icpc international collegiate programming contest

ICPC Asia Pacific Contests

Asia Yokohama Regional Contest

Official Problem Set

asia pacific

2022 2023

icpc.foundation

icpc global sponsor programming tools

icpc diamond multi-regional sponsor

Problem A
Hasty Santa Claus
Time Limit: 2 seconds

Hasty Santa Claus has arrived at the town on December 1st. Realizing that it is a little bit too early for Christmas, he plans to leave the presents before (or even after) Christmas while families are out on vacation trips.

Santa knows which families depart and return on which days, but he can visit only a limited number of houses a day. He is stuck with finding which houses are to be visited on which days to distribute the presents to every family. Please help him solving the problem, not only for Santa but also for kids anxiously awaiting for the presents!

Input

The input consists of a single test case of the following format.

$$\begin{array}{l} n\ k \\ a_1\ b_1 \\ \vdots \\ a_n\ b_n \end{array}$$

The first line has two positive integers, n and k, the number of houses to leave the presents and the maximum number of houses that Santa Claus can visit a day, respectively.

The i-th line of the following n lines has two positive integers a_i and b_i. They indicate that he can visit the i-th house between the a_i-th and b_i-th days, inclusive.

n and k satisfy $1 \le k \le n \le 1000$. For each i, a_i and b_i satisfy $1 \le a_i \le 25 \le b_i \le 31$.

Output

Print n lines of one integer describing a plan for Santa to complete his task. The integer on the i-th line means the date on which Santa should visit the i-th house.

At least one solution is guaranteed to exist. If there are two or more solutions, any of them is accepted.

Sample Input 1	Sample Output 1
5 1 23 25 23 27 24 25 25 25 25 26	23 27 24 25 26

Sample Input 2	Sample Output 2
7 2 1 31 1 31 1 31 1 31 1 31 1 31 1 31	1 1 2 2 3 3 4

Sample Input 3	Sample Output 3
6 2 24 25 24 25 24 25 25 26 25 26 25 26	24 25 24 26 25 26

The first sample is depicted in the figure below. Santa can leave the presents during the periods shown as horizontal lines with short vertical markers at both ends. For the House 4, Santa can visit only on a specific day. The triangles show the days on which Santa should visit each house.

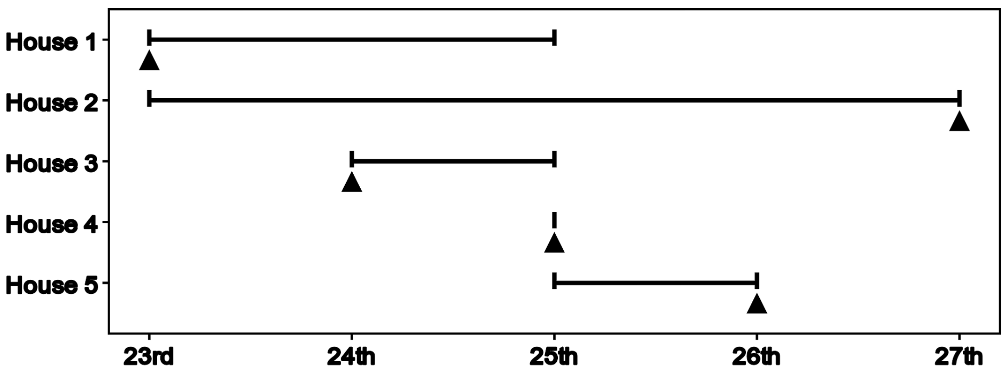


Figure A.1. Sample 1

Problem B
Interactive Number Guessing
Time Limit: 2 seconds

This is an interactive problem.

Your task is to write a program that guesses a secret number through repetitive queries and responses. The secret number is a non-negative integer less than 10^{18}.

Let x be the secret number. In a query, you specify a non-negative integer a. In response to this, the *digit sum* of $(x + a)$ will be returned. Here, the digit sum of a number means the sum of all the digits in its decimal notation. For example, the digit sum of 4096 is $4 + 0 + 9 + 6 = 19$.

Interaction

You should start with sending a query to the standard output and receiving the response from the standard input. This interaction can be repeated a number of times. When you have become confident of your guess through these interactions, you can send your answer.

A query should be in the following format, followed by a newline.

`query` a

Here, a is an integer between 0 and $10^{18} - 1$, inclusive. In response to this query, the digit sum of $(x + a)$, where x is the secret number, is sent back to the standard input, followed by a newline.

You should send your answer to the standard output in the following format, followed by a newline.

`answer` y

Here, y is the secret number you have identified, an integer between 0 and $10^{18} - 1$, inclusive.

You can send the answer only once, so you have to become sure of your guess through repeated interactions before sending the answer. However, you can send *no more than 75 queries,* and thus you have to choose your queries cleverly. After sending the answer, your program should terminate without any extra output.

