South Pacific
Competitive
Programming
Association

ICPC SOUTH PACIFIC PRELIMINARY CONTEST LEVEL A

AUGUST 31, 2024

Contest Problems

This contest contains twelve problems. 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
- $(a - b)$ 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:

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

Problem A
Arranging Sticks
Time limit: 2 seconds

Janine has a collection of sticks, each with a unique length. She wants to arrange these sticks in such a way that for any three consecutive sticks, the median length of the three is not the middle stick. This specific arrangement is called a *delightful* order.

For instance, if Janine has sticks with lengths 2, 3, 5, 7, 11, 13, and 17, she can order them as follows: 13, 5, 7, 2, 17, 3, 11. In this arrangement, the medians of each triplet $(13, 5, 7)$, $(5, 7, 2)$, $(7, 2, 17)$, $(2, 17, 3)$, and $(17, 3, 11)$ are 7, 5, 7, 3, and 11 respectively, and none of these medians are the middle stick of the triplet, so this is a delightful order.

Janine has proven that a delightful ordering is always possible, but is having a hard time finding one. Given the lengths of the sticks, arrange them in a delightful order.

Input

The first line of input contains a single integer n $(3 \leq n \leq 2\,000)$, which is the number of sticks.

The next line contains n distinct integers s_i $(1 \leq s_i \leq 100\,000)$, which are the lengths of the n sticks.

Output

Display a delightful ordering. If there are multiple correct solutions, any will be accepted.

Sample Input 1	Sample Output 1
7 2 3 5 7 11 13 17	13 5 7 2 17 3 11

Sample Input 2	Sample Output 2
4 10 100 1 1000	1000 1 100 10

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

This page is intentionally left (almost) blank.

Problem B
Birthday Wizard
Time limit: 1 second

Nathan is a wizard, who is turning N years old today. He is very lonely and awkward, so each year he celebrates his birthday alone.

Last year, he turned $N - 1$ years old, so he decorated his birthday cake with $N - 1$ candles. However, he felt that this was a waste of candles, and started thinking of more concise ways to represent his age. Eventually, using his wizard powers, he recognised that the traditional system represents his age in unary (base 1). This year, he plans to save candles by expressing his age in a different base.

For example, suppose Nathan turns five this year.

- The traditional unary representation is `11111`, with each digit represented by an individual candle, using five candles in total.
- If he instead uses ternary (base 3), his age could be written as `12`. He could represent this with one candle for the `1`, and a cluster of two candles for the `2`, for a total of three candles.
- Binary (base 2) is even better; in this system his age is `101`. He can place one candle for the first digit, leave a gap to represent the `0`, and place the second and final candle to represent the third digit.
- The best of all is base 5, where his age is `10`, requiring only one candle.

As Nathan always celebrates his birthday alone, he is not concerned with ambiguity. In particular, it does not matter that `101` could be misinterpreted as `11` or `1001`, or that `12` might be read as `21` from the other side of the cake.

Nathan would like to use as few candles as possible, by choosing an appropriate base between 2 and 10 inclusive. Help Nathan determine the smallest number of candles to use.

Input

The input consists of a single line containing a single integer N ($1 \leq N \leq 10^{18}$), the age to be represented on Nathan's cake this year.

Output

Display the smallest number of candles that Nathan can use.

Sample Input 1	Sample Output 1
5	1

Sample Input 2	Sample Output 2
51	3

Sample Input 3	Sample Output 3
1000000000000	1

This page is intentionally left (almost) blank.

Problem C

Crushing Monsters

Time limit: 1 second

Jimothy is facing the final boss in his favourite game, *Monster Hunter*. The boss has h hit points (HP), and Jimothy possesses potions and swords to diminish the boss's HP. Potions reduce the boss's HP by a percentage equal to the potion's potency, while swords reduce the boss's HP by a flat amount equal to their strength.

Items may be used in any order. For example, if Jimothy has a potion with 30 potency and a sword with 20 strength against a boss with 50HP, using the potion first would leave the boss with 35HP (30% off 50HP $=$ 35HP), and using the sword afterward would further reduce it to 15HP (35HP $-$ 20HP $=$ 15HP). Alternatively, using the sword first and then the potion would leave the boss with 21HP (50HP $-$ 20HP $=$ 30HP and 30% off 30HP $=$ 21HP). Potion percentages are calculated based on the boss's current HP, so if Jimothy had two potions with 50 potency each, using them would reduce the boss to 25% of its original HP (the first potion removes 50%, then the second potion removes 50% of the remaining, leaving 25% of the original HP).

