros or spaces. Note that the number of students standing on the floor is not fixed.

Example

lesson.in	lesson.out
2 2	6

Problem K. Save the Fisher

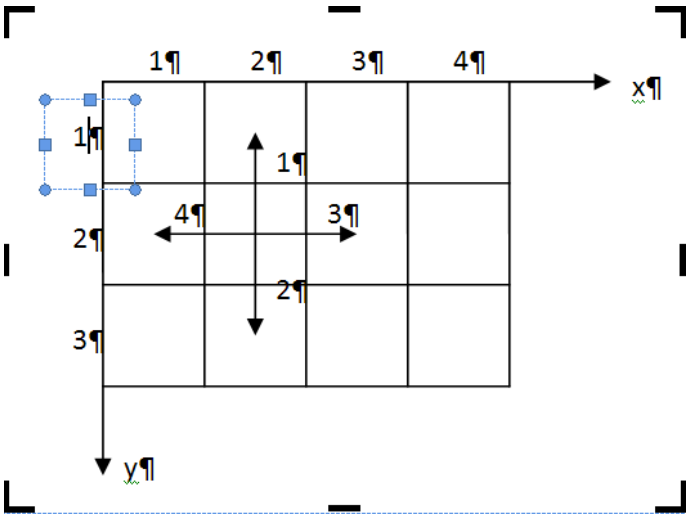
Input file:	`fisher.in`
Output file:	`fisher.out`
Time limit:	2 seconds
Memory limit:	64 megabytes

Peter has a boat. He loves to use it for fishing on the Berland Lake. But today an accident happen with him. A huge fish ate the hook, and when Peter tried to take the fish out of the lake, he suddenly discovered that the fish was stronger. Unfortunately Peter was unable to stay on the board and he fell down into the water. But he did not want to lose such a catch. So the fish took him somewhere...

The Peter is now somewhere inside the lake and lost his boat and his rod. Luckily, Peter knows each bermeter of this lake (bermeter is the length measurement unit in Berland). You will help Peter to create the plan to save himself.

The lake is a rectangle of $N \times M$ bermeters. Let us consider an $N \times M$ grid of 1×1 squares within this rectangle. Peter can move within the lake according to the following:

- When Peter is in the center of some cell, his real velocity is the sum of the vectors of the velocity of the flow and Peter's movement vector. He moves with this velocity until he reaches the center of some other cell, where his new velocity will be calculated independently of the previous velocity. The velocity could not be changed in any place other than the center of some cell.
- For each cell you know the flow velocity, it can be one of the following five vectors: 1: $(0, -1)$, 2: $(0, 1)$, 3: $(1, 0)$, 4: $(-1, 0)$, 0: $(0, 0)$. All velocities are in bermeters per minute. Peter's movement vector also is chosen among these five vectors, and he can change it any time when he is in the center of some cell. E. g. if the flow velocity is $(0, -1)$ and the Peter's movement vector is $(1, 0)$ then the resulting velocity will be $(1, -1)$, i. e. Peter will move diagonally. Note that Peter may stay in the center of any cell during any period of time (may be non-integer amount of minutes) by selecting movement vector opposite to the flow speed. If Peter moves with the same vector as the flow, he will reach the center of the next cell in half a minute.
- The boat moves according to the same laws as Peter, but the boat's movement speed is always zero.
- Peter is saved when he is in the same point as the boat (it is not neccessary to reach the center of any cell) or if he reaches the shore of the lake. The boat can also reach the shore, if it happens, the boat stops there.



Input

The first line of the input contains two integers N and M ($1 \le N, M \le 500$) separated by one space. Each of the next N lines contains M digits each — the description of the lake. The digit is the number of the flow velocity vector in the corresponding cell, according to the statement. The last two lines contain two integers each, the first corresponds to initial coordinates of Peter, the second is for the boat.

Output

Output the minimal time in minutes required for Peter to save himself, rounded up to two digits after decimal point. If Peter is not able to save himself, output "SOS".

Example

fisher.in	fisher.out
4 6 222222 444444 444444 111111 5 3 6 2	2.00
3 2 00 32 11 1 2 2 3	0.67
3 5 32240 33442 31140 2 2 5 2	SOS

Problem L. Elevator

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

There is only one elevator in the tall building with N floors. The parking for this building is at the basement floor which is located under the first floor. All floors are enumerated from 1 to N, growing up. At i-th floor there are A_i people who wish to descend from the floor to parking. You know that the elevator is unable to carry more than C people at any time. Descending or ascending one floor takes P seconds. Your task is to find the maximum possible number of people the elevator may deliver to parking within T seconds of operation, if it is located at the parking in the beginning. You may assume that stopping at a stage to load or unload people is done instantly.

Input

In the first line of input file there are four integers N, C, P, T ($1 \le N \le 100$, $1 \le C \le 10^9$, $1 \le P \le 10^9$, $1 \le T \le 10^9$). The second line contains the sequence of N integers A_1, A_2, ..., A_N ($0 \le A_i \le 10^9$). The sum of all A_i does not exceed 10^9 too.

Output

Output the maximum possible number of people who can reach the parking.

Examples

elevator.in	elevator.out
4 5 2 15 0 1 2 3	3
4 5 2 18 0 1 2 3	5
3 2 1 9 1 1 1	3