培正喇沙編程挑戰賽
LA SALLE – PUI CHING PROGRAMMING CHALLENGE

第五屆 · 2021

2021 年 8 月 8 日 (星期日)

香港培正中學

題號		名稱	時間限制	記憶體限制
A		Allowance Exhaustion	3 秒	256 MB
B		Boat Assignment	1 秒	256 MB
C		Copy of the String	0.5 秒	256 MB
D		Double Queue	0.5 秒	256 MB
E		Escape the Cube	0.5 秒	256 MB
F		Furthest Travel	1 秒	256 MB
G		Gold Medal Bout	0.5 秒	256 MB
H		How to Get Rice	0.5 秒	256 MB
I		Inno Per Gli Sconfitti	0.5 秒	256 MB
J		Just Skip It	0.5 秒	256 MB
K		Kario Mart	0.5 秒	256 MB
L		Lockout	0.5 秒	256 MB

Pascal	`ppcx64-3.0.0 -O2 -Sg -v0 -dEVAL -XS`
C	`gcc-7 -static -Wno-unused-result -DEVAL -lm -s -O2`
C++	`g++-7 -static -std=c++17 -Wno-unused-result -DEVAL -lm -s -O2`

編繹時間限制 10 秒　編繹記憶體限制 512 MB

作者/編製　鄭希哲 招朗軒 鍾懷哲 李卓傑 李呈禧 李啟峰

吳有孚 衛家熙 黃敏恆 黃梓駿 葉景輝 袁樂勤

NOTES

1. Output strings are checked case-sensitively.
2. Trailing spaces in output lines are allowed.
3. The end-of-line character at the end of the output is optional.
4. Using scientific notation for real number output is not allowed.
5. 64-bit integer types (e.g. `int64` for Pascal, `long long` for C/C++) may be required for some problems. The C format string token is `%lld`.

A Allowance Exhaustion

Time Limit: 3 seconds
Memory Limit: 256 MB
Input: standard input
Output: standard output

Melissa lives in London. She has been longing for a chance to explore London, but she has been trapped in her desk by large piles of documents in front of her. Today she finally gets T minutes from her manager to travel around, after which she must come back to the office.

Junctions and streets in London can be modelled as a simple undirected graph of N nodes and M edges. Nodes are numbered from 1 to N inclusive. For simplicity, let's assume it takes one minute to traverse any street. Yet, different streets bring on different levels of satisfaction to Melissa. The final satisfaction is the sum of the satisfaction levels of each street she has gone through. If Melissa passes through the same street twice, that street will contribute twice the satisfaction. As a rational agent, she would like to maximise her final satisfaction.

Just before Melissa steps out of the office, her manager has presented her with an unreasonable restriction: She must come back to the office after exactly T minutes have elapsed. Throughout the T minutes, she must not stay still at any junction/street. In other words, she must choose a route consisting of exactly T edges, starting and ending at her office (located on node 1).

Help Melissa calculate the maximum satisfaction she could possibly attain. If no such route exists, tell her it's impossible.

Please be aware that London was brought up by sound people - There won't be any street directing back to itself nor multiple streets between the same pair of junctions.

Input

The first line consists of 3 integers, N, M and T, which corresponds to the number of nodes, edges in the graph, and the number of minutes that Melissa has. ($1 \leq N \leq 1000$, $0 \leq M \leq 10^4$, $0 \leq T \leq 10^9$)

Input is then followed by M lines, each having 3 integers, u, v and w, denoting that an edge exists between u and v, and going through which would yield a satisfaction level of w. ($0 \leq w \leq 10^9$)

Output

Output a single integer: the maximum satisfaction possible. If no route fits in the restrictions, output -1 instead.

Examples

input

```
5 6 6
1 2 2
1 4 4
2 3 6
2 5 0
3 4 5
3 5 9
```

output

```
36
```

input

```
5 6 7
1 2 2
1 4 4
2 3 6
2 5 0
3 4 5
3 5 9
```

output

```
38
```

input

```
5 6 3
1 2 2
1 4 4
2 3 6
2 5 0
3 4 5
3 5 9
```

output

```
-1
```

Note

In the first sample, one best route is $1 \to 4 \to 3 \to 5 \to 3 \to 4 \to 1$.

In the second sample, one best route is $1 \to 2 \to 5 \to 3 \to 5 \to 3 \to 4 \to 1$.

B Boat Assignment

Time Limit: 1 second
Memory Limit: 256 MB
Input: standard input
Output: standard output

In the world of Animal Crossing, Timmy and $N-1$ other animals are waiting to cross the river. Unfortunately, the bridge is destroyed by a typhoon accidentally and therefore they have to reach another side of the river by boats.