Notes on interactive judging

When your output violates any of the conditions above (invalid format, a or y being out of the range, too many queries, an extra output after sending your answer, and so on), your submission will be judged as a wrong answer. As some environments require flushing the output buffers, make sure that your outputs are actually sent. Otherwise, your outputs will never reach the judge.

Read	Sample Interaction	Write
		`query 3`
`15`		
		`query 25`
`1`		
		`answer 75`

In this example, the secret number x is 75. In response to the first query, 15 is returned because $x + a = 75 + 3 = 78$ and its digit sum is $7 + 8 = 15$. After the second response, you can conclude that the only possible secret number is 75.

Problem C
Secure the Top Secret
Time Limit: 2 seconds

You are responsible for the security of ICPC (the Institute for Computer Program Critiques). The institute is in a one-storied building. Its rectangular floor is partitioned into square sections of the same size in a grid form. Two sections are said to be adjacent if they share an edge. Some of the sections in the building are blocked. All the sides of a blocked section are walled up and no entry is possible. All the other sections have no walls in between, and adjacent sections normally intercommunicate. However, roll-down shutters are equipped between some of the adjacent sections, so that closing such a shutter makes direct moves between the two sections impossible.

The top-secret research is being conducted in one of the outermost sections of the building. The section is called the top-secret section. The building has only one entrance at one of its outermost sections, which should be the only entry to the building. However, you have noticed that a window of one of the outermost sections is so fragile that it may allow trespassers to enter the building.

You must secure the top secret from trespassers. To do so, you may have to close some of the shutters so that breaking two or more closed shutters is required to make a route from the section with the fragile window to the top-secret section. In addition, there should exist at least one route from the entrance section to the top-secret section with no shutters closed on it.

You are to write a program that finds the minimum number of shutters to close to secure the top secret.

Input

The input consists of a single test case of the following format.

$$
\begin{array}{l}
n\ m \\
s_1 \\
\vdots \\
s_n \\
k \\
r_1\ c_1\ d_1 \\
\vdots \\
r_k\ c_k\ d_k
\end{array}
$$

n and m are integers between 2 and 100, inclusive, representing that the building floor has n rows and m columns of sections. The section in the j-th column of the i-th row is identified as section (i, j). The i-th line of the following n lines has a string s_i of length m describing the sections in the i-th row. Each character of s_i is one of `.`, `#`, `S`, `T`, and `U`. If the j-th character of s_i is `#`, section (i, j) is blocked and is impassable; otherwise, the section is passable. The j-th character of s_i being `S` means that section (i, j) is the entrance section, `T` means the top-secret section, and `U` means the entry point of the trespassers, that is, the section with a fragile window. Each of `S`, `T`, and `U` occurs exactly once in the input as an outermost section. The top-secret section is reachable from the entrance through passable sections with no shutters closed.

k is the number of the shutters in the building. The i-th line in the following k lines describes a shutter by two integers r_i and c_i, and a character d_i. d_i is either `r` or `b`. If d_i is `r`, $1 \leq r_i \leq n$ and $1 \leq c_i < m$ hold, and a shutter is equipped between sections (r_i, c_i) and $(r_i, c_i + 1)$. If d_i is `b`, $1 \leq r_i < n$ and $1 \leq c_i \leq m$ hold, and a shutter is equipped between sections (r_i, c_i) and $(r_i + 1, c_i)$. The same combination of r_i, c_i, and d_i appears only once. Each shutter is equipped only between two unblocked adjacent sections.

Output

Output a single integer in a line which is the minimum number of shutters to close to secure the top secret. If that is not possible, output -1. If trespassing to the top-secret section is not possible with all shutters open, output 0.

Sample Input 1	Sample Output 1
3 3 S.. #.. U.T 7 1 2 b 1 3 b 2 2 b 2 2 r 2 3 b 3 1 r 3 2 r	3

Sample Input 2	Sample Output 2
2 2 ST .U 4 1 1 r 1 1 b 1 2 b 2 1 r	-1

Sample Input 3	**Sample Output 3**
7 10 U......... .... ###....... ....### .... S........T 18 4 4 r 5 4 r 6 7 r 7 7 r 3 4 b 3 5 b 3 6 b 3 7 b 3 8 b 3 9 b 3 10 b 5 1 b 5 2 b 5 3 b 5 4 b 5 5 b 5 6 b 5 7 b	14

Sample Input 1 is depicted in the following figure. The dotted lines represent where the shutters are equipped.

