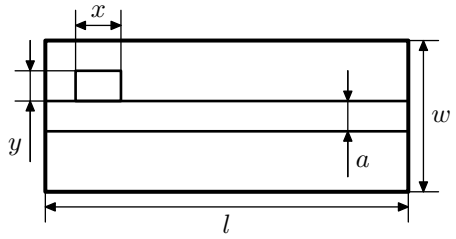
Problem A. Aisle seat please

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

People who travel by planes frequently like to ask for an aisle seat. If you have an aisle seat you can stand and walk at any time without disturbing your neighbours.

The «Aerotram» company is developing new aircraft «T-239-n». Engineers need to construct salon so that the number of aisle seats will be maximized. Let's use the following model of the aircraft. Horizontal projection of the salon is a rectangle of length l and width w centimeters. The chair is a rectangle of size x by y centimeters and is located in the salon so that the side of length x is parallel to the side of the salon with length l. Corridor is a strip with width a parallel to the side of the salon with length l. Corridor is going through the whole salon.



You need to find places for n chairs in the salon so that the number of chairs near the corridor will be maximized. You should arrange at least one corridor in the salon. A chair is located near the corridor if it has at least one common side with the corridor.

Input

Input file contains six integer numbers: n, l, w, x, y and a ($1 \le n \le 10\,000$, $1 \le l, w, x, y, a \le 10^4$).

Output

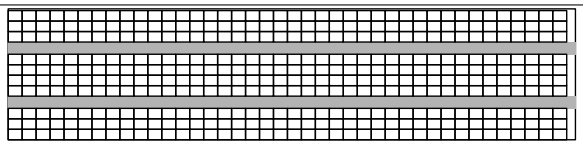
If it is impossible to arrange n chairs in the salon so that there will be at least one corridor — print only number «−1». Otherwise, print the maximal number of chairs possible to arrange near the corridor.

Examples

aisle.in	aisle.out
400 3250 750 80 60 70	160
450 3250 750 80 60 70	-1

Note

The optimal arrangement for the first sample is following:

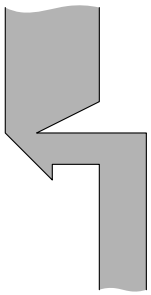


Problem B. Bridge

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

Flatland government decided to build a new bridge across the Down Flat river going from the south to the north through the whole country territory. Due to the financial crisis they decided to build the bridge of the minimal possible length.

Let's introduce the coordinate system so that OY axis is oriented from the south to the north, OX axis — from the west to the east. River banks are polylines that are infinite in both direction. Left river bank starts with a beam directed to the south from the point $(x_{1,1}, y_{1,1})$, continues with segments $(x_{1,1}, y_{1,1}) - (x_{1,2}, y_{1,2})$, $(x_{1,2}, y_{1,2}) - (x_{1,3}, y_{1,3}), \ldots, (x_{1,m-1}, y_{1,m-1}) - (x_{1,m}, y_{1,m})$ and ends with a beam directed to the north from the point $(x_{1,m}, Y_{1,m})$. Similarly, the right river bank starts with a beam directed to the south from the point $(x_{2,1}, y_{2,1})$, continues with segments $(x_{2,1}, y_{2,1}) - (x_{2,2}, y_{2,2})$, $(x_{2,2}, y_{2,2}) - (x_{2,3}, y_{2,3}), \ldots, (x_{2,n-1}, y_{2,n-1}) - (x_{2,n}, y_{2,n})$ and ends with a beam directed to the north from the point $(x_{2,n}, y_{2,n})$.



Find the minimal possible length of the bridge.

Input

First line of the input file contains integer number m ($2 \le m \le 100$). The following m lines contain two integer numbers each — coordinates of the vertices of the left river bank: $x_{1,1}, y_{1,1}$, $x_{1,2}, y_{1,2}$, $\ldots$, $x_{1,m}, y_{1,m}$.

