Problem A. Playing with an Array

Input file:	`array.in`
Output file:	`array.out`
Time limit:	2 seconds
Memory limit:	1024 megabytes
Feedback:	

Vasya's parents gave him an array of integers for his birthday. He decided the present is lame, but none the less he quickly invented a way to entertain himself with it. He learned how to perform two operations: to change the value of an element of the array, and to count the number of values that occur an odd number of times within the segment of the array. He started to perform the operations, and for each counting operation he wrote down the result. Your task is to repeat his operations and get the same list of results.

Input

The first line of input contains the integers N ($1 \leqslant N \leqslant 500{,}000$), the number of elements in the array, and Q ($1 \leqslant Q \leqslant 100{,}000$), the number of operations Vasya performed. The second line contains N integers a_i from the range $1, \ldots, 1000$, the elements of the array. Each of the following Q lines describes an operation. The first token on each of these lines is a character `S` or `Q`, the type of operation.

1. For operation `S` ("set"), two integers i and V, the index of the element ($1 \leqslant i \leqslant N$) and its new value V (integer from the range $1, \ldots, 1000$), follow. Vasya assigns V to the i^{th} element of the array.
2. For operation `Q` ("query") two indices ℓ and r follow ($1 \leqslant \ell \leqslant r \leqslant N$). Vasya looks at the segment of the array starting with the element a_l and ending with the element a_r (inclusive) and counts the number of values that occur an odd number of times in this segment. He writes down the result.

The tokens on each line are separated by single spaces.

Output

Output as many integers, each on a separate line, as there were operations of type `Q`. The numbers must be the results Vasya wrote down, in the order they were written.

Examples

`array.in`	`array.out`

`5 5`
`1 2 3 4 5`
`Q 1 5`
`S 1 2`
`Q 1 3`
`S 3 2`
`Q 3 3`
	`5`
	`1`
	`1`

Problem B. Airlines 3

Input file:	`avia3.in`
Output file:	`avia3.out`
Time limit:	2 seconds
Memory limit:	1024 megabytes
Feedback:	

Oh! Bytelandavia, the only 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 the 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, none the less, mandatory (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 Basilky to Old Moscow, then to Williamburg and then return to New Basilky, he will have to use at least two different companies. However, if he also passes through New Beaversk, there would be no such requirement.

Additionally, if an airline is present in timetable of two airports, then it must be possible to fly between these airports with no more than one intermediate stop using only this airline.

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

Input

The only line of input contains N, the number of airports in Byteland ($1 \leqslant N \leqslant 100$).

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 ($1 \leqslant i \leqslant N-1$) 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

avia3.in	avia3.out

3	2
	1 1
	2

4	3
	1 2 2
	2 2
	3

Note

The first problem in this series was used in this contest in 2007 and the second one in 2012.

Problem C. Card Game

Input file:	`cards.in`
Output file:	`cards.out`
Time limit:	2 seconds
Memory limit:	1024 megabytes
Feedback:	

Surely you know that in most card games (such as preference, for example) the probability of winning depends not only on the skills of the player, but also on the way the cards are dealt. The combination of the strength of the hand of a player and his skills is called the player's *standing*.

There's a card game played by N players in several rounds where the cards are dealt afresh in each round. The standing of the i^{th} player ($1 \leqslant i \leqslant N$) is a random real value drawn uniformly from the range $[a_i, b_i]$. The player with the strongest standing (as determined by the random value) wins the round. If the maximal standing in a round is shared by several players, the round ends in a draw.

Determine the probability of winning a round for each of the given players.

Input

The first line of input contains the integer N ($2 \leqslant N \leqslant 300$). Each of the following N lines contains two integers a_i and b_i ($0 \leqslant a_i < b_i \leqslant 1{,}000{,}000{,}000$) defining the range from which the standing of the player i is drawn.

Output

Output N lines, where the i^{th} line contains one real value: the probability that the i^{th} player will win the round. The values may not differ from the correct ones by more than 10^{-9}.

Examples

cards.in	cards.out

2
2 3
1 4
	0.500000000000
	0.500000000000

2
0 1
0 2
	0.250000000000
	0.750000000000

5
0 5
1 5
2 5
3 5
4 5
	0.069722222222
	0.088472222222
	0.121111111111
	0.192638888889
	0.528055555556

Problem D. Castle Site

Input file:	`castle.in`
Output file:	`castle.out`
Time limit:	2 seconds
Memory limit:	1024 megabytes
Feedback:	

Even though the bastard (the son of a noble born out of wedlock) is considered a nobleman, he has no claim on his father's lands. However, recently he has acquired a triangular plot of land. Obviously, any proper landlord needs a castle, so one has to be built. As the first step, a location needs to be selected…

An old friend (a stargazer with whom, truth to be told, many bottles of good wine have been shared) suggested after yet another drinking night: "You will need roads from the castle to all corners of the plot in any case. Why don't you build the castle so as to minimize the total cost of the roads?"