The boss's HP can even drop below zero, though potions have no effect when the boss's HP is negative (swords still reduce the boss's HP). So if the boss has -10HP, then potions have no effect, and a sword with 5 strength would reduce the boss's HP to -15HP.

Each item can only be used once, and Jimothy has a limit, L, on the total number of items he can use. For instance, if the limit is 2 items, he could use 2 swords and 0 potions, 2 potions and 0 swords, or 1 of each. Jimothy aims to minimise the boss's remaining HP after using the items.

Given the potencies of the potions, strengths of the swords, and a restriction on the number of items Jimothy can use, determine the minimum HP the boss can have after using up to the allowed number of items.

Input

The first line of input contains four integers h ($1 \leq h \leq 100\,000$), which is the initial HP of the boss, n ($1 \leq n \leq 1\,000$), which is the number of potions Jimothy has, m ($1 \leq m \leq 1\,000$), which is the number of swords Jimothy has, and L ($1 \leq L \leq n + m$), which is the limit on the number of items they can use.

The second line contains n integers $p_1, p_2, \ldots, p_n$ ($1 \leq p_i \leq 99$), which are the potencies of the potions.

The third line contains m integers $s_1, s_2, \ldots, s_m$ ($1 \leq s_i \leq 100$), which are the strengths of the swords.

Output

Display the minimum HP the boss can have after using up to the allowed number of items. Your answer should have an absolute or relative error of less than 10^{-4}.

Sample Input 1	Sample Output 1

50 1 1 1
30
20
	30.0000000000

Sample Input 2	Sample Output 2

50 1 1 2
30
20
	15.0000000000

Sample Input 3

```
100 2 1 2
50 50
10
```

Sample Output 3

```
25.0000000000
```

Sample Input 4

```
100 2 1 2
50 90
1
```

Sample Output 4

```
5.0000000000
```

Sample Input 5

```
100 2 2 2
10 50
5 25
```

Sample Output 5

```
25.0000000000
```

Sample Input 6

```
5 2 1 2
10 5
10
```

Sample Output 6

```
-5.5000000000
```

Problem D
Dr Carboi
Time limit: 1 second

Dr Carboi loves toy cars and racing toy cars from his collection on his circular race track. But Dr Carboi has a problem — he spent all his pocket money on toy cars and now he has too many toy cars and only one race track!

If he puts all of his cars at the start of the track at once, they will crash into each other. Dr Carboi knows what to do though: he will stagger the starting times. He will place the first car on the track then wait exactly G seconds before placing the next car on the track, and so on, waiting exactly G seconds between placing each car at the start of the track. Cars immediately start driving once they are placed on the track.

The track is circular, so the cars will go round and round the track as they do laps. Dr Carboi doesn't want the first car hitting his fingers when it comes back around to start its next lap, so he won't place any more cars on the track at or after it finishes its first lap. Dr Carboi has worked out the number of seconds, T_i, that it takes for each of his R cars to do one lap around the track, and a car is considered to have finished a lap exactly T_i seconds after it started the lap.

The track also only has one lane, so cars cannot pass each other. If a car catches up with another car, it will crash into it and Dr Carboi will be sad. A car P crashes into another car Q if it started a lap after Q did but would finish the lap before or at the same time that Q finishes its lap.

Dr Carboi has realised that with these constraints he may not be able to race all his cars in one go, so he needs to settle for only racing some of them. If he has to choose only some of his cars, he wants to race only his fastest (and hence coolest) cars from his collection for L laps. That is, if he races K cars, then he will pick the top K fastest cars. Note that lower T_i means faster, and he doesn't care about how ties for T_i are broken.

Just after a car finishes its last lap, Dr Carboi will scoop it up off the track away from danger. Up until then, including the instant it finishes its last lap, it may still crash into other cars.

Dr Carboi wants to know K, the maximum number of toy cars he can race on the track at once for L laps, without any cars crashing into each other and using his K fastest cars. Dr Carboi can place the K cars on the track in any order (staggering their start times by G seconds, as described above) and they all must be placed before the first car finishes its first lap to avoid it hitting his fingers.