The next line of the input file contains integer number n ($2 \le n \le 100$). The following n lines contain two integer numbers each — coordinates of the vertices of the right river bank: $x_{2,1}, y_{2,1}$, $x_{2,2}, y_{2,2}$, $\ldots$, $x_{2,n}, y_{2,n}$.

It is known that $x_{1,1} < x_{2,1}$, every polyline does not have any selfintersections and selftouches. Polylines do not have common points. All segments of every polyline have positive length. All coordinates are bounded by 10^4 in an absolute value.

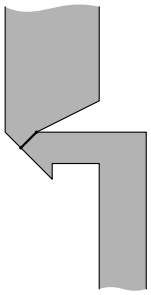
Output

Print one floating number to the ouput file: the minimal possible bridge length. Your answer will be checked with precicssion 10^{-5}.

Examples

bridge.in	bridge.out
4 6 1 3 1 3 0 0 3 3 9 3 2 3 6 5	1.41421356237309505

The optimal bridge position for the example is shown on the following picture:



Problem C. Almost nonprefix codes

Input file:	codes.in
Output file:	codes.out
Time limit:	2 seconds
Memory limit:	256 Megabytes

In coding theory often uses *nonprefix codes*. These are the set of words from which we can't select a pair such that the first word is a prefix [1] of the second. For example, a set of words «aba», «aa» and «bac» is a nonprefix code. And a set of words «abac», «aba», «ba» not because word «aba» is a prefix of word «abac».

Professor Encrypto works in unimportant information research laboratory. He explores his new invention named *almost nonprefix codes*. A set of words can be called *almost nonprefix codes* of level k if lengths of greatest common prefix for all pairs of words from set do not exceed k. For example, a set of «abac», «ab» and «ba» can be called almost nonprefix code of level 2. But a set of «abac», «abab», «ba» can't because greatest common prefix of words «abac» and «abab» has length 3.

Professor Encrypto now wants to solve a new problem. He wants to select the maximal number of words from a given set of words that can form an almost nonprefix code of given level k. You are the junior assistant of the professor and you need to write program that solves given tasks.

Input

The first line of the input contain two integers n and k ($1 \le n \le 100\,000, 0 \le k \le 200$) — the number of given words and the level of almost nonprefix code ($1 \le n \le 100\,000$, $0 \le k \le 200$).

The following n lines contain a single word each. These words consist of lowercase latin letters and their length is in range from 1 to 200 characters. The total length of all words not greater than 10^6. And of course all words are different.

Output

The first line should contain one integer n — the maximal possible size of set selected from given one that can be called almost nonprefix code of level k.

The following m lines should contain selected words.

Examples

codes.in	codes.out
6 2 aba bacaba abacaba baca abac caba	3 aba bacaba caba

Problem D. Cut the cake!

Input file:	cake.in
Output file:	cake.out
Time limit:	3 seconds
Memory limit:	256 Megabytes

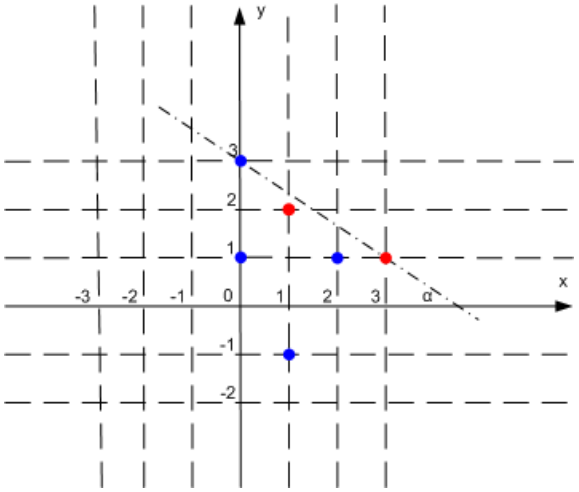
Pasha gets a cake on his birthday. On this cake there are two types of candies: blue and red. Pasha wants to cut himself a piece of cake with at least one red candy, but without blue candies. If you look on the cake from upwards it has a square shape. All candies are located stictly inside this square.