"Wise words!" the nobleman thought. There's a complication, however: a mile of the road from corner A costs three times as much as a mile of the road from corner C, while a mile of the road from corner B costs half as much as a mile from corner A…

Solving such problems is not for noblemen. So you are to figure out the optimal location for the castle.

Input

Each of the three lines of input contains two integers: the coordinates of the corners A, B, and C, respectively. The absolute values of the coordinates do not exceed 10^9 and it is known the triangle is not a degenerate one.

Output

Output two numbers: the coordinates of the castle. The values may not differ from the correct ones by more than 10^{-4}.

Examples

castle.in	castle.out

0 0
1000000000 1000000000
-999999999 -1000000000
	0 0

Problem E. Counting Rhyme

Input file:	`rhyme.in`
Output file:	`rhyme.out`
Time limit:	2 seconds
Memory limit:	1024 megabytes
Feedback:	

Eeny, meeny, miny, moe,
Catch a tiger by the toe.
If he hollers, let him go,
Eeny, meeny, miny, moe

Let's remember the backyard games form the childhood… One of them is the counting out game. Often this game preceeds another game, to select $K+1$ players among $N+1$ children. This is how it works:

N children stand in a circle. They are numbered in the order they stand, starting form 1. The remaining child is the leader of the next game and starts to recite a counting rhyme to pick the team. With each subsequent word the leader points to the next child in the circle (starting from the first child with the first pass through the rhyme).

The child pointed to with the last word of the rhyme gets picked for the next game and leaves the circle. After that the same rhyme is recited again, starting from the next child after the one who just left. The rhyme is repeated K times and the K children selected this way form the team for the next game together with the leader.

Pete has become a leader and wants to pick only his friends for the next game. Help him select a rhyme with the least possible number of words to achieve that!

Input

The first line of input contains the integers N and K ($1 \leqslant K \leqslant N \leqslant 16$). The second line contains K distinct integers, each in the range from 1 to N: the numbers of friends of Pete according to the initial ordering of children in the circle.

Output

Output one integer, the remainder from division of the number of words in the rhyme by 2013. If there's no suitable rhyme, output -1.

Examples

rhyme.in	rhyme.out

`4 2`
`1 3`
	`5`

`6 3`
`1 3 5`
	`19`

Problem F. Divisibility by 11

Input file:	`div11.in`
Output file:	`div11.out`
Time limit:	2 seconds
Memory limit:	1024 megabytes
Feedback:	

You're given a positive integer N, written with no leading zeroes. Reorder its digits to get the maximal possible number that is divisible by 11 and contains no leading zeroes.

Input

The only line of input contains the integer N ($1 \leqslant N \leqslant 10^{1000}$).

Output

On the only line of output, print the maximal possible number that can be composed from the digits of N that is evenly divisible by 11. If there's no such number, output -1.

Examples

div11.in	div11.out
1247	-1
12347	72413
12357	57321
21111	12111

Problem G. Extremal Permutations 2

Input file:	`extremal2.in`
Output file:	`extremal2.out`
Time limit:	2 seconds
Memory limit:	1024 megabytes
Feedback:	

A member a_i of the sequence $a_1, a_2, \ldots, a_N$ is called a *local extremum* if either $a_i > a_{i-1}$ and $a_i > a_{i+1}$ (local maximum) or $a_i < a_{i-1}$ and $a_i < a_{i+1}$ (local minimum). A sequence $p_1, p_2, \ldots, p_N$ is called a *permutation* of the integers from 1 to N if each of the integers appears in the sequence exactly once. A permutation is called *extremal* if each member (except the first and the last) is a local extremum.

You are given an extremal permutation of $1, \ldots, N$ and asked to find the following one in the lexicographical order of all extremal permutations of these elements. The name of the lexicographic order comes from its generalizing the order given to words in a dictionary: a sequence $x_1, x_2, \ldots, x_k$ comes before a sequence $y_1, y_2, \ldots, y_k$ if and only if the first x_i, which is different from y_i, is less than y_i.