Input

The first line contains three integers, L ($2 \leq L \leq 10$) which is the number of laps in the race, G ($1 \leq G \leq 1000$) which is the number of seconds Dr Carboi waits between placing each toy car, and R ($1 \leq R \leq 2000$) which is the number of toy cars.

The following R lines each contain a single integer T_i ($1 \leq T_i \leq 2 \times 10^7$) which is the number of seconds for the ith car to do one lap of the track.

Output

Display the maximum number of cars Dr Carboi can race.

Sample Input 1	Sample Output 1

2 100 2
100
100
	1

Sample Input 2

```
2 100 2
200
150
```

Sample Output 2

```
1
```

Sample Input 3

```
2 100 2
100
149
```

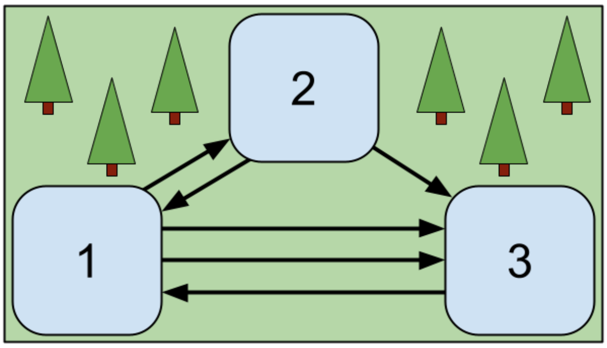
Sample Output 3

```
2
```

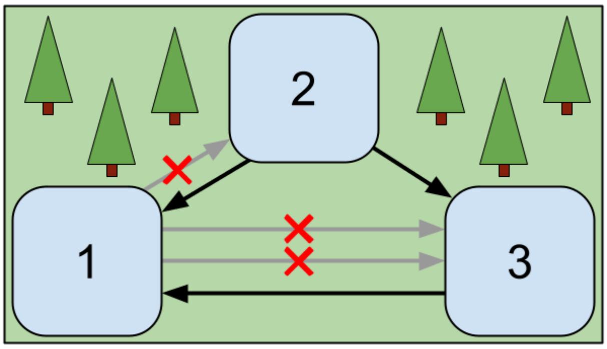
Problem E
Eliot's Friends
Time limit: 6 seconds

Eliot is planning a garden maze for all his friends to enjoy on the grounds of his palatial manor. The maze consists of N clearings connected by M passageways. Each passageway can only be traversed in one direction since the thorns of the plants making up the walls of the maze all point in the same direction. There can be multiple passageways connecting a pair of clearings and these passageways do not all need to go in the same direction.

Eliot has lots of friends who will traverse the maze one at a time. He requires that each friend start at the designated start clearing and traverse the maze through some path ending at the designated finish clearing. All the clearings have been given the indices 1 to N and all the passageways have been given the indices 1 to M. The start clearing's index is 1 and the finish clearing's index is N. A path is a sequence of passageways $p_1, p_2, \ldots, p_k$ such that each pair of consecutive passageways $p_i = (a, b), p_{i+1} = (c, d)$ are connected by a clearing, so $b = c$.



Because Eliot wants his friends to enjoy the maze, he insists on them each choosing a different path. He considers two paths $p_1, p_2, \ldots, p_k$ and $q_1, q_2, \ldots, q_\ell$ different if they do not share any passageways, so $p_i \neq q_j$ for all combinations of i, j. Eliot will viciously attack any friend who does not follow this rule, and they will not be his friend afterwards. Eliot will also mercilessly attack any friend who does not make it to the finish clearing, and they will not be his friend afterwards.



There is a problem. Eliot has too many friends. He only wants to have at most F friends remaining after all his friends have attempted to traverse the garden maze. To this end, Eliot needs to destroy some of the passageways to ensure that at most F friends could possibly make it through the maze without being attacked. Since Eliot likes his garden, he wants to destroy the minimum number of passageways possible to achieve his goal. Note that Eliot does not mind having fewer than F friends; he only wants to ensure that no matter what order the friends traverse the maze in and which paths they choose to take, he will have at most F friends remaining.

Help Eliot destroy the minimum number of passageways so that he has at most F friends.

The images depict the first sample input. Eliot can ensure he has 0 friends by destroying all the passageways exiting the start clearing.

Input

The first line contains two integers N $(2 \leq N \leq 250)$ and F $(0 \leq F \leq 10^9)$, denoting that there are N clearings and that Eliot wants at most F friends.