There are M boats on this side of the river. The maximum load of the i-th boat is X_i kg. Timmy weighs A_1 kg and the i-th of the other $N-1$ animals weighs A_{i+1} kg. Each boat can carry any number of animals that their total weight does not exceed the maximum load X_i. Also, boats cannot be combined to increase the maximum load.

Being a best friend of Timmy, you are asked to assign the animals to the boats optimally. Please tell Timmy the minimum number of boats needed for all animals to cross the river, or it is impossible under the given conditions.

Input

The first line contains two integers $1 \leq N \leq 20$ and $1 \leq M \leq 50$.

The second line contains N integers A_i, representing the weight of the animals. ($1 \leq A_i \leq 10^8$)

The third line contains M integers X_i, representing the maximum load of each boat. ($1 \leq X_i \leq 10^9$)

Output

One integer, the minimum number of boats needed for all the animals to cross the river. Or `-1`, if it is impossible under the given condition.

Examples

input

```
3 3
7 3 9
5 10 9
```

output

```
2
```

input

```
1 2
10
5 5
```

output

```
-1
```

Note

In the first sample, we can assign the first and second animal to boat 2 and assign the third animal to boat 3.

This page is intentionally left blank.

C Copy of the String

Time Limit: 0.5 seconds
Memory Limit: 256 MB
Input: standard input
Output: standard output

Charlie and Cara play the same online game in their free time.

In the game, each player has a string with length N that contains only lowercase letters from the English alphabet. The strength of the player is then calculated by a mysterious function, with the string as the input.

A player is allowed to change their string by paying coins. Two operations can be performed arbitrary times:

1. Reorder all characters with K coins.
2. Modify a character to its adjacent character, in terms of alphabet, with 1 coin. For example, (`a`, `b`), (`z`, `y`) are pairs of adjacent characters, but (`a`, `c`), (`a`, `z`) are not.

Charlie has no idea what the mysterious function is, but he noticed that Cara's string S has been performing very well in the game. Therefore, Charlie decided to copy her string by changing his own string T using the two given operations.

As a friend of Charlie, you are asked to find the minimum coins Charlie needs to change his string into Cara's string.

Input

The first line contains two integers N and K. ($1 \le N \le 5 \times 10^5$, $0 \le K \le 10^9$).

The second line contains a string S with length N.

The third line contains a string T with length N.

Output

A single integer, the minimum coins Charlie needs in order to change his string into Cara's string.

Examples

input	output
4 3 dadc ddcc	4
5 5 feeaf ffdgc	10

This page is intentionally left blank.

D Double Queue

Time Limit: 0.5 seconds
Memory Limit: 256 MB
Input: standard input
Output: standard output

Dan would like to buy an apple from Alice's shop, and a banana from Bob's shop.

At time 0, Q_a customers suddenly appear and queue at Alice's shop, while Q_b customers appear and queue at Bob's shop. Dan can choose to join in one of the queues right after those customers in negligible time.

It takes S_a seconds for Alice to serve 1 customer, and S_b seconds for Bob to serve 1 customer. Both Alice and Bob would handle customers on a first-come-first-serve basis. If Dan and other customers enter the queue at the same time, Dan will queue after those customers.

After buying a fruit, it takes W seconds for Dan to walk from Alice's shop to Bob's shop, and vice versa.

Dan knows that there will be N more customers coming to Alice's shop, and M more customers coming to Bob's shop after time 0. He also knows when they will enter the queue.

What is the minimum time required for Dan to buy an apple and a banana from the shops?

Input

The first line consists of 5 integers, Q_a, Q_b, S_a, S_b, W ($0 \leq Q_a, Q_b \leq 10^5$, $1 \leq S_a, S_b, W \leq 100$).

The second line consists of 2 integers, N, M ($1 \leq N, M \leq 10^5$).

The third line consists of N integers, which is the time when new customers enter the queue for Alice's shop. The i-th integer correspond to A_i ($1 \leq A_i \leq 10^6$).

The fourth line consists of M integers, which is the time when new customers enter the queue for Bob's shop. The i-th integer correspond to B_i ($1 \leq B_i \leq 10^6$).

It is guaranteed that A_i and B_i are sorted in non-descending order.

Output

Output a single integer, the minimum time required for Dan to buy the fruits (in seconds).