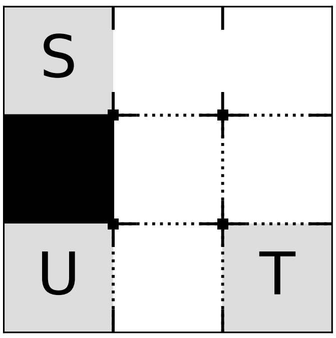


Figure C.1. Sample Input 1

Problem D
Move One Coin
Time Limit: 2 seconds

Some coins are placed on grid points on a two-dimensional plane. No two coins are stacked on the same point. Let's call this placement of coins the *source pattern.* Another placement of the same number of coins, called the *target pattern,* is also given.

The source pattern does not match the target pattern, but by moving exactly one coin in the source pattern to an empty grid point, the resulting new pattern matches the target pattern. Your task is to find out such a coin move.

Here, two patterns are said to *match* if one pattern is obtained from the other by applying some number of 90-degree rotations on the plane and a parallel displacement if necessary, but without mirroring. For example, in the source pattern on the left of Figure D.1, by moving the coin at $(1, 0)$ to $(3, 1)$, we obtain the pattern on the right that matches the target pattern shown in Figure D.2.

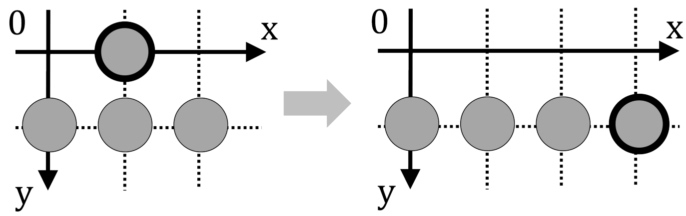


Figure D.1. Source pattern and a move

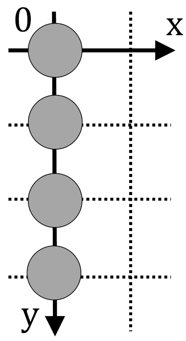


Figure D.2. Target Pattern

Input

The input consists of a single test case of the following format.

h w
$p_{0,0} \cdots p_{0,w-1}$
$\vdots$
$p_{h-1,0} \cdots p_{h-1,w-1}$
H W
$P_{0,0} \cdots P_{0,W-1}$
$\vdots$
$P_{H-1,0} \cdots P_{H-1,W-1}$

The first line consists of two integers h and w, both between 1 and 500, inclusive. h is the height and w is the width of the source pattern description. The next h lines each consisting of w characters describe the source pattern. The character $p_{y,x}$ being `o` means that a coin is placed at (x, y), while `x` means no coin is there.

The following lines with integers H and W, and characters $P_{y,x}$ describe the target pattern in the same way.

Output

If the answer is to move the coin at (x_0, y_0) to (x_1, y_1), print x_0 and y_0 in this order separated by a space character in the first line, and print x_1 and y_1 in this order separated by a space character in the second line.

It is ensured there is at least one move that satisfies the requirement. When multiple solutions exist, print any one of them.

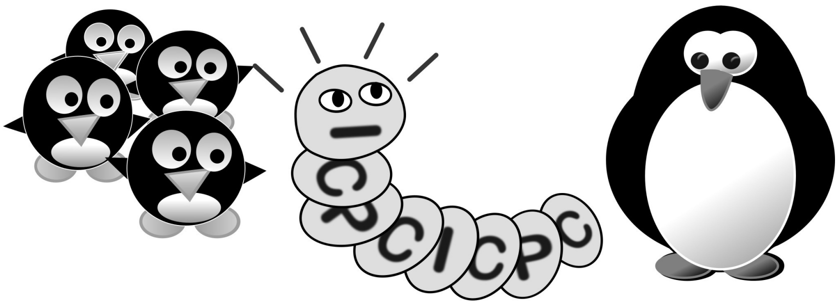
Note that $0 \le x_0 < w$ and $0 \le y_0 < h$ always hold but x_1 and/or y_1 can be out of these ranges.

Sample Input 1	Sample Output 1
2 3 xox ooo 4 2 ox ox ox ox	1 0 3 1

Sample Input 2	Sample Output 2
3 3 xox oxo xox 4 4 oxxx xxox xoxo xxxx	1 2 -1 -1

Problem E
Incredibly Cute Penguin Chicks
Time Limit: 6 seconds

The *Incredibly Cute Penguin Chicks* love to eat worms. One day, their mother found a long string-shaped worm with a number of letters on it. She decided to cut this worm into pieces and feed the chicks.



The chicks somehow love *International Collegiate Programming Contest* and want to eat pieces with *ICPC-ish* character strings on them. Here, a string is ICPC-ish if the following conditions hold.

- It consists of only C's, I's, and P's.
- Two of these three characters appear the same number of times (possibly zero times) in the string and the remaining character appears strictly more times.

For example, ICPC and PPPPPP are ICPC-ish, but PIC, PIPCCC and PIPE are not.