The next N lines describe the passageways. Each line contains N integers. The jth integer on the ith line denotes the number of different passageways that go from clearing i to clearing j and is in the inclusive range from 0 to 10^9. It is guaranteed that this number will be zero if $i = j$.

Output

Display N lines each containing N integers. These denote which passageways remain after Eliot has had the opportunity to destroy some passageways. The jth integer on the ith line denotes the number of different passageways that go from clearing i to clearing j afterwards.

If there are multiple correct solutions, any will be accepted.

Sample Input 1

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

Sample Output 1

```
0 0 0
1 0 1
1 0 0
```

Sample Input 2

```
4 2
0 2 2 0
0 0 0 2
0 0 0 2
0 0 0 0
```

Sample Output 2

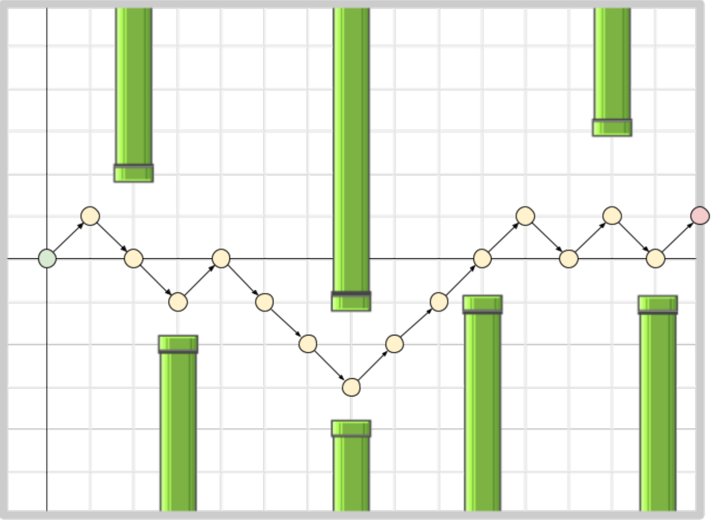
```
0 0 2 0
0 0 0 2
0 0 0 2
0 0 0 0
```

Problem F
Flappy Bird
Time limit: 9 seconds

Flappy Bird is a popular game where a bird navigates through obstacles. Each obstacle is either an up-pipe or a down-pipe. Up-pipes force the bird below a certain point, and down-pipes force it above a certain point.

The bird has 2 movement options each turn: flap (increase height by 1 and move right by 1) or no-flap (decrease height by 1 and move right by 1). If a down pipe is located at (x_D, y_D), then the bird may not touch any point (x_D, y) such that $y \leq y_D$. Similarly, if an up-pipe is located at (x_U, y_U), then the bird may not touch any point (x_U, y) such that $y \geq y_U$.

Pipes may intersect, potentially completely blocking the path if both up- and down-pipes overlap.



Flappy Bird starts at $(0, 0)$ and wins by passing the rightmost pipe without touching any pipe along the way (i.e., if the rightmost pipe is in column x_R, then the bird must reach column $x_R + 1$). Determine if winning is possible and devise a winning strategy if so.

Input

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

The next n lines describe the pipes. Each of these lines contains a character c (c is either `U` or `D`), which indicates if this pipe is an up-pipe or a down-pipe, an integer x ($1 \leq x \leq 200\,000$), which is the column of this pipe, and an integer y ($-200\,000 \leq y \leq 200\,000$), which is the row of this pipe.

Output

If it is not possible to win the game, display `Impossible`. Otherwise, display `Possible`, followed by a winning strategy. The winning strategy is a string with $x_R + 1$ characters in it, where x_R is the column of the rightmost pipe. If the ith character of the string is a +, then the bird should flap on turn i. Otherwise, the ith character of the string is a −, indicating that the bird should not flap on turn i.

If there are multiple solutions, any will be accepted.

Sample Input 1

```
7
U 2 2
D 3 -2
U 7 -1
D 7 -4
D 10 -1
U 13 3
D 14 -1
```

Sample Output 1

```
Possible
+--+---++++-+-+
```

Sample Input 2

```
2
D 5 10
U 6 -10
```

Sample Output 2

```
Impossible
```

Sample Input 3

```
3
D 2 1
D 1 0
D 3 2
```

Sample Output 3

```
Possible
++++
```

Sample Input 4

```
2
D 3 1
U 3 1
```

Sample Output 4

```
Impossible
```

Sample Input 5

```
2
U 3 -5
D 3 5
```

Sample Output 5

```
Impossible
```

Problem G
Grouping Words
Time limit: 1 second

The puzzle club at your university is working on a new word association game. The game consists of nine words, arranged in a three-by-three grid. To win the game, a player has to categorise the words into three groups of three words each, so that each pair of words in the same group is related.

