ICPC SOUTH PACIFIC REGIONALS (DIVISION 1)

JANUARY 23, 2021

Contest Problems

A: Allocating Shares
B: Bartering Stonks
C: Card Folding
D: Dank Invader
E: Encrypting Intervals
F: Flip Tool
G: Golden Grove
H: Hedge Maze
I : Iguana Date
J : Judge's Dogged Divination
K: Keeping Social Distance
L: Legendary Walking Trails

This contest contains twelve problems over 26 pages. Good luck.

For problems that state "*Your answer should have an absolute or relative error of less than* 10^{-9}", your answer, x, will be compared to the correct answer, y. If $|x - y| < 10^{-9}$ or $\frac{|x-y|}{|y|} < 10^{-9}$, then your answer will be considered correct.

Definition 1

For problems that ask for a result modulo m:
If the correct answer to the problem is the integer b, then you should display the unique value a such that:

- $0 \leq a < m$ and
  and
- $(a - b)$ is a multiple of m.

Definition 2

A string $s_1 s_2 \cdots s_n$ is lexicographically smaller than $t_1 t_2 \cdots t_\ell$ if

- there exists $k \leq \min(n, \ell)$ such that $s_i = t_i$ for all $1 \leq i < k$ and $s_k < t_k$
  or
- $s_i = t_i$ for all $1 \leq i \leq \min(n, \ell)$ and $n < \ell$.

Definition 3

- Uppercase letters are the uppercase English letters $(A, B, \ldots, Z)$.
- Lowercase letters are the lowercase English letters $(a, b, \ldots, z)$.

Definition 4

Unless otherwise specified, the distance between two points (x_0, y_0) and (x_1, y_1) is defined as its Euclidean distance:

$$\sqrt{(x_0 - x_1)^2 + (y_0 - y_1)^2}.$$

Definition 5

A positive number is a number that is strictly greater than 0, while a negative number is strictly less than 0. This means that zero is neither negative nor positive. A number is nonpositive if it is not positive and a number is nonnegative if it is not negative.

Definition 6

The expected value is the arithmetic mean of doing a procedure many, many times (think 10^{10000} times). For example, imagine flipping a coin 5 times, and you want to count the number of times it lands on tails. When you flip the coin 5 times, sometimes you get 0 tails, sometimes you get 1 tails, and so on. But after many trials, the arithmetic mean will be 2.5 tails. Thus, the expected number of tails is 2.5.

Problem A
Allocating Shares
Time limit: 4 seconds

The newest huge maths company, South Pacific Computation Corporation (SPPC), is here! SPPC has shares that they need to distribute to their employees. As their CEO, you will receive your shares over n days. On day 1, you receive n shares. On day two, you receive $n/2$ shares (rounded down). On day three, you receive $n/3$ shares (rounded down). On day i, you receive n/i shares (rounded down). On the final day (day n), you receive $n/n = 1$ share. For example, if $n = 3$, then you would receive $3+1+1 = 5$ shares.

How many shares in total do you receive?

Input

The input consists of a single line containing one integer n ($1 \leq n \leq 10^{12}$), which is the number of days over which you receive your shares.

Output

Display the number of shares in total that you receive.

Sample Input 1	Sample Output 1
1	1

Sample Input 2	Sample Output 2
2	3

Sample Input 3	Sample Output 3
3	5

Sample Input 4	Sample Output 4
5	10

This page is intentionally left (almost) blank.

Problem B
Bartering Stonks
Time limit: 25 seconds

You are trying to make money trading *stonks*. There is only a single type of stonk that you are willing to trade. You start with your life savings, an integer amount of cash C which you will use to trade stonks for a period of N days.

On each day, the stonk will have an integer price of P_i dollars. Every day, you do exactly one of three things:

1. Sell some positive integer number of stonks
2. Buy some positive integer number of stonks
3. Nothing

However, at no point in time can you have more stonks than the capacity of your special stonk bag.

Every time you buy or sell stonks, you must pay tributes. The tributes paid is a certain percentage of the total price of the stonks being bought or sold on each day. For example, if today's percentage fee is 7% and you would like to buy 10 stonks for $12 each, then you must pay $8.40 in tributes (7% of $120).