You are given the string on the worm the mother found. The mother wants to provide one or more parts of the worm all with an ICPC-ish string, without wasting remains. Your task is to count the number of ways the worm can be prepared in such a manner.

Input

The input is a single line containing a character string S consisting of only C's, I's, and P's, which is on the string-shaped worm the mother penguin found. The length of S is between 1 and 10^6, inclusive.

Output

Print in a line the number of ways to represent the string S as a concatenation of one or more ICPC-ish strings modulo a prime number $998\,244\,353 = 2^{23} \times 7 \times 17 + 1$.

Sample Input 1	Sample Output 1
`ICPC`	`2`

Sample Input 2	Sample Output 2
`CCCIIIPPP`	`69`

Sample Input 3	Sample Output 3
`PICCICCIPPI`	`24`

In the first sample, the string “ICPC” can be represented in the following two ways.

- A single ICPC-ish string, “ICPC”.
- Concatenation of four ICPC-ish strings, “I”, “C”, “P”, and “C”.

Problem F
Make a Loop
Time Limit: 2 seconds

Taro is playing with a set of toy rail tracks. All of the tracks are arc-shaped with a right central angle (an angle of 90 degrees), but with many different radii. He is trying to construct a single *loop* using all of them. Here, a set of tracks is said to form a single loop when both ends of all the tracks are smoothly joined to some other track, and all the tracks are connected to all the other tracks directly or indirectly. Please tell Taro whether he can ever achieve that or not.

Tracks may be joined in an arbitrary order, but their directions are restricted by the directions of adjacent tracks, as they must be joined smoothly. For example, if you place a track so that a train enters eastward and turns 90 degrees northward, you must place the next track so that the train enters northward and turns 90 degrees to either east or west (Figure F.1). Tracks may cross or even overlap each other as elevated construction is possible.

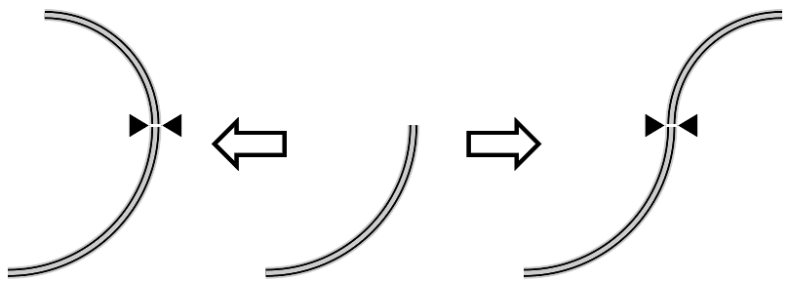


Figure F.1. Example of smoothly joining tracks

Input

The input consists of a single test case in the following format.

n
$r_1 \cdots r_n$

n represents the number of tracks, which is an even integer satisfying $4 \le n \le 100$. Each r_i represents the radius of the i-th track, which is an integer satisfying $1 \le r_i \le 10000$.

Output

If it is possible to construct a single loop using all of the tracks, print `Yes`; otherwise, print `No`.

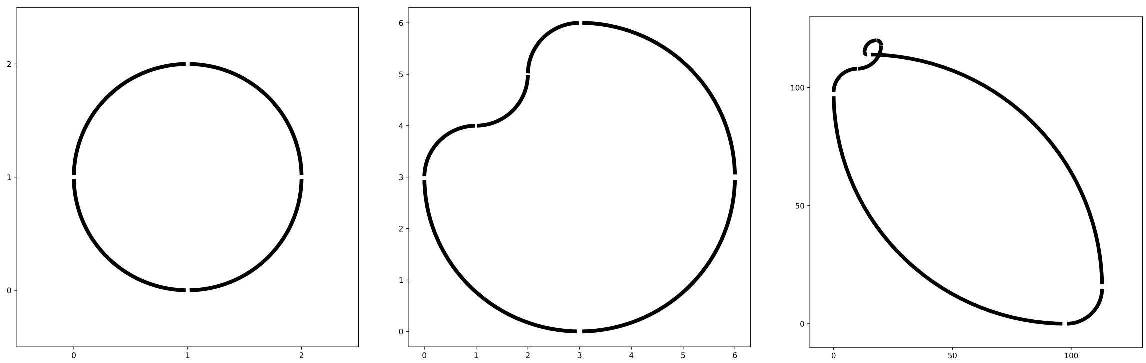
Sample Input 1	Sample Output 1
4 1 1 1 1	Yes

Sample Input 2	Sample Output 2
6 1 3 1 3 1 3	Yes

Sample Input 3	Sample Output 3
6 2 2 1 1 1 1	No

Sample Input 4	Sample Output 4
8 99 98 15 10 10 5 2 1	Yes

Possible loops for the sample inputs 1, 2, and 4 are depicted in the following figures.