[1] Word α is called the prefix of a word β if α can be gotten from β by deleting zero or several character in the end. For example, words «», «a», «ab» and «aba» are prefixes of word «aba»

To cut a piece Pasha wants to make one straight cut. Between several cuts he prefers the cut with the smallest angle to the axis OX.

Your task is to check could Pasha cut the cake as he wants, and if he could then calculate the smallest angle. If the candy lies on the cut, then we could choose on what side we put it.

On the picture you could see the optimal cut for the first example.



Input

The first line of the input contains one integer number n ($1 \le n \le 100000$) — the number of red candies. Each of the next n lines contains two integer — the coordinates of the candies.

Next line contains one integer number m ($1 \le n \le 100000$) — the number of blue candies. Each of the next m lines contain two integers — the coordinates of the candies.

No two points are coincide. All coordinates do not exceed 10^6 by the absolute value.

Output

If it is impossible to make the satisfied cut, then you should print "*Impossible*", Otherwise, print the minimal angle between the cut and the axis OX. The angle should be in radians. The error of 10^{-4} is allowed.

Examples

cake.in	cake.out
2 1 2 3 1 4 0 1 2 1 0 3 1 -1	0.5880026035475675
1 1 1 2 0 1 2 1	0
1 10 10 4 10 11 10 9 11 10 9 10	Impossible

Problem E. Depth First Search

Input file:	dfs.in
Output file:	dfs.out
Time limit:	2 seconds
Memory limit:	256 Megabytes

Once upon a time on some programming course lesson Petya learn depth first search algorithm. This is very useful algorithm to solve many problems about graphs. Petya tries to write code of DFS immediately on his favourite programming language.

```
int a[maxn + 1][maxn + 1];
int visited[maxn + 1];

void dfs(int v) {
  printf("%d\n", v);
  visited[v] = 1;
  for (int i = 1; i <= n; i++) {
    if ((a[v][i] != 0) &&
        (visited[i] == 0)) {
      dfs(i);
      printf("%d\n", v);
    }
  }
}

void graph_dfs() {
  for (int i = 1; i <= n; i++) {
    if (visited[i] == 0) {
      dfs(i);
```

```
    }
  }
}
```

Petya's program stores graph using adjacency matrix in array «a». Vertices are numbered from 1 to n. In array «visited» he marks which vertex was already visited by our search.

Petya call functions «graph_dfs» for some undirected graph G with n vertices and he saves the output. Unfortunately, he forgets what graph did he use. So, he wants to know how many edges might be in given graph G. Help him please!

Input

The first line of the input contains two integers n and l — the number of vertices in graph and amount of numbers in sequence printed by Petya's program ($1 \le n \le 300$, $1 \le l \le 2 \cdot n - 1$).

The following l lines contain single integer each — output of Petya's program. It is guaranteed that there exists at least one graph on which Petya's program gives given outputs.

Output

The first line of the output shoudl contain single integer n — the maximal possible number of edges in given graph.

The following m lines might contain two integers each — the indices of vertices connected by edge. The returned graph should not contain nor loop, nor parallel edges in this graph.

Examples

dfs.in	dfs.out
6 10 1 2 3 2 4 2 1 5 6 5	6 1 2 1 3 1 4 2 3 2 4 5 6

Problem F. Gems

Input file:	gems.in
Output file:	gems.out
Time limit:	2 seconds
Memory limit:	256 Megabytes

In one eastern country camelcades travel to the present days. Merchants carry spices, jewellery and expensive fabrics. Of course main target of merchants is to sell goods as expensive as they can. One camelcade recently came to palace of one imperious shah.

Merchants want to sell to the shah n gems which they brought to the palace. So they put these gems in front of the shah in one row. Then the shah estimates gems and think about buying them.

There are little amounts of types of gems. Total is just 26. So we can represent types of gems with lowercase latin letters.