Note that in order to sell or buy stonks, you first need available cash to pay the tributes. Your cash cannot be negative at any point. Selling x stonks in a single day requires paying the tributes for all of them before you receive any of the cash from selling.

What is the maximum total cash you can have at the end of N days?

Input

The first line of the input contains three integers N ($1 \leq N \leq 50$), which is the number of days in the scenario, C ($0 \leq C \leq 100\,000$), which is your initial starting cash in dollars, and K ($0 \leq K \leq 100\,000$), which is the maximum capacity of your stonk bag.

The next N lines describe the price and tribute fee on each day. Each of these lines contain two integers P ($1 \leq P \leq 100\,000$), which is the price of the stonk in dollars on this day, and F ($1 \leq F \leq 99$), which is this day's percentage rate for computing the tributes.

Output

Display the maximum cash in cents you could have after the final day. There are exactly 100 cents in a dollar.

Sample Input 1

```
4 100 1000
50 3
40 3
60 3
70 3
```

Sample Output 1

```
15340
```

Sample Input 2

```
4 5 1
5 25
4 25
6 25
7 25
```

Sample Output 2

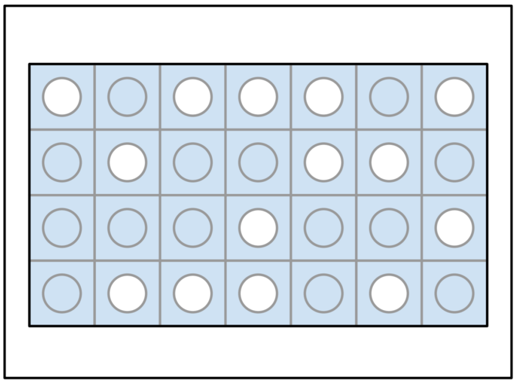
```
500
```

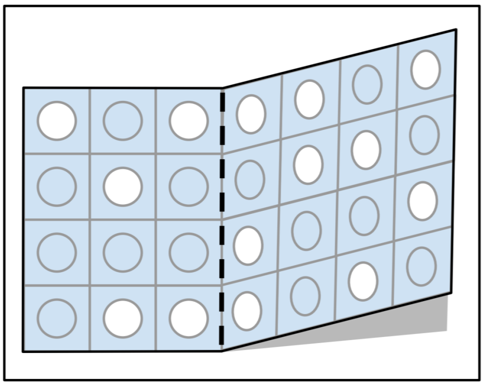
This page is intentionally left (almost) blank.

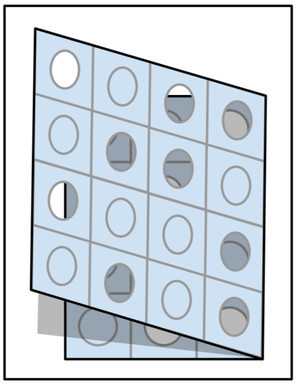
Problem C
Card Folding
Time limit: 2 seconds

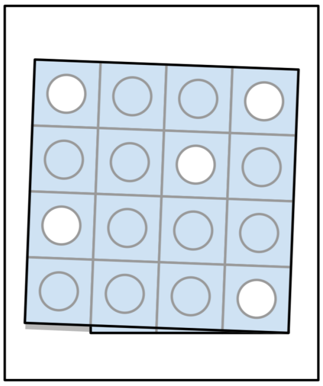
In the prehistoric era of computing, programs were represented as *punch cards*. A punch card is an $R \times C$ grid of cells. A cell has a tiny piece of paper that can be removed to make a hole. Each cell of the grid is either a hole or not.

Savine is trying to program a punch card machine. Unfortunately, she does not have the ability to punch any new holes. Luckily, she already has a punch card with holes punched in it. Her plan is to fold the punch card vertically between two columns of the grid in such a way that the rightmost column is folded to the left. After folding, some of the cells of the grid will line up. For example, say Savine has a punch card with 7 columns. If she folds between columns 3 and 4, then the cells in columns 3 and 4 will line up, the cells in columns 2 and 5 will line up, the cells in columns 1 and 6 will line up, and the cells in column 7 will overhang to the left of column 1, not lining up with any cells in the original punchcard. This new punchcard has 4 columns (from left to right: the 7 column, the (1,6) column, the (2,6) column, and the (3,4) column). There is a hole in this new punchcard only if both of the lined up cells were holes in the original punchcard.