(a) Sample 1 (b) Sample 2 (c) Sample 4

Figure F.2. Possible loops for the sample inputs

Problem G
Remodeling the Dungeon
Time Limit: 2 seconds

The Queen of Icpca resides in a castle peacefully. One day, she heard that many secret agents had been active in other nations, and got worried about the security of the castle.

The castle has a rectangular dungeon consisting of h rows of w square rooms. Adjacent rooms are separated by a wall. Some walls have doorways making the separated rooms intercommunicate. The entrance and the exit of the dungeon are in the rooms located at two diagonal extreme corners. The dungeon rooms form a *tree,* that is, the exit room is reachable from every room in the dungeon through a unique route that passes all the rooms on the route only once.

In order to enhance the security of the castle, the Queen wants to remodel the dungeon so that the number of rooms on the route from the entrance room to the exit room is maximized. She likes the tree property of the current dungeon, so it must be preserved. Due to the cost limitation, what can be done in the remodeling is to block one doorway to make it unavailable and to construct one new doorway instead on a wall (possibly on the same wall).

Your mission is to find a way to remodel the dungeon as the Queen wants.

Input

The input consists of a single test case in the following format.

$$
\begin{array}{l}
h\ w \\
c_{1,1}c_{1,2}\cdots c_{1,2w+1} \\
c_{2,1}c_{2,2}\cdots c_{2,2w+1} \\
\vdots \\
c_{2h+1,1}c_{2h+1,2}\cdots c_{2h+1,2w+1}
\end{array}
$$

h and w represent the size of the dungeon, each of which is an integer between 2 and 500, inclusive. The characters $c_{i,j}$ $(1 \le i \le 2h+1,\ 1 \le j \le 2w+1)$ represent the configuration of the dungeon as follows:

- $c_{2i,2j}$ = '`.`' for $1 \le i \le h$ and $1 \le j \le w$, which corresponds to the room i-th from the north end and j-th from the west end of the dungeon; such a room is called the room (i, j).
- $c_{2i+1,2j}$ = '`.`' or '`-`' for $1 \le i \le h-1$ and $1 \le j \le w$, which represents the wall between the rooms (i, j) and $(i+1, j)$; the character '`.`' means that the wall has a doorway and '`-`' means it has no doorway.

- $c_{2i,2j+1}$ = '`.`' or '`|`' for $1 \le i \le h$ and $1 \le j \le w-1$, which represents the wall between the rooms (i,j) and $(i,j+1)$; the character '`.`' means that the wall has a doorway and '`|`' means it has no doorway.
- $c_{1,2j} = c_{2h+1,2j}$ = '`-`' $(1 \le j \le w)$ and $c_{2i,1} = c_{2i,2w+1}$ = '`|`' $(1 \le i \le h)$, which correspond to the outer walls of the dungeon.
- $c_{2i+1,2j+1}$ = '`+`' for $0 \le i \le h$ and $0 \le j \le w$, which corresponds to an intersection of walls or outer walls.

The entrance and the exit of the dungeon are in the rooms $(1,1)$ and (h,w), respectively. The configuration satisfies the tree property stated above.

Output

Output the maximum length of the route from the entrance room to the exit room achievable by the remodeling, where the length of a route is the number of rooms passed including both the entrance and exit rooms.

Sample Input 1

```
2 3
+-+-+-+
|.....|
+.+.+.+
|.|.|.|
+-+-+-+
```

Sample Output 1

```
6
```

Sample Input 2

```
2 3
+-+-+-+
|...|.|
+.+.+.+
|.|...|
+-+-+-+
```

Sample Output 2

```
4
```

Sample Input 3	**Sample Output 3**

`5 5`
`+-+-+-+-+-+`
`|...|...|.|`
`+-+.+.+.+.+`
`|...|.|.|.|`
`+.+.+.+-+.+`
`|.|...|.|.|`
`+.+.+-+.+.+`
`|.|.....|.|`
`+-+.+.+-+.+`
`|...|.....|`
`+-+-+-+-+-+`
	`15`

In the first sample, if you block the doorway between the rooms $(1,1)$ and $(1,2)$ and construct a new doorway between the rooms $(2,1)$ and $(2,2)$, then the unique route from $(1,1)$ to $(2,3)$ passes all of the 6 rooms, which is obviously the maximum.

In the second sample, any remodeling keeps the length of the unique route from $(1,1)$ to $(2,3)$ to be exactly 4.

In the third sample, one of the optimal ways is blocking the doorway between the rooms $(4,2)$ and $(4,3)$ and constructing a new doorway between the rooms $(2,4)$ and $(3,4)$.