Let us look at the way how the shah estimates gems:

- He likes some pairs of types of gems (a_1, b_1), (a_2, b_2), $\ldots$, (a_k, b_k). He calls these pairs *beautiful*.
- He presents the gems in front of him as a string s with length n.
- The shah counts number of pairs (i, j) such that $1 \le i < j \le n$ and gems s_i and s_j form a beautiful pair, i. e., there exists $1 \le q \le k$ such that $s_i = a_q$ and $s_j = b_q$.
- If the number of beautiful gems is large then shah buy all of them.

Unfortunately, this time merchants brought very large amount of gems and shah can't calculate this number by himself. So he asks you to help him.

Input

The first line of the input contains two integers n and k ($1 \le n \le 100\,000$, $1 \le k \le 676$) — the number of gems and the number of beautiful pairs of gems.

The second line contains a string that presents types of the gems brought by merchants.

The following k lines contain two latin lowercase letter each — pair of gems such that the shah thinks it's beautiful.

Output

The sole line of the output should contain single integer — number of beautiful pairs of gems.

Examples

gems.in	gems.out
7 1 abacaba aa	6
7 3 abacaba ab ac bb	7

Problem G. Interesting numbers

Input file:	numbers.in
Output file:	numbers.out
Time limit:	2 seconds
Memory limit:	256 Megabytes

Vitalic collects numbers which seems interesting for him. Now he thinks that positive number is *interesting* if its representation in the numeral system with base k ends with odd number of zeroes. For example, if $k = 2$ then $2_{10} = 10_2$ and $24_{10} = 11000_2$ are interesting numbers.

Vitalic wants to update his collection so he wants to find n-th interesting number in increasing order. But n can be big and it might be difficult to do it himself.

Your task is to help him. Write a program that finds the number he needed to update collection.

Input

Single line of the input contains two integer numbers n and k ($1 \le n \le 10^{15}$, $2 \le k \le 10$).

Output

The sole line of the output should contain one integer number — n-th interesting number in increasing order. You need print it in decimal numeral system.

Examples

numbers.in	numbers.out
1 2	2
10 10	110

Problem H. Optimize

Input file:	optimize.in
Output file:	optimize.out
Time limit:	2 seconds
Memory limit:	256 Megabytes

There are n workers in «QQQ» company. The new project consists of m independent parts. Project manager analyses time needed to complete each part of the project. Suppose that duration of work does not depend on worker. Then he distributes all m parts of work between all n workers in some way. (one worker could get more than one part) As the result some workers need more time to finish their work than the others (because they have work). So, Project manager wants to update his distribution by using the following procedure:

- Choose two different workers.
- Choose one part of project which was given to the first worker.
- Choose one part of project which was given to the second worker.
- Give the first chosen part to the second worker.
- Give the second chosen part to the first worker.

If after this operation the maximum of times needed by first worker and second worker decreased then we call that operation to be *optimization* operation. For example, consider the project consisting of five parts with completion times 3, 6, 4, 8, 2, and there are three workers. Let the distribution be

- The first and the second parts were distributed to the first worker (total time he needed is $3+6=9$).
- The forth part was distributed to the second worker (total time he needed is 8).
- The third and the fifth parts were distributed to the third worker (total time he needed is $4+2=6$).

If we apply described operation on the first and the third worker, and swap first and fifth tasks, the first worker will finish in $6+2=8$ total time and the third in $3+4=7$. Because, $\max(9,6) > \max(8,7)$ this operation is optimization operation.

You are given a number of workers in company, number of parts in project, time needed to complete every part and the distribution. You need to calculate the number of different optimization operations on this distribution.

Input

The first line contains two integers n and m — the number of workers in company and the number of parts in project ($1 \le n, m \le 10^5$).

The second line contains m natural numbers t_i — the time needed to complete i-th part of the project ($1 \le t_i \le 10^9$).

The following n lines contain the distribution: number of parts distributed to worker and theirs indies.

Output

The sole line of the ouput should contain the number of optimization operation.