(1)


(2)


(3)


(4)

Given the original punchcard and the column where she is planning on doing the fold, what will the new punchcard look like?

Input

The first line of input contains three integers R ($1 \le R \le 50\,000$), which is the number of rows, C ($2 \le C \le 10^5$ and $R \times C \le 10^5$), which is the number of columns, and Y ($1 \le Y \le C - 1$), which indicates that Savine plans to fold the punch between column Y and column $Y + 1$.

The next R lines describe the punch card. Each of these lines contains a string of length C containing only the characters '`o`' and '`x`'. An '`o`' represents a cell with a hole and an '`x`' represents a cell without a hole.

Output

Display the folded punch card.

Sample Input 1

```
4 7 3
oxooooxo
xoxxoox
xxxoxxo
xoooxox
```

Sample Output 1

```
oxxo
xxox
oxxx
xxxo
```

Sample Input 2

```
3 3 1
oxx
xox
xxo
```

Sample Output 2

```
xx
xx
ox
```

Sample Input 3

```
3 3 2
oxx
xox
xxo
```

Sample Output 3

```
ox
xx
xx
```

Sample Input 4

```
2 3 1
oox
xoo
```

Sample Output 4

```
xo
ox
```

Problem D
Dank Invader
Time limit: 20 seconds

D-Unit the Dank Invader has landed on a peculiar planet. This planet has a series of platforms going from left to right of varying heights. D-Unit must visit a sequence of platforms from left to right, with strictly increasing height.

After D-Unit has finished, they look at the set of platforms they visited. If there are no platforms that D-Unit could have added to the visited platforms that would have maintained the strictly increasing requirement, then D-Unit successfully completed a *dank walk*.

For example, if the platforms are of height $[11, 12, 13, 15, 14]$, and D-Unit visited $12, 13, 14$, then this is not a dank walk since they could have also visited the platform of height 11. Similarly, $11, 13, 14$ is not a dank walk because they could have visited the platform of height 12. Visiting $11, 12, 13, 14$ is a dank walk (even though the platform of height 15 is higher than 14, D-Unit's path would not be left to right).

What is the minimum number of platforms that D-Unit can visit among all dank walks?

Input

The first line of input contains a single integer N ($1 \leq N \leq 10\,000$), which is the number of platforms.

The second line contains N integers, H_i ($1 \leq H_i \leq 10^9$), each denoting the height of the i-th platform.

Output

Display the minimum number of platforms that D-Unit can visit among all dank walks.

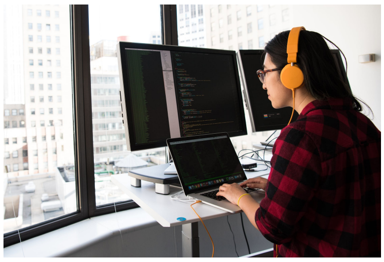
Sample Input 1	Sample Output 1
3 1 3 2	2

Sample Input 2	Sample Output 2
5 2 5 1 2 3	2

Sample Input 3	Sample Output 3
6 1 2 3 4 2 5	3

This page is intentionally left (almost) blank.

Problem E
Encrypting Intervals
Time limit: 8 seconds



Gabriella has just made an incredible breakthrough in the world of encryption! Her new encryption algorithm allows communication between two people. Most of the details of this new encryption scheme are incomprehensible, but the crucial part of the encryption algorithm involves two intervals of positive integers.

The strength of the encryption algorithm is equal to the number of pairs of positive integers (one from each interval) that are relatively prime (that is, no positive integer other than 1 divides both numbers).

For example, if the two intervals are $[1, 2, 3, 4, 5]$ and $[3, 4, 5, 6]$, then the pairs (1,3), (1,4), (1,5), (1,6), (2,3), (2,5), (3,4), (3,5), (4,3), (4,5), (5,3), (5,4), (5,6) are all relatively prime. Thus, the strength is 13.

Given the two positive intervals, what is the strength of the encryption algorithm?