Example

input

```
2 3 2 1 1
3 2
3 5 6
2 5
```

output

```
8
```

Note

Dan can buy from both shops in 8 seconds doing the following:

- At time 0, queue in Alice's shop after 2 customers.
- Wait for 4 seconds to be served by Alice.
- Takes 2 seconds to buy an apple from Alice.
- Walk to Bob's shop in 1 second.
- Get served immediately since no customers are queueing.
- Takes 1 seconds to buy a banana from Bob.

Total time used $= 4 + 2 + 1 + 1 = 8$.

E Escape the Cube

Time Limit: 0.5 seconds
Memory Limit: 256 MB
Input: standard input
Output: standard output

培正 喇沙 編程挑戰賽
LA SALLE – PUI CHING PROGRAMMING CHALLENGE
第五屆・2021年8月8日

The Cube (1997) is a science-fiction movie where some people are trapped in a mysterious cube-shaped room together. Some rooms contain traps that can kill people and they are to escape the cube without getting killed. After some exploration, they were able to discover the hidden patterns inside the cube and cracked the design of the cube. Given the hidden patterns, can you do the same?

The cube is sub-divided into uniquely-sized rooms. The dimension is $26 \times 26 \times 26$ rooms. Each room is represented by a serial of three numbers each having exactly 3 digits. The serial numbers never change. The sum of digits of each of the numbers give the initial x, y, and z coordinates of the room. For example, the room `242 614 064` is initially located at the coordinate $(2+4+2, 6+1+4, 0+6+4) = (8, 11, 10)$. In case you are concerned, the coordinates are designed so that no rooms will end up having the same coordinates.

The difficult part about the cube is that the rooms will move from time to time, making it very hard to travel and escape the cube. The moving pattern of the cube is described below: For each movement, the three coordinates will move with the same rule. For the i-th move, if i is not a multiple of 3, then the coordinates will change by the difference between the $(i \bmod 3)$-th digit and the $(i \bmod 3 + 1)$-th digit of the serial number; if i is a multiple of 3, then the coordinates will change by the difference between the last digit and the first digit of the serial number. Please refer to the following examples for details. (Note that the following examples only show the effect of ONE particular move, but you will be required to simulate ALL moves in this task.)

Example	Serial Number	Move i	i mod 3	Change in x coordinate	Change in y coordinate	Change in z coordinate
#1	242 614 064	1	1	2 - 4 = -2	6 - 1 = 5	0 - 6 = -6
#2	123 888 002	30	0	3 - 1 = 2	8 - 8 = 0	2 - 0 = 2

Given the serial number of a room, the number of movements M, determine the final location of the room.

Input

The first line contains the serial number of the room. It is represented by 3 strings that contains 3 digits each.

The second line contains an integer M – the number of movements. ($0 \le M \le 10^{18}$)

Output

Output 3 integers representing the x, y and z coordinates of the final position of the room after the M-th movement.

Examples

input

```
242 614 064
0
```

output

```
8 11 10
```

input

```
242 614 064
1
```

output

```
6 16 4
```

input

```
456 111 232
1000000
```

output

```
14 3 6
```

Note

Samples 1 and 2 show that the room moved by $(-2, 5, -6)$ for the 1st movement.

F Furthest Travel

Time Limit: 1 second
Memory Limit: 256 MB
Input: standard input
Output: standard output

Flores has invented a new pre-programmed drone. The drone has to perform a trial mission in a cave of N rooms numbered from 1 to N. The cave has $N-1$ bidirectional corridors, each connecting two rooms, such that it is possible to visit between any rooms. In terms of graph theory, the rooms and corridors form a tree.

The mission is going to last for T days, which can be represented as a sequence of $T+1$ rooms $r_0, r_1, r_2, \ldots, r_T$. At the beginning of the first day, the drone shall start in any of the N rooms, denoted as room r_0 in the sequence. On each of the T days, the drone shall travel to the room furthest away from the room it starts, and stay in the destination room overnight. If there are more than one furthest room, the drone is allowed to reach any of them. The distance between two rooms is defined as the minimum number of corridors required to travel.

Formally:

- The drone can start in any room, denoted as r_0, at the beginning of day 1.
- On each day i ($1 \le i \le T$), the drone travels from room r_{i-1} to r_i, where room r_i is one of the furthest room from room r_{i-1}. The drone will stay in the room r_i overnight.

Flores would like to know what is the number of possible sequences $r_0, r_1, \ldots, r_T$ that satisfy all the requirements mentioned.

Input

The first line contains a single integer N ($2 \le N \le 3000$).

The next $N-1$ line, each contains two integers x and y, representing a corridor connecting room x and room y ($1 \le x, y \le N$, $x \ne y$).

The last line contains a single integer T ($1 \le T \le 10^{18}$).