Consider the example below.

PACIFIC	NEW	FIJI
PROGRAMMING	ZEALAND	CPLUSPLUS
PYTHON	SOUTH	JAVA

A player would be expected to know of many related words:

- all pairs of words in "South Pacific Competitive Programming Association",
- all pairs of words in country names (Australia, New Zealand, South Africa, and Fiji) are related to each other, and
- all pairs of programming languages (C++, Python, and Java) are related to each other.

There is only one way to group these words into three groups of three:

PACIFIC	*NEW*	*FIJI*
PROGRAMMING	*ZEALAND*	CPLUSPLUS
PYTHON	**SOUTH**	JAVA

The club needs your help to check whether a given puzzle can be solved or not! Given a grid of words, and a list of pairs of related words from the grid, report whether the puzzle can be solved, and if so, find a valid grouping.

Input

The first three lines of input each contain three words, and each word comprises up to 15 uppercase letters. All nine words are distinct.

The next line contains a single integer m ($9 \leq m \leq 36$), which is the number of pairs of related words.

The next m lines each represent a pair of related words. Each such line contains two distinct words out of the nine listed in the first three lines. No pair of related words appears more than once (in either ordering). So if `a b` appears, then `b a` does not.

Output

If there is no valid grouping, display `Impossible`.

Otherwise, display `Possible`, then three lines each containing three words. Each of these lines must correspond to a group, so the first three words must be pairwise related, as well as the next three, and the final three.

Sample Input 1

```
CCC A AAA
DDD DDDD BB
DD BBB D
18
BBB DDD
DDDD CCC
A DD
A D
DD DDDD
CCC DDD
BB BBB
BB DD
DDD DDDD
AAA D
D BBB
DDD D
DD DDD
DD D
AAA A
DDDD D
DD BBB
A DDDD
```

Sample Output 1

```
Possible
CCC DDD DDDD
A AAA D
BB DD BBB
```

Sample Input 2

```
PACIFIC NEW FIJI
PROGRAMMING ZEALAND CPLUSPLUS
PYTHON SOUTH JAVA
12
SOUTH PACIFIC
SOUTH PROGRAMMING
PROGRAMMING PACIFIC
SOUTH ZEALAND
SOUTH FIJI
NEW ZEALAND
NEW FIJI
NEW SOUTH
FIJI ZEALAND
CPLUSPLUS JAVA
CPLUSPLUS PYTHON
JAVA PYTHON
```

Sample Output 2

```
Possible
PACIFIC PROGRAMMING SOUTH
NEW FIJI ZEALAND
CPLUSPLUS PYTHON JAVA
```

Sample Input 3

```
A AA AAA
B BB BBB
C CC CCC
12
A AA
AA AAA
B BB
BB BBB
C CC
CC CCC
A B
AA BB
AAA BBB
A C
AA CC
AAA CCC
```

Sample Output 3

```
Impossible
```

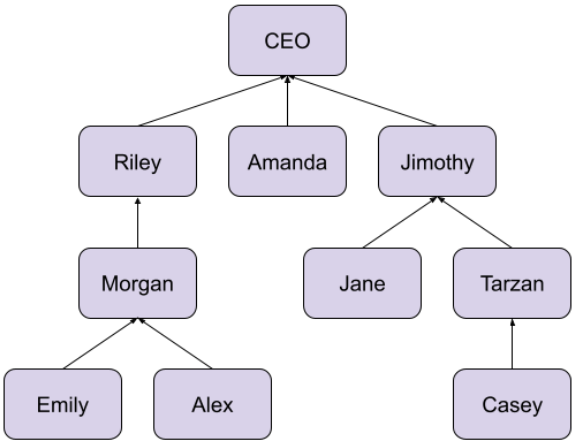
This page is intentionally left (almost) blank.

Problem H
Human Resources
Time limit: 1 second

James works at MegaCorp, where every employee except the CEO has exactly one manager. An employee's reporting chain includes themselves, their manager, their manager's manager, and so on. The CEO is at the top of everyone's reporting chain. MegaCorp needs to train some employees so that each employee can communicate with a trained manager.