The configurations after the remodeling described above are as follows.

```
+-+-+-+    +-+-+-+    +-+-+-+-+-+
|.|...|    |...|.|    |...|...|.|
+.+.+.+    +.+.+.+    +-+.+.+.+.+
|...|.|    |.|...|    |...|.|.|.|
+-+-+-+    +-+-+-+    +.+.+.+.+.+
                      |.|...|.|.|
                      +.+.+-+.+.+
                      |.|.|...|.|
                      +-+.+.+-+.+
                      |...|.....|
                      +-+-+-+-+-+
```

Problem H
Cake Decoration
Time Limit: 10 seconds

You are ordering a cake to celebrate the New Year. You have to decide the numbers of decoration items to be topped on the cake. Available items are dog figures, cat figures, red candies, and blue candies.

You want all four items to decorate the cake, and all the numbers of the four items to be different from each other. You also want the number of figures (the sum of dogs and cats) to be within a certain range.

An extra charge for the decoration items is added to the price of the cake. The extra is, although quite queer, the product of the numbers of the four decoration items. You want the cake to look gorgeous as far as the budget allows. Thus, you are not satisfied with a decoration if you can add any of the four items without violating the budget constraint.

The conditions stated above are summarized as follows. Let d, c, r, and b be the numbers of dog figures, cat figures, red candies, and blue candies, respectively. All these numbers should be different positive integers satisfying the following conditions for given X, L, and R:

- $L \le d + c < R$,
- $d \times c \times r \times b \le X$,
- $(d+1) \times c \times r \times b > X$,
- $d \times (c+1) \times r \times b > X$,
- $d \times c \times (r+1) \times b > X$, and
- $d \times c \times r \times (b+1) > X$.

More than one combination of four numbers of decoration items may satisfy the conditions. Your task is to find how many such combinations exist.

Input

The input consists of a single test case of the following format.

X L R

Here, X, L, and R are integers appearing in the conditions stated above. They satisfy $1 \le X \le 10^{14}$ and $1 \le L < R \le 10^{14}$.

Output

Output the number of combinations of four numbers of decoration items that satisfy the conditions stated above modulo a prime number $998\,244\,353 = 2^{23} \times 7 \times 17 + 1$.

Sample Input 1	Sample Output 1
`24 4 6`	`12`

Sample Input 2	Sample Output 2
`30 5 6`	`4`

Sample Input 3	Sample Output 3
`30 9 20`	`0`

Sample Input 4	Sample Output 4
`100000000000000 1 100000000000000`	`288287412`

For Sample Input 2, four combinations of $(d, c, r, b) = (2, 3, 1, 5)$, $(2, 3, 5, 1)$, $(3, 2, 1, 5)$, and $(3, 2, 5, 1)$ satisfy all the conditions. $(d, c, r, b) = (1, 4, 2, 3)$ is not eligible because its extra cost for the decoration items does not exceed $X = 30$ even after adding one more cat figure. $(d, c, r, b) = (1, 5, 2, 3)$ is also ineligible because $d + c < R = 6$ does not hold.

Problem I
Quiz Contest
Time Limit: 8 seconds

The final round of the annual World Quiz Contest is now at the climax!

In this round, questions are asked one by one, and the competitor correctly answering the goal number of questions first will be the champion. Many questions have already been asked and answered. The numbers of questions correctly answered so far by competitors may differ, and thus the numbers of additional questions to answer correctly to win may be different.

The questions elaborated by the judge crew are quite difficult, and the competitors have completely distinct areas of expertise. Thus, for each question, exactly one competitor can find the correct answer.

Who becomes the champion depends on the order of the questions asked. The judge crew know all the questions and who can answer which, but they do not know the order of the remaining questions, as the questions have been randomly shuffled. To help the judge crew guess the champion of this year, count the number of possible orders of the remaining questions that would make each competitor win. Note that the orders of the questions left unused after the decision of the champion should also be considered.

Input

The input consists of a single test case of the following format.

$$n\ m$$
$$a_1\ \cdots\ a_n$$
$$b_1\ \cdots\ b_n$$

Here, n is the number of competitors and m is the number of remaining questions. Both n and m are integers satisfying $1 \le n \le m \le 2 \times 10^5$. Competitors are numbered 1 through n. The following line contains n integers, $a_1, \ldots, a_n$, meaning that the number of remaining questions that the competitor i can answer correctly is a_i, where $\sum_{i=1}^{n} a_i = m$ holds. The last line contains n integers, $b_1, \ldots, b_n$, meaning that the competitor i has to answer b_i more questions correctly to win. $1 \le b_i \le a_i$ holds.

Output

Let c_i be the number of question orders that make the competitor i the champion. Output n lines, each containing an integer. The number on the i-th line should be c_i modulo a prime number $998\,244\,353 = 2^{23} \times 7 \times 17 + 1$. Note that $\sum_{i=1}^{n} c_i = m!$ holds.