Output

A single integer, the number of possible sequences $r_0, r_1, \ldots, r_T$ that fulfill all the requirements mentioned. Please output the answer modulo $10^9 + 7$.

Examples

input

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

output

```
6
```

input

```
4
1 2
1 3
1 4
28
```

output

```
207959545
```

Note

In the first sample, the 6 possible sequences are $[1,4,5]$, $[1,5,4]$, $[2,4,5]$, $[3,5,4]$, $[4,5,4]$, $[5,4,5]$.

In the second sample, the answer is $1,207,959,552 \mod 1,000,000,007 = 207,959,545$.

G Gold Medal Bout

Time Limit: 0.5 seconds
Memory Limit: 256 MB
Input: standard input
Output: standard output

Today is the last day of the Tokyo 2020 Summer Olympic Games, with the closing ceremony going to take place tonight at 7PM HKT. In the last 16 days, Hong Kong Team has been continuously making history with by-far the best result of 1 Gold, 2 Silver and 3 Bronze Medals, bringing pride to all Hong Kong citizens. Such credit also belongs to all other Hong Kong athletes, even without medals, since they too have paid numerous effort and fought to their best.

Steve is particularly into this year's Olympics, especially when he saw two LSC Alumni participating in the Games: Choi Chun Yin Ryan (2015) in Fencing and Lam Siu-hang (2015) playing Table Tennis. The three matches played by Lam were tough, yet Lam was able to withstand the pressure and managed to make a comeback in both Round 1 and Round 2, before losing to Japan's world No. 4 Harimoto Tomokazu, still forcing the opponent into ties at 10-10 for three times.

Steve decides to learn more about table tennis rules. The scoring system is as follows:

1. A player can win a game of table tennis by scoring 11 points. The first to 11 points is declared the winner of the game.
2. If the points are tied at 10-10, a player then has to strive for a two-point lead to win the game.
3. A table tennis match consists of multiple games. Singles matches are played best-of-seven, i.e. the first player to win 4 **games** wins the **match**.

To mark the end of the Tokyo 2020 Olympics, Steve challenges you with a little twist. Steve will pick any arbitrary match, sums up the points of all games, and tell you how much points each player scores, as well as the winner of the match. You then have to try to construct a possible set of game scores to satisfy all the conditions, or determine if such match result is impossible.

Input

A single line containing three integers P_1, P_2, W – Player 1's total points, Player 2's total points, and which player wins the match. ($0 \leq P_1, P_2 \leq 1000, 1 \leq W \leq 2$)

Output

Output the match scores in the first line, followed by individual game points in each line.

If there are multiple set of solutions, output any of them. If there are no possible solutions, output `-1`.

Examples

input

```
69 63 1
```

output

```
4 3
7 11
11 9
6 11
11 6
11 4
12 14
11 8
```

input

```
61 65 1
```

output

```
4 3
11 7
7 11
4 11
5 11
11 9
12 10
11 6
```

input

```
51 56 2
```

output

```
1 4
9 11
11 13
11 8
10 12
10 12
```

input

```
45 0 1
```

output

```
-1
```

Note

Samples 1-3 are the actual match results between Lam Siu-hang and (1) Brian Afanador from Puerto Rico, (2) Sathiyan Gnanasekaran from India, and (3) Harimoto Tomokazu from Japan.

H How to Get Rice

Time Limit: 0.5 seconds
Memory Limit: 256 MB
Input: standard input
Output: standard output

Once upon a time in India, there was a king who was a big fan in chess and had the habit of challenging wise visitors to a game of chess. One day, a *sage* visits the country and was challenged by the king. The sage gladly accepted. The king was so confident in himself that he offered any reward the sage asks for. The sage modestly asked just for a few grains of rice in the following manner: on the 8×8 chessboard, the king was to put a single grain of rice on the first chess square, then two grains on the second square, four grains on the third and so on, doubling each time while placing on every consequent square. The king accepted the sage's request.

The sage won the chess game, and it is time for the king to reward the sage. He ordered his servants to bring a bag of rice to the chessboard, and started placing rice grains according to the arrangement. The king quickly realised that he was unable to fulfill his promise, because by the 20th square there would be $1,048,575$ grains of rice placed on the chessboard, and when all 64 squares were filled, the king would have to put a total of $18,446,744,073,709,551,615$ on the chessboard.

From this point onwards versions of this story start to differ. Some say the king was furious and had the sage killed for tricking him; some have a happy ending and the sage became the wealthiest person in the world, with the king paying the debt in instalments. Here we modify the game: instead of placing rice grains on the chessboard, we place them in a bowl for reasons of hygiene; and instead of using a 8×8 chessboard, we use a customized $1 \times N$ one.