Input

The first line of input consists of two integers, a and b ($1 \le a \le b \le 10^{12}$ and $b - a \le 10^6$), which are the two (inclusive) endpoints of the first interval.

The second line of input consists of two integers, c and d ($1 \le c \le d \le 10^{12}$ and $d - c \le 10^6$), which are the two (inclusive) endpoints of the second interval.

Output

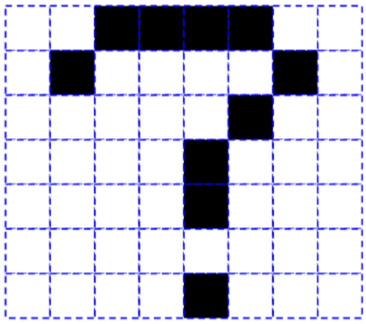
Display the strength of the encryption algorithm.

Sample Input 1	Sample Output 1
1 5 3 6	13

Sample Input 2	Sample Output 2
10 50 51 100	1232

This page is intentionally left (almost) blank.

Problem F
Flip Tool
Time limit: 15 seconds



Orso is using some simple software to create 2D images. Unfortunately, the software has stopped working correctly. Only the flip tool is working. Orso has already drawn some pixels black and some pixels white (every pixel is either black or white). He wants to get back to a state where every pixel is the same colour.

The way the flip tool works is by first having the user select a pixel. After this pixel is selected, the flip tool finds the largest 8-connected region of pixels all of the same colour containing the selected pixel, then flips the colour of each pixel in that region (from white to black or from black to white). In other words, the flip tool starts at the selected pixel, then selects each adjacent pixel that shares an edge or corner with it, and which has the same colour as it, then it selects each of their neighbours of the same colour, and so on. All the pixels selected this way have their colour flipped.

What is the fewest number of flip tool applications required to get to a state where every pixel is the same colour?

Input

The first line of input contains two integers R ($1 \leq R \leq 1\,000$) and C ($1 \leq C \leq 1\,000$), which are the number of rows and columns of pixels in the image respectively.

The next R lines describe the image. Each of these lines contains a string of length C containing only the characters '`.`' and '`x`'. A '`.`' represents a white pixel and a '`x`' represents a black pixel.

Output

Display the minimum number of flip tool operations required.

Sample Input 1	Sample Output 1

`6 6`
`.xx...`
`xxx...`
`....xx`
`......`
`xxxxxx`
`......`
	`2`

Sample Input 2	Sample Output 2

`2 8`
`..xx.xx.`
`x..xxx.x`
	`2`

Sample Input 3	Sample Output 3
7 8 ..xxxx.. .x....x. .....x.. ....x... ....x... ....x...	1

Problem G
Golden Grove
Time limit: 4 seconds

Gwen is the governor of Golden Grove. She has decided to reward her citizens with coins based on their productivity. All n citizens have a productivity value. Gwen numbers the citizens from 1 to n. Their productivity numbers are $p_1, p_2, \ldots, p_n$. Person 1 receives no coins. For everyone else, Gwen compares p_i and p_{i-1}. If $p_i \geq p_{i-1}$, then person i receives $p_i - p_{i-1}$ gold coins. If $p_i < p_{i-1}$, then person i receives $p_{i-1} - p_i$ silver coins.

Gwen must buy enough coins to give out to all citizens. She will buy exactly k gold coins and k silver coins (for some value of k). This means that if more gold coins are needed, then there will be wasted silver coins. Similarly, if more silver coins are needed, there will be wasted gold coins. Gwen wants to waste as few coins as possible (she does not care if the wasted coins are gold or silver). She notices that the order in which she numbers the citizens affects the number of wasted coins.

For example, if the citizens have productivity values of $\{11, 15, 17, 22\}$, and Gwen numbers them such that $p_1 = 15, p_2 = 22, p_3 = 17, p_4 = 11$, then person 1 receives no coins, person 2 receives seven gold coins $(22-15)$, person 3 receives five silver coins $(22-17)$, and person 4 receives six silver coins $(17-11)$. In total, Gwen needs to give out seven gold coins and eleven silver coins, so she must purchase $k = 11$ gold and silver coins. This wastes four gold coins. However, if Gwen arranges them such that $p_1 = 15, p_2 = 22, p_3 = 11, p_4 = 17$, then a total of 13 golden coins and 11 silver coins are needed, so there would only be two wasted coins (with $k = 13$).