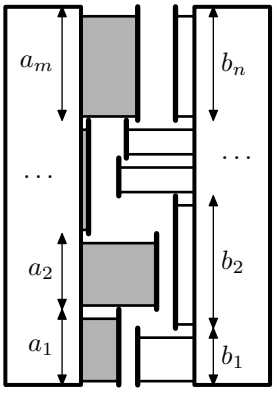
Examples

optimize.in	optimize.out
3 5 3 6 4 8 2 2 1 2 1 4 2 3 5	2
2 4 1 2 3 4 2 1 2 2 3 4	4

Problem I. Shelves

Input file:	shelves.in
Output file:	shelves.out
Time limit:	2 seconds
Memory limit:	256 Megabytes

Regional branch of one well-known bank buys two large safes for storing some documents about clients. Every safe consists of several shelves with different heights. Looking from bottom to top the heights of the shelves in the first safe are $a_1, a_2, \ldots, a_m$ and in the second safe are $b_1, b_2, \ldots, b_n$.



Safes were set in front of each other and very close to each other. So it's impossible to use two shelves that are on same hight at the same time. Bank employees take documents very often so it is more convenient to open shelves for whole day.

It is not necessary to use every shelf because branch hasn't many clients at that moment. Employees decided to use so many shelves as possible such that they can be opened at the same time.

You need to help employees to find this maximum amount.

Input

The first line of the input contains two integer numbers n and m — the number of shelves in the first safe and in the second safe ($1 \le n, m \le 100\,000$).

The second line contains n integer numbers $a_1, a_2, \ldots, a_n$ — the heights of shelves in the first safe ($1 \le a_i \le 10^9$).

The third line contains m integer numbers $b_1, b_2, \ldots, b_m$ — the heights of shelves in the second safe ($1 \le b_i \le 10^9$).

Output

In the first line of the output print two integers k and l — the amounts of open shelves in the first and the second safes. You need to maximize sum $k + l$.

In the second line print k integers — indices of open shelves in the first safe.

In the third line print l integers — indices of open shelves in the second safe.

Examples

shelves.in	shelves.out
5 5 1 2 3 4 5 6 4 3 2 1	3 4 1 2 3 2 3 4 5

Problem J. Prince

Input file:	prince.in
Output file:	prince.out
Time limit:	2 seconds
Memory limit:	256 Megabttes

After a very long adventure our prince became very close to the princess. There is only one corridor left filled with traps.

The corridor is the line that is infinite in both directions. At the beginning Prince is in the point 0. In x meters from him he sees a door that leads to Princess. In one second, Prince could run 1 meter in any direction or rest in place.

Using Sands of time, Prince already discovered n traps. These traps work as follows: i-th trap will appear a_i seconds after the beginning and disappear t_i seconds after appearance. The trap takes the part of the corridor from l_i to r_i meters (the enumeration of meters is done from Prince in direction of Princess), and if Prince accidentally will ran inside of a trap he will die. He could safely be at the ends of the traps, and could enter the door even at the time when trap appears.

Given the description of traps, you have to calculate the minimal time when Prince could enter the door to Princess.

Input

The first line of the input contains two integer numbers n and x ($0 \le n \le 1000, 1 \le x \le 10^5$) — the number of traps and the positions of a door.

The following n lines contain the description of traps. Each description consists of four integers a_i, t_i, l_i and r_i ($1 \le a_i, t_i \le 10^6, -10^6 \le l_i < r_i \le 10^6$) — the time of appearance, the duration of existence, the coordinates of left and right ends. Traps could intersect.

Output

If Prince could not enter the door at all, print "*Impossible*". Otherwise, print one integer — the minimal time at which Prince could get to the door.