The distance between an employee and someone in their reporting chain is defined as the number of managers between them, plus one. For example, an employee's direct manager is at distance 1, their manager's manager is at distance 2, and so forth. The distance from am employee to themself is 0. MegaCorp's policy dictates that for every employee (including the CEO), the distance to the nearest trained manager in their reporting chain (including themself) must be at most K.

Consider the following employee structure:



If $K = 5$, training the CEO alone suffices because everyone is within a distance of 5 from them. However, if $K = 2$, at least one additional manager needs training because some employees are at a distance of 3 from the CEO. Training the CEO, Morgan, and Tarzan is enough to satisfy the requirements.

Given the manager for each employee, determine the minimum number of managers that MegaCorp needs to train to ensure that every employee is within at most distance K from a trained manager in their reporting chain.

Input

The first line of input contains two integers n ($2 \leq n \leq 300$), which is the number of employees, and K ($1 \leq K \leq n$), which is the maximum distance from an employee to the nearest trained manager.

The next line contains $n - 1$ integers $m_2, m_3, \ldots, m_n$ ($1 \leq m_i < i$), which indicates that the manager of employee i is m_i. The CEO is employee 1.

Output

Display the minimum number of managers that MegaCorp needs to train.

Sample Input 1	Sample Output 1

`10 5`
`1 1 1 2 4 4 5 5 7`
	`1`

Sample Input 2

```
10 2
1 1 1 2 4 4 5 5 7
```

Sample Output 2

```
3
```

Problem I
Iguana Gift
Time limit: 1 second

The legendary iguanas Izzy and Iggy have been married for almost two years now! Izzy said that as an anniversary gift she wants a really nice palindrome. A palindrome is a string that is the same forwards and backwards.

Iggy wants to give Izzy the perfect palindrome, but he unfortunately procrastinated. Iggy currently has a string (which may or may not already be a palindrome). Iggy and Izzy's anniversary date is in just a few minutes so Iggy needs your help making a palindrome from what he has so far as quick as possible. However, Iggy likes what he has so far so the resulting palindrome must start with his original string.

What is the minimum number of characters Iggy must add to the end of his string to create a palindrome?

Input

The first and only line of input contains a single string consisting of at least 1 and at most 20 lowercase letters. This is the string Iggy currently has.

Output

Display the minimum number of letters Iggy needs to add to the end of his string to form a palindrome that starts with his original string.

Sample Input 1	Sample Output 1
`iloveyou`	`7`

Sample Input 2	Sample Output 2
`xoxoxoxoxoxoxoxoxoxo`	`1`

Sample Input 3	Sample Output 3
`icpc`	`1`

Sample Input 4	Sample Output 4
`tacocat`	`0`

Sample Input 5	Sample Output 5
`x`	`0`

This page is intentionally left (almost) blank.

Problem J
ja$on Playpen
Time limit: 4 seconds

Jay's son ja$on is back and he's got a new trick. ja$on is ten times as slick as the last time you saw him and he's trapped most of his extended family inside a playpen.

There are N playpen posts, which are points on a two dimensional plane. ja$on initially made his playpen the smallest convex polygon which contains all the playpen posts while not having any three consecutive collinear vertices (note that there may be posts on the edge of his playpen or completely inside his playpen that are not used as vertices). But, ja$on thinks his family is too free to range so he wants to make the playpen much smaller.

ja$on will select one post of the playpen and replace it with any other post. Replacing a post P with another post Q is equivalent to removing P, then reattaching the now unattached fencing to Q instead. The new post may already be on the boundary of the playpen, and the resulting playpen does not need to be a convex playpen, or may even become two playpens.

ja$on wants to minimise the area that his family can live in. Can you help ja$on trap his family by telling him the smallest total area he can achieve?

See figures 1 and 2 for an example of replacing a post of the playpen and the resulting set of playpens.

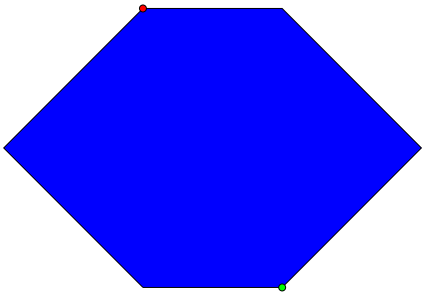


Figure 1: A convex polygon.