Given the productivity values of all citizens, determine the minimum number of wasted coins among all possible numberings of citizens.

Input

The first line of input contains a single integer n $(1 \leq n \leq 200\,000)$, which is the number of citizens in Golden Grove.

The next n lines describe the citizens' productivity. Each of these lines contains a single integer p $(1 \leq p \leq 10^9)$, which is the productivity value of this citizen.

Output

Display the minimum number of wasted coins among all possible numberings of citizens.

Sample Input 1	Sample Output 1
4 11 15 17 22	2

Sample Input 2	Sample Output 2
3 2 1 2	0

Sample Input 3	Sample Output 3
2 1 1000000	999999

Problem H
Hedge Maze
Time limit: 5 seconds

In the old but popular game, RollerCoaster Tycoon, one of the attractions you can add to your park is a Hedge Maze. In this attraction, guests enter the maze and walk around until they reach the exit.

You are designing a new, special variant of the Hedge Maze. Your maze is modelled as an $n \times n$ grid where some of the cells are blocked off with hedges and other cells are cleared out. Guests can only occupy cells that are cleared out. Every minute, a guest will move using the following strategy:

First, the guest chooses a direction (either north, south, east, or west) uniformly at random. They will then check the four adjacent cells in clockwise order starting in that direction. For example, if they chose south as the direction, they will check the adjacent south cell, then the west cell, then the north cell, and finally the east cell. The guest will move to the first empty cell that they check. If all four directions are blocked off, then the guest will remain in the same cell.

Note that more than one guest can occupy the same cell in the grid at the same time.

You have created a maze for your guests. Also, all of your guests have arrived and each is at some cell in the grid. What is the expected number of guests in each cell after t minutes?

Input

The first line of input contains two integers n ($1 \leq n \leq 8$), which is the length of one side of the grid, and t ($1 \leq t \leq 10^{18}$), which is the number of minutes the guests will walk.

The next n lines describe the grid. Each of these lines contains a string of length n containing only the characters '`.`', '$\star$', and '`#`'. A '`.`' represents a cell that is cleared out and which guests can travel through. A '$\star$' represents a cell that is also cleared out but a guest currently is standing at. Finally, a '`#`' represents a cell blocked off with hedges.

Output

Display the expected number of guests in every cell of the maze after t minutes have elapsed.

It can be proven that the expected number of guests in every cell of the maze after t minutes can be expressed in the form $x/4^t$, where x is an integer. For each cell, display x modulo the prime 1 000 000 007.

Sample Input 1	Sample Output 1
3 5 #.# .*. #.#	0 256 0 256 0 256 0 256 0

Sample Input 2	Sample Output 2
3 1 #.. .*# .#.	0 1 0 3 0 0 0 0 0

Sample Input 3	Sample Output 3
3 2 #.. .*# .#*	0 0 3 0 10 0 3 0 16

This page is intentionally left (almost) blank.

Problem I
Iguana Date
Time limit: 2 seconds

Iggy the Iguana is back! This time, he brought his date, Ignia, with him to the corn maze. The corn maze is a square grid where some of the cells are blocked off with impassable corn plants and others are cleared out. Iggy and Ignia can only travel in and through cells that are cleared out. They can move to a cell in any of the four cardinal directions (north, south, east, and west).

The corn maze they chose for their date is special in that every cell that is cleared out can be reached from the start (the top-left of the maze) through exactly one direct path (without revisiting cells).

Iggy and Ignia start in the top-left cell. They stay in that cell for one minute, then move to an adjacent cell. Every minute Iggy and Ignia will move to an adjacent cell. When they reach the bottom-right cell, they stay there for one minute and then leave the maze and their date ends. Both of the iguanas are having a wonderful time and do not want the fun times to end. Therefore, they will choose a path through the maze (possibly revisiting cells) that makes their date last as long as possible. However, Iggy and Ignia do not want to get lost forever, so they will never enter a cell from the same direction more than once.

How long can Iggy and Ignia make their date last?