Sample Input 1	**Sample Output 1**
2 4 2 2 1 2	20 4

Sample Input 2	**Sample Output 2**
4 6 1 1 2 2 1 1 1 2	168 168 336 48

Sample Input 3	**Sample Output 3**
4 82 20 22 12 28 20 22 7 8	746371221 528486621 148054814 913602744

Problem J
Traveling Salesperson in an Island
Time Limit: 2 seconds

You are a salesperson at one of the ports in an island. You have to visit all the ports of the island and then come back to the starting port. Because you cannot swim and are scared of the sea, you have to stay on the land during your journey.

The island is modeled as a polygon on a two-dimensional plane. The polygon is *simple,* that is, its vertices are distinct and no two edges intersect or touch, other than consecutive edges which touch at their common vertex. In addition, no two consecutive edges are collinear. Each port in the island is modeled as a point on the boundary of the polygon. Your route is modeled as a closed curve that does not go outside of the polygon.

In preparation for the journey, you would like to compute the minimum length of a route to visit all the ports and return to the starting port.

Input

The input consists of a single test case of the following format.

$$
\begin{array}{l}
n\ m \\
x_1\ y_1 \\
\vdots \\
x_n\ y_n \\
x'_1\ y'_1 \\
\vdots \\
x'_m\ y'_m
\end{array}
$$

The first line contains two integers n and m, which satisfy $3 \le n \le 100$ and $2 \le m \le 100$. Here, n is the number of vertices of the polygon modeling the island, and m is the number of the ports in the island. Each of the next n lines consists of two integers x_i and y_i, which are the coordinates of the i-th vertex of the polygon, where $0 \le x_i \le 1000$ and $0 \le y_i \le 1000$. The vertices of the polygon are given in counterclockwise order. Each of the m following lines consists of two integers x'_j and y'_j, which are the coordinates of the j-th port. The route starts and ends at (x'_1, y'_1). It is guaranteed that all the ports are on the boundary of the polygon and pairwise distinct.

Output

Output in a line the minimum length of a route to visit all the ports and return to the starting port. The relative error of the output must be within 10^{-6}.

Sample Input 1	**Sample Output 1**
4 4 0 0 2 0 2 2 0 2 1 0 1 2 2 1 0 1	5.656854249492381

Sample Input 2	**Sample Output 2**
8 2 0 0 4 0 4 4 0 4 0 3 3 3 3 1 0 1 0 0 0 4	16.64911064067352

These samples are depicted in the following figures. The shortest routes are depicted by the thick lines. The gray polygons represent the islands. The small disks represent the ports in the islands. Note that the route does not have to be simple, i.e., the route may intersect or overlap itself as in the second sample, in which the same path between the two ports is used twice.

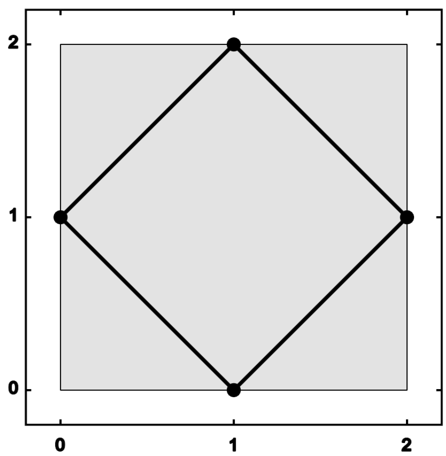


Figure J.1. Sample 1

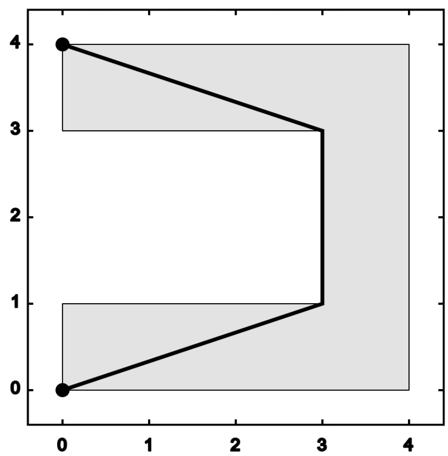


Figure J.2. Sample 2

Problem K
New Year Festival
Time Limit: 7 seconds

The *ICPC (Incredibly Colossal and Particularly Comfortable) Theater* is giving a number of traditional events to celebrate the New Year!

Each of the events has its own unalterable duration. Start times of the events are flexible as long as no two events overlap. An event may start immediately after another event ends.

Start times of the events influence the costs. The cost of an event is given by a continuous piecewise linear function (a polygonal line function) of its start time. Different events may have different cost functions.

You are asked to schedule all the events minimizing the total cost.