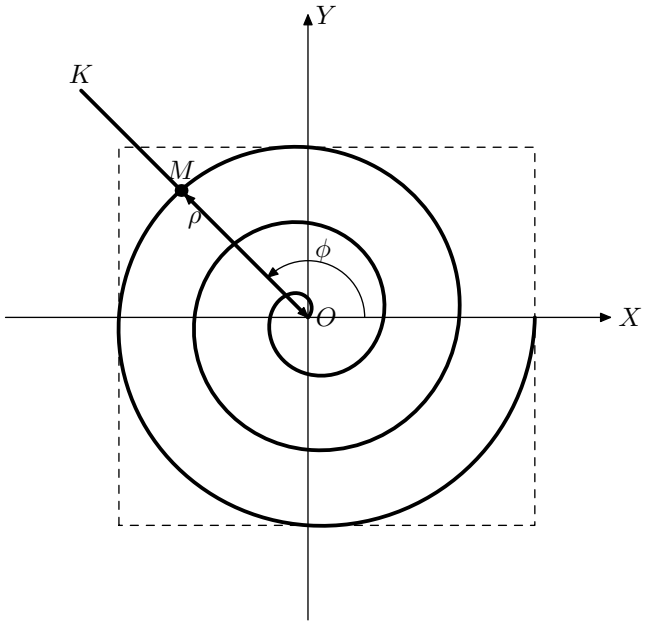
Examples

prince.in	prince.out
3 2 1 1 -1 2 3 2 1 3 6 1 0 2	6

Problem K. Arhimed spiral

Input file:	spiral.in
Output file:	spiral.out
Time limit:	2 seconds
Memory limit:	256 Megabytes

Dima has got a new work. And his first task was to investigate *Arhimed spirals*. Arhimed spiral — the curve on a plane, that shows the trajectoyy of a point M, that move with constant velocity along ray OK with start in O, while at the same time OK turn with constant velocity around O (see picture). In other words, the distance of the point to center $\rho = OM$ depends linearly on the angle ϕ of ray OK.



The movement of point M could be defined with several parameters:

- starting angle α of OK (in degrees, counter-clockwise from the positive direction of OX);
- angle velocity ω of OK (in degrees per unit of time);
- starting distance OM;
- movement velocity V of point M on ray OK.

Given these parameters without bounding the time of movement, the trajectory becomes infinite, that makes it difficult to investigate. Dima suggested to look only on the part of the trajectory that is built starting from time 0 and ending at time T. Dima wants to find a rectangle with the smallest area that fully cover the trajectory and has sides parallel to axes. Help him!

Input

The only line of the input contains four integer numbers: ω ($1 \le \omega \le 100$), V ($1 \le V \le 100$), R ($0 \le R \le 100$) and T ($1 \le T \le 1000$). In this problem we consider α to be equal to 0.

Output

The first line of the output should contain two real numbers — the coordinates of left-bottom corner of the rectangle. The second line of the output should contain two real numbers — the coordinates of righ-upper corner of the rectangle.

The answer will be accepted, if the values are differs from expected by no more than 10^{-5}.

Examples

spiral.in	spiral.out
60 10 0 18	-150.302843 -165.275488 180.000000 135.336204

Problem L. Sum

Input file:	sum.in
Output file:	sum.out
Time limit:	2 seconds
Memory limit:	256 Megabytes

Vasya likes to do some interesting things with numbers. Today he has three numbers a, b and c in his notepad and he wants to know if he can multiply some of these numbers by some power of ten to make the sum of first two numbers be equal to third number. For example, if he has 9, 34 and 43 it is possible to do because the

sum of 9 and 34 already equals to 43. As for the second example, if he has numbers 23, 7 and 93, he can multiply 7 by 10, and get 70. Then $23 + 70 = 93$.

You are given three natural numbers a, b and c. You need to find three nonnegative integer numbers n, m and k such that $a \cdot 10^n + b \cdot 10^m = c \cdot 10^k$.

Input

The first line of input contains one integer a, the second line — number b, and the third — line number c. Each number is not less than one and is not greater than $10^{100\,000}$.

Output

If such numbers exist print `YES` on the first line. In this case the second line of the output shoudl contain three integers n, m and k separated by spaces. Numbers must not be greater than 10^6. If there exist more than one answer, you can print any of them. Otherwise, print `NO` on the first line.

Examples

sum.in	sum.out
9 34 43	YES 0 0 0
23 7 93	YES 0 1 0
1 2 4	NO