Input

The first line of input contains a single integer n ($2 \leq n \leq 100$), which is the length of each side of the square grid representing the maze.

The next n lines describe the grid. Each of these lines contains a string of length n containing only the characters '`.`' and '`#`'. A '`.`' represents a cell that is cleared out and which Iggy and Ignia can travel through, and a '`#`' represents a cell blocked off with corn plants.

Output

Display the length of Iggy and Ignia's date.

Sample Input 1
```
3
...
.#.
#..
```

Sample Output 1
```
7
```

Sample Input 2
```
4
..#.
#...
..##
#...
```

Sample Output 2
```
15
```

Sample Input 3
```
4
..##
#.#.
....
#.#.
```

Sample Output 3
```
13
```

This page is intentionally left (almost) blank.

Problem J
Judge's Dogged Divination
Time limit: 2 seconds

Today is the day of the South Paddock Programming Contest (SPPC) down on the farm. The judges are the farm's working dogs who have competed online for years behind the anonymity of the internet, and the contestants are individual sheep and goats. As in recent years, the judges are also running a contest amongst themselves for the program that best predicts the SPPC results. The chief judge, Erica, has been so busy with the problem set for the main event that she has not yet had time to implement her prediction program, so she needs your help with this.

Erica's approach is based on her knowledge of both the problem set and the contestants. For each problem, she has estimated how long it would take her to solve it. For each contestant, she has assigned a multiplier. She estimates that the time it will take each contestant to solve each problem is her solving time multiplied by that contestant's multiplier. When it comes to the matter of who tackles which problem when, Erica assumes that each contestant will spend their first 10 minutes determining their preference order of the problems. After the 10 minutes, each contestant will use either a sheep strategy or a goat strategy. It is their strategy that "separate the sheep from the goats".

Contestants following the goat strategy stubbornly do their own thing, ignoring the scoreboard. They work through the problems one-by-one in their preference order until either the contest ends or they finish all problems.

Contestants following the sheep strategy follow the scoreboard. Whenever they are not working on a problem, they wait (perhaps zero time) until the scoreboard shows one or more problems solved that they have not yet solved, at which point they pick their most preferred such problem and work on it (even if they do not have enough time to finish this problem and there is enough time to finish another). Note that if no problem is solved, a contestant following the sheep strategy would wait for some problem to be solved before working on a problem.

There are a few timing details that the program needs to handle. The contest starts at time 0. The contest lasts T minutes and it is not possible to submit a solution at or after time T. A solution that is submitted at time t, will first appear on the scoreboard (as solved) at time $t + 1$. (The scoreboard is not frozen during the contest.)

Please implement Erica's predictions so that she can try to win the inter-judge event.

Input

The first line of input contains three integers T ($1 \leq T \leq 1\,000$), which is the length of the contest in minutes, P ($1 \leq P \leq 10$), which is the number of problems (which are numbered 1 to P), and C ($1 \leq C \leq 100$), which is the number of contestants.

The next line contains P integers, one per problem, in ascending problem number order. Each of these integers S ($1 \leq S \leq 30$) is Erica's estimate of her solving time in minutes for this problem.

The remaining C lines describe the contestants, one per line. Each contestant line contains $P + 2$ items: a string "`baa`" (for a sheep) or a string "`maa`" (for a goat), an integer M ($1 \leq M \leq 100$), which is Erica's multiplier for this contestant, then a permutation of the integers 1 to P, which is this contestant's preference order, giving the problem numbers from most preferred first to least preferred last.

Output

For each contestant, in the order that they appear in the input, display their result on a separate line showing their performance on each problem, in ascending problem number order, as either the time in minutes at which they submitted their solution or 0 for a problem that they did not submit.

Sample Input 1

```
300 3 3
5 20 5
maa 15 1 2 3
maa 15 3 2 1
baa 20 2 3 1
```

Sample Output 1

```
85 0 0
0 0 85
286 0 186
```

Sample Input 2

```
300 3 3
5 20 5
maa 15 1 2 3
baa 1 1 2 3
baa 1 3 2 1
```

Sample Output 2

```
85 0 0
91 0 0
91 0 0
```

Problem K
Keeping Social Distance
Time limit: 2 seconds

To stay safe, parks and other areas have put social distancing rules in place. You have been people-watching from a distance at a park and want to calculate how well people are social distancing. To do so, you will calculate the "Social Distance Factor" of the people at the park.