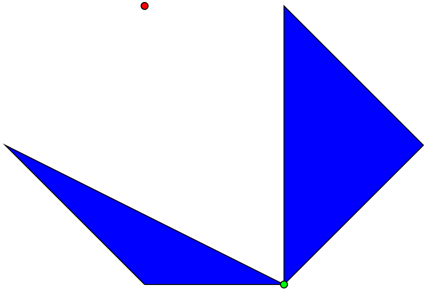


Figure 2: Figure 1 after replacing the top left point with the bottom right one.

Input

The first line contains an integer N ($3 \leq N \leq 200000$) which is the number of playpen posts.

The following N lines each contain two integers X and Y ($-10^9 \leq X, Y \leq 10^9$), representing the coordinates of each playpen post. No two posts are in the same location. There may be three or more posts that are collinear. It is guaranteed that there exists a convex playpen with non-zero area.

Output

Display a single integer showing the minimum area of the playpen(s), multiplied by two.

Sample Input 1

```
4
1 3
2 3
4 4
2 5
```

Sample Output 1

```
0
```

Sample Input 2

```
6
0 0
1 0
2 1
1 2
0 2
-1 1
```

Sample Output 2

```
2
```

Sample Input 3

```
6
0 0
15 0
30 0
10 20
20 20
15 30
```

Sample Output 3

```
0
```

Problem K
Knowledgeable AI Startup
Time limit: 4 seconds

There is a hot new platform game. It is called Marizzo and features a plumber who fixes skibidi toilets. The game is very popular and Mary has pitched a new startup to venture capital investors. Her startup idea is simple: use AI to play the game so that impressive speed-runs can be generated automatically and uploaded as NFTs to TouYube to collect ad revenue. Unfortunately, the deep learning methods Mary planned to use are not working well on Marizzo. You have been hired to develop an algorithm that can play the game optimally.

The game is played in a 2D space with width W and unlimited height. A sequence of N blocks fall from the sky one after the other. The game lasts a total of $N + 1$ seconds. During the ith second, a block with a width and height of 1 falls from above. The ith block falls at location l_i and falls directly down vertically. Each block will collide with either the ground or another block at the end of the ith second. Once it collides with anything, it stops there for the remainder of the game. No block falls in the final second of the game (the $N + 1$-th second).

The player controls Marizzo and can move him left or right. Marizzo has a height and width of 1 and can move very fast completing as many steps as needed between falling blocks. He can climb up or down to an adjacent block so long as the height difference between his current location is at most 1. Formally, Marizzo can step from location x to location y only if $|x - y| = 1$ and $|H_x - H_y| \leq 1$ where H_x denotes the height of location x. If a block falls on Marizzo, he will be crushed and the game is over. The goal of the game is to survive as long as possible so the score is the number of seconds survived. Note that if Marizzo can survive until the end of the game, that gets a score of $N + 1$.

Input

The first line of input contains two integers, N ($1 \leq N \leq 200\,000$) and W ($1 \leq W \leq 200\,000$). The next line contains N integers describing the sequence of falling blocks. The ith integer is the location of the ith block, l_i ($1 \leq l \leq W$).

Output

Display the maximum possible score.

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

Sample Input 2	Sample Output 2
4 2 1 1 2 2	3

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

This page is intentionally left (almost) blank.

Problem L
LLM
Time limit: 1 second

Large Language Models (LLMs) are super popular these days. However, you need tons of compute power in order to use them. As complex as they seem, the basics of LLMs are pretty straight forward: Given previous context, generate the next word/token/letter using some predetermined formulas. Janet wants to build a simplified version of this. This simplified LLM is called JanetBot. JanetBot can only generate 27 letters: the 26 lowercase letters and the *stop token* (described below).

JanetBot works by generating one letter at a time. To determine which letter to generate next, JanetBot looks at the last letter in the already generated text, computes a probability distribution, and then randomly selects a letter to append to the text from that probability distribution. The process then repeats for the next letter, and the next, and the next, until JanetBot generates the *stop token*, which is a special letter telling JanetBot to stop generating. The stop token is not appended to the text.