Input

The first input line contains the integer N ($1 \leqslant N \leqslant 100{,}000$) and the second line contains N integers: the values of p_1, p_2, ..., p_N.

Output

The only line of the output file should contain N integers: the values q_1, q_2, ..., q_N such that $q_1, q_2, \ldots, q_N$ is the permutation immediately following the permutation p_1, p_2, ..., p_N in the lexicographical order of all extremal permutations of $1, \ldots, N$. If the input contains the lexicographically last extremal permutation, print the lexicographically first one as the output.

Examples

extremal2.in	extremal2.out
3 2 3 1	3 1 2
3 3 1 2	1 3 2

Note

The first task in this series was used in this contest in 2009.

Problem H. Guests

Input file:	`guests.in`
Output file:	`guests.out`
Time limit:	2 seconds
Memory limit:	1024 megabytes
Feedback:	

There are N persons, numbered from 1 to N for simplicity. For any two of them, it is known whether they know each other (if A knows B, then also B knows A). We want to invite as many of them as possible to a party, subject to the following two conditions:

- for each guest there must be no less than K others he knows;
- for each guest there must be no less than K others he does not know.

Input

The first line of input contains two integers N and K ($1 \leqslant K < N \leqslant 1000$). Each of the following N lines contains N integers, each of which is either zero or one: the j^{th} number on the input line $i+1$ indicates whether the person i knows the person j (1—knows, 0—doesn't know). The i^{th} number on the input line $i+1$ will always be 0.

Output

On the first output line, print the maximal number of guests that can be invited to the party under the conditions given above, and on the second line the numbers of persons to invite (in any order).

Examples

guests.in	guests.out

2 1
0 1
1 0
	0

3 1
0 1 0
1 0 1
0 1 0
	0

4 1
0 1 0 1
1 0 1 0
0 1 0 1
1 0 1 0
	4
	1 2 3 4

Problem I. Lumberjacks

Input file:	`lumberjacks.in`
Output file:	`lumberjacks.out`
Time limit:	2 seconds
Memory limit:	1024 megabytes
Feedback:	

Two lumberjacks, Petr and Vasya, love their job, which is to chop wood...

Once, when they had to chop several trees, they made a bet: the one who delivers the last blow will have to carry the wood back to the village. After a hard day's work neither was interested in carrying the extra weight. Both Petr and Vasya are very strong and will chop through any branch and even any trunk with a single blow of the axe. To make the bet fair, they marked in advance all the places where the trees could be chopped and added a rule not to chop further any piece that has already fallen to the ground—these will be carried home and any final chopping will be done there.

Petr is the older of the two and gets to choose who will start the chopping. Help him decide who has to start to ensure he wins the bet. If Petr has to start, also tell him which mark he should chop first.

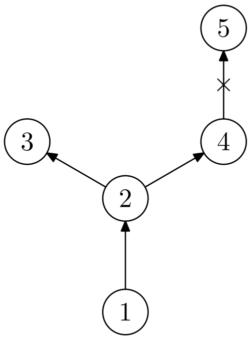
To formalize, the trees will be modeled as rooted trees. The base of a tree trunk will be modeled as the root of a tree, the chopping marks on branches and trunks as arcs (directed edges) and crotches (places where a branch splits from the trunk or another branch) as vertices.

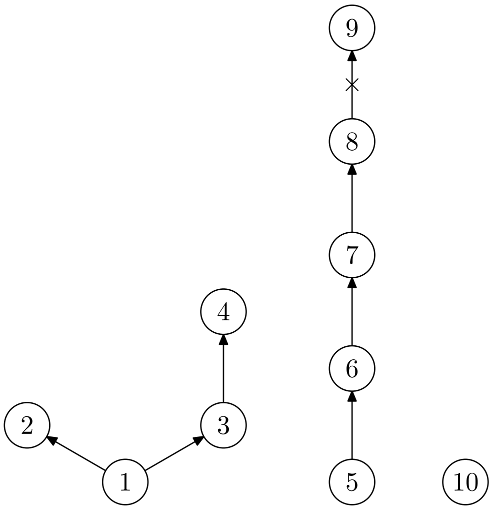
Input

The first line of input contains two integers N and M, the numbers of vertices and arcs, respectively ($1 \leqslant N, M \leqslant 100{,}000$). Each of the following M lines contains two integers x and y, denoting that there's an arc from the vertex x to the vertex y. It may be assumed that the input always describes a forest.