Input

The input consists of a single test case of the following format.

n
Description$_1$
$\vdots$
Description$_n$

The first line contains an integer n, representing the number of events. $2 \le n \le 11$ holds. The descriptions of the events follow. The description of the i-th event, Description$_i$ $(1 \le i \le n)$, has the following format.

m l
x_1 y_1
$\vdots$
x_m y_m

The integer m in the first line is the number of vertices of the cost function of the event. The integer l in the same line is the duration of the event. $1 \le m \le 60$ and $1 \le l \le 10^8$ hold.

The following m lines describe the cost function. The j-th line of the m lines consists of the two integers x_j and y_j specifying the j-th vertex of the cost function. $0 \le x_1 < \cdots < x_m \le 10^8$ and $0 \le y_j \le 10^8$ hold. In addition, $(y_{j+1} - y_j)/(x_{j+1} - x_j)$ is an integer for any $1 \le j < m$.

The start time t of the event must satisfy $x_1 \le t \le x_m$. For j $(1 \le j < m)$ satisfying $x_j \le t \le x_{j+1}$, the cost of the event is given as $y_j + (t - x_j) \times (y_{j+1} - y_j)/(x_{j+1} - x_j)$.

The total number of the vertices of all the cost functions is not greater than 60.

Output

Output the minimum possible total cost of the events in a line.

It is guaranteed that there is at least one possible schedule containing no overlap. It can be proved that the answer is an integer.

Sample Input 1

```
3
3 50
300 2500
350 0
400 3000
2 120
380 0
400 2400
4 160
0 800
400 0
450 100
950 4600
```

Sample Output 1

```
1460
```

Sample Input 2

```
4
2 160
384 0
1000 2464
3 280
0 2646
441 0
1000 2795
1 160
544 0
2 240
720 0
1220 2000
```

Sample Output 2

```
2022
```

For Sample Input 1, making the (start time, end time) pairs of the three events to be (330, 380), (380, 500), and (170, 330), respectively, achieves the minimum total cost without event overlaps. The cost of the event 1 is $2500 + (330 - 300) \times (0 - 2500)/(350 - 300) = 1000$. Similarly, the costs of the events 2 and 3 are 0 and 460, respectively.

For Sample Input 2, the minimum cost is achieved by (384, 544), (104, 384), (544, 704), and (720, 960) for the four events.

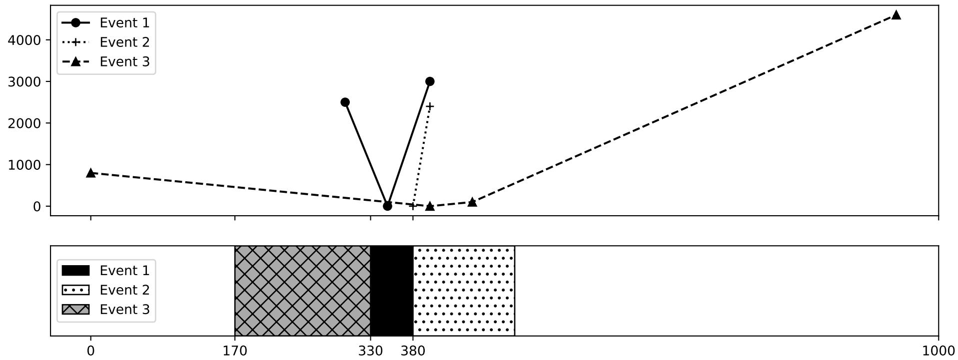


Figure K.1. Cost functions in Sample Input 1 and a sample schedule

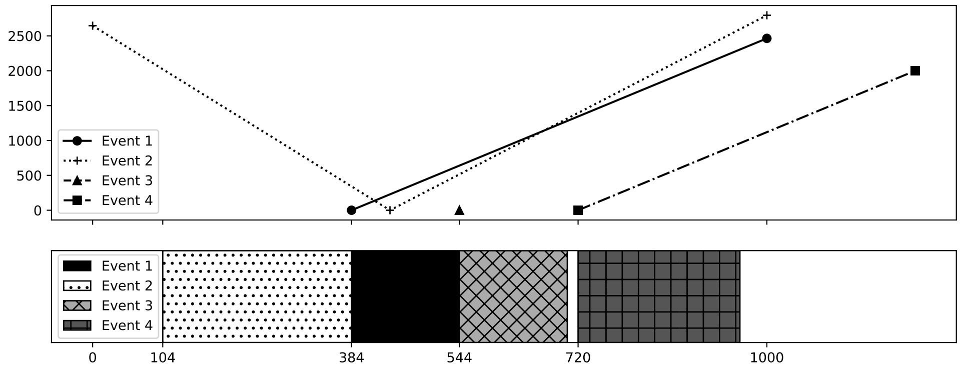


Figure K.2. Cost functions in Sample Input 2 and a sample schedule

In Figures K.1 and K.2, polylines in the top figure represent cost functions, and rectangles in the bottom figure represent event durations of a schedule achieving the minimum total cost.