The Social Distance Factor is defined as the minimum distance between any two people that are not in the same group. Each person at the park belongs to exactly one group and each group can contain either one or two people.

For example, say there are 3 people at the park, Alice, Bart, and Cathy and their locations are $(0, 0)$, $(3, 2)$, and $(4, 1)$, respectively. If all three people were in groups by themselves, the Social Distance Factor would be $\sqrt{2}$ because the distance between Bart and Cathy is $\sqrt{(3-4)^2 + (2-1)^2} = \sqrt{2}$. But, if Bart and Cathy were in a group together, the Social Distance Factor would be $\sqrt{13}$ because the distance between Alice and Bart is $\sqrt{(0-3)^2 + (0-2)^2} = \sqrt{13}$. Note that although Bart and Cathy are closer together, this distance does not count because they are in a group together.

You do not know which people are in groups together. So, you have decided to consider all of the possible ways that the people in the park could be in groups. What is the maximum Social Distance Factor among all possible group assignments?

Input

The first line of the input contains a single integer n ($3 \leq n \leq 100$), which is the number of people at the park.

The next n lines describe the locations of the people at the park. Each line contains exactly two integers x ($-10\,000 \leq x \leq 10\,000$) and y ($-10\,000 \leq y \leq 10\,000$), which are the coordinates of the person at the park. No two people are at the same location.

Output

Display the maximum possible Social Distance Factor squared.

Sample Input 1

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

Sample Output 1

```
13
```

Sample Input 2

```
4
1 1
5 5
2 2
4 4
```

Sample Output 2

```
8
```

Sample Input 3

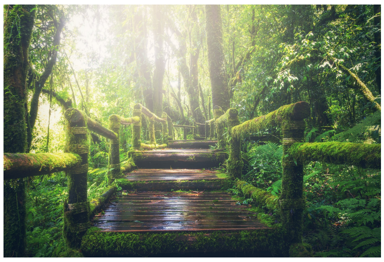
```
3
-1 3
3 4
-2 -1
```

Sample Output 3

```
17
```

This page is intentionally left (almost) blank.

Problem L
Legendary Walking Trails
Time limit: 2 seconds



Alex runs a national park. The national park maintains a number of walking trails. Alex needs to ensure that every walking trail is safe and remains legendary. Alex plans to hire zero or more walking trail inspectors. Each walking trail inspector charges the same flat fee, so Alex wants to hire the minimum number of walking trail inspectors.

The national park is on the side of a large mountain. There are N walking trail endpoints on the mountain, each at a unique altitude. Each of these is labelled from 1 to N in strictly descending order of their altitude. There are M walking trails. Each trail connects two endpoints.

A walking trail inspector can be instructed to start at any walking trail endpoint, take a sequence of walking trails that are connected by their endpoints, and finish at any endpoint. However, walking trail inspectors are peculiar: they refuse to travel uphill, so they will only take a path comprising a sequence of trails that is always going downhill.

Alex needs to make sure that every walking trail is inspected by at least one walking trail inspector. Note that it is acceptable for two or more walking trail inspectors to inspect the same walking trail as they inspect their path of trails. Likewise, it is acceptable for two or more inspectors to pass the same walking trail endpoint.

What is the minimum number of walking trail inspectors required?

Input

The first line of input contains two integers N ($1 \leq N \leq 100$), which is the number of endpoints, and M ($0 \leq M \leq \frac{N(N-1)}{2}$), which is the number of walking trails.

The next M lines describe the walking trails. Each of these lines contains exactly two integers u and v ($1 \leq u < v \leq N$), which indicates that there is a trail between endpoints u and v. Each pair of endpoints is distinct.

Output

Display the minimum number of walking trail inspectors required.

Sample Input 1	Sample Output 1
3 2 1 2 1 3	2

Sample Input 2	Sample Output 2
8 8 1 2 1 3 2 4 3 4 4 5 4 6 6 7 6 8	3

This page is intentionally left (almost) blank.