Output

Print the word `Vasya` on the first line of output if Vasya should start the chopping. If Petr should start, print on the first line the word `Petr` and on the second line two integers x and y, denoting that Petr should chop the arc between these two vertices with his first blow to win the bet even if Vasya plays optimally. If there are several solutions, output any one of them.





Examples

lumberjacks.in	lumberjacks.out

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

10 7
1 2
1 3
3 4
5 6
6 7
7 8
8 9
	Petr
	8 9

Problem J. Pilgrims

Input file:	`pilgrims.in`
Output file:	`pilgrims.out`
Time limit:	2 seconds
Memory limit:	1024 megabytes
Feedback:	

To celebrate one of the main religious holidays of Byteland, pilgrims flock to the church entrance...

There's a park in front of the church. The layout of the park is a grid with N lines and M columns. The park entrance is in the North-Western cell $(1, 1)$ and the church entrance, towards which the pilgrims rush, is in the South-Eastern cell (N, M). The pilgrims move in the park only towards East or South, from one cell to the next.

Some cells contain food stands for the pilgrims tired of the long walk. There are $A_{i,j}$ pieces of food in the cell (i, j). The vendors will be happy if the pilgrims eat all the food from their stands. The pilgrims, on the other hand, take one piece of food from each stand they pass where there's still any food left...

Determine the minimal number of pilgrims needed to eat all the food from all the stands.

Input

The first line of input contains the integers N and M ($1 \leqslant N, M \leqslant 100$). Each of the following N lines contains M digits, the values of $A_{i,j}$ ($0 \leqslant A_{i,j} \leqslant 9$).

Output

Output a single integer, the minimal number of pilgrims needed to eat all the food.

Examples

pilgrims.in	pilgrims.out

2 4
1111
1111
	2

5 7
2200000
0221110
0010010
0011120
0000022
	2

4 4
0110
0001
0010
1000
	3

Problem K. Skis and Skiers

Input file:	`skis.in`
Output file:	`skis.out`
Time limit:	2 seconds
Memory limit:	1024 megabytes
Feedback:	

There are N skiers in a sports school. The height of the skier i ($1 \leqslant i \leqslant N$) is denoted a_i. The school's warehouse has issued N pairs of skis to the group for a training. The length of the skies in the pair i ($1 \leqslant i \leqslant N$) is denoted b_i.

Ideally, the length of the skis should equal the length of the skier. Alas, this is not always possible... The coach has to distribute the skis so as to minimize the discomfort. More precisely, the administration of the school demands that the average absolute value of the difference between the height of the skier and the length of the skis must be minimized; that is, the coach needs such permutation π of $\{1, 2, ..., N\}$ that the value

$$f(\pi) = \frac{1}{n} \sum_{i=1}^{n} |a_i - b_{\pi(i)}|$$

would be as small as possible.

Input

The first line of input contains the value of N ($1 \leqslant N \leqslant 300$). The second line contains N integers, the values a_i ($1 \leqslant a_i \leqslant 100{,}000$). The third line contains the values b_i ($1 \leqslant b_i \leqslant 100{,}000$) in a similar fashion.

Output

Output a single number, the minimal possible average absolute value of the difference, with the absolute or relative error not exceeding 10^{-6}.

Examples

skis.in	skis.out

3
2 5 6
1 6 7
	1.000000000

5
2 3 5 5 1
1 3 5 2 5
	0.000000000

Problem L. One More Round on the Spiral. . .

Input file:	`spiral.in`
Output file:	`spiral.out`
Time limit:	2 seconds
Memory limit:	1024 megabytes
Feedback:	

An infinite sheet of graph paper is filled with the sequence of positive integers written down on a spiral. An example of the beginning of one such spiral (there are several possible directions):

37	36	35	34	33	32	31
38	17	16	15	14	13	30
39	18	5	4	3	12	29
40	19	6	1	2	11	28
41	20	7	8	9	10	27
42	21	22	23	24	25	26
43	44	45	46	47	48	. . .

The coordinates of the starting point of the spiral and the spiral's direction are not known. It is known, however, that the cell $(0, 0)$ contains the value A. What is the maximal possible value in the cell (x, y)?

Input

The first line of input contains the value A, a positive integer not exceeding 10^8. The second line contains two integers x and y, the coordinates of the cell we're interested in ($|x|, |y| \leqslant 10^8$).

Output

Output a single integer, the maximal value that could be in the cell (x, y).

Examples

`spiral.in`	`spiral.out`

`16`
`1 6`
	`251`

`16`
`16 16`
	`1296`