How do we compute that probability distribution? We train JanetBot! We give JanetBot many examples (called *training data*) of real text so that it can adjust its probabilities so that it generates reasonable text. Say ℓ_0 was the most recently generated letter, then the probability of generating some letter ℓ_1 (ℓ_1 may or may not be the stop token) is

$$\frac{\text{\# of times the substring `}\ell_0\ell_1\text{' appears in the training data}}{\text{\# of times `}\ell_0\text{' appears in the training data}}.$$

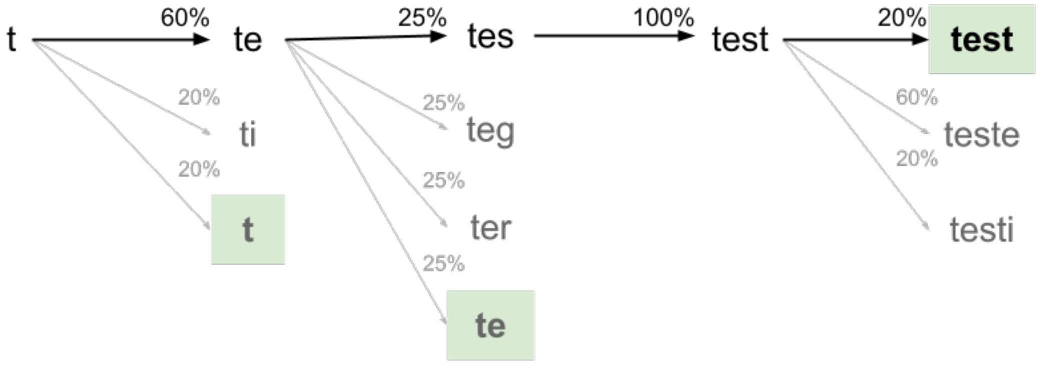
If ℓ_0 does not appear in the training data, then JanetBot will always generate the stop token as the next letter. There is a stop token at the end of each word in the training data.

For example, if the training data consists of 4 words: `international`, `collegiate`, `programming`, and `contest`, and the existing text is `test`, then the probability distribution for the next letter depends on the `t`'s in the input and is as follows:

Next Letter	`e`	`i`	Stop Token	Everything Else
Probability	$\frac{3}{5}$	$\frac{1}{5}$	$\frac{1}{5}$	$\frac{0}{5}$

Thus, there is a 60% chance the `e` is selected (making the text `teste`) and generation continues with the probability distribution described by the `e`'s in the input. There is a 20% chance the `i` is selected (making the text `testi`) and the generation continues. And there is a 20% chance the stop token is selected and the final text is `test`. On the other hand, if the existing text was `xyz`, then JanetBot is guaranteed to generate the stop token next since `z` does not appear in the training data.

So if JanetBot wants to generate the full word `test` (with `t` as the existing text), then there is a 60% chance to generate an `e` following a `t`, 25% to generate an `s` following an `e`, 100% chance of generating a `t` following an `s`, and a 20% chance to generate the stop token after a `t`. Since these chances are independent, the total probability is $60\% \times 25\% \times 100\% \times 20\% = 3\%$.



Janet is validating that JanetBot works properly. She has a target text she is attempting to generate. The text starts as the first letter of the target text. Given the training data and a target text, what is the probability that JanetBot generates the remainder of the target text (including the stop token)?

Input

The first line of input contains a single string T, which is the target string (T consists of only lowercase letters and has a length between 1 and 20, inclusive).

The second line of input contains a single integer n ($1 \leq n \leq 100$), which is the number of words in the training data.

The next n lines describe the training data. Each of these lines contains a single string S, which is a word in the training data (S consists of only lowercase letters and has a length between 1 and 20, inclusive).

Output

Display the probability that JanetBot generates T, given that it has already generated the first letter of T. The probability is a value between 0 and 1 inclusive. Your answer will be considered correct if it has an absolute or relative error of less than 10^{-9}.

Sample Input 1

```
test
4
international
collegiate
programming
contest
```

Sample Output 1

```
0.030000000000
```

Sample Input 2

```
ab
2
jab
table
```

Sample Output 2

```
0.500000000000
```

Sample Input 3

```
x
3
x
xy
xyz
```

Sample Output 3

```
0.333333333333
```

Sample Input 4

```
abc
1
ab
```

Sample Output 4

```
0.000000000000
```

Sample Input 5

```
aa
1
aa
```

Sample Output 5

```
0.250000000000
```

Sample Input 6	Sample Output 6
a 3 a a aa	0.750000000000

This page is intentionally left (almost) blank.