The operations are also modified. The game starts by placing a piece of chess on the leftmost Cell 1, and 1 grain of rice in the bowl, as well as a *multiplier* $K = 1$. Each time you are allowed to perform one of two operations:

1. Move 1 cell to the right, add 1 grain of rice to the bowl, and add 1 to the multiplier K.
2. Move K cells to the right, and multiply the amount of rice in the bowl by K. The multiplier remains unchanged.

Your reward will be the amount of rice you have in the bowl once the piece of chess reaches Cell N. Obviously you cannot move that chess beyond Cell N and out of the chessboard. What is the maximum amount of rice you can get? Since the actual amount can be astronomical as we saw in the story, output your answer modulo $1,000,000,007$.

Input

A single integer N – the length of the board. ($1 \leq N \leq 10^9$)

Output

Output a single integer R, the maximum amount of rice you can get, modulo $1,000,000,007$.

Examples

input	output
5	5

input	output
6	9

Note

In Sample 1,

- Start at Cell 1, $R = 1, K = 1$
- Move to Cell 2, $R = 2, K = 2$
- Move to Cell 4, $R = 4, K = 2$
- Move to Cell 5, $R = 5, K = 3$

In Sample 2,

- Start at Cell 1, $R = 1, K = 1$
- Move to Cell 2, $R = 2, K = 2$
- Move to Cell 3, $R = 3, K = 3$
- Move to Cell 6, $R = 9, K = 3$

Inno Per Gli Sconfitti

Time Limit: 0.5 seconds
Memory Limit: 256 MB
Input: standard input
Output: standard output

Jane used to work as a cashier at a grocery store in Hackerland. Due to COVID-19, she has lost the job. Therefore, she has decided to chase her long time dream by teaming up with a few friends to form a band called "Dear Jane".

The band recently released a new song "Inno Per Gli Sconfitti". To promote the song, they are planning to visit 3 different schools in Hackerland. There are N schools in Hackerland, numbered from 1 to N. The selected schools will be decided by public voting on Instagram. Each school has a separate post that Instagram users can give Like to. The 3 schools that receive the most Likes will be selected. If there are ties, then the tieing schools will have equal chance of getting selected.

Jane actually wants her 3 favourite schools X, Y, Z to be selected so she decided to rig the poll by buying Likes and end the poll immediately after that. She already knows how many Likes each school has. School i currently has A_i likes. In order to make her 3 favourite schools to have strictly more Likes than the other schools, what is the minimum number of Likes she has to buy? The ranking among the 3 favourite schools do not matter.

Input

The first line contains an integer N – the number of schools. ($4 \leq N \leq 100000$)

The second line contains N integers $A_1, A_2, \ldots, A_N$ – the current number of Likes each school has. ($0 \leq A_i \leq 10^6$).

The third line contains 3 distinct integers X, Y, Z – the indices of Jane's favourite schools. ($1 \leq X, Y, Z \leq N$)

Output

Output a single integer, the minimum number of Likes that Jane needs to buy.

Examples

input

```
5
10 20 10 15 10
1 3 5
```

output

```
33
```

input

```
9
11 9 1 12 14 3 5 10 7
8 4 5
```

output

```
2
```

Note

In the first sample, Jane needs to buy 11 Likes for each of her favourite school to beat the 4th place school which has 10 Likes.

This page is intentionally left blank.

J Just Skip It

Time Limit: 0.5 seconds
Memory Limit: 256 MB
Input: standard input
Output: standard output

Justin likes watching videos on the video-sharing platform PuiLaTube.

Recently, he thinks PuiLaTube is putting more advertisements in videos. He categorizes the advertisements into 3 formats, as shown in the following:

1. Skippable video ads
   - Skippable video ads allow viewers to skip ads after 5 seconds.
   - Plays in video player. At least 12 seconds in length, at most 6 minutes (option to skip after 5 seconds).
2. Non-skippable video ads
   - Non-skippable video ads must be watched before a video can be viewed.
   - Plays in video player. 15 or 20 seconds in length, depending on regional standards.
3. Bumper ads
   - Short, non-skippable video ads of 6 seconds that must be watched before a video can be viewed.
   - Plays in video player, 6 seconds long.

Every time Justin clicks into a video, either no advertisement is shown, or one of the above types of advertisement is shown. The same thing happens when he refreshes the page.

Justin mastered the skill to check on the type of advertisement and discerning them in negligible time. He could also refresh immediately (in negligible time) by pressing F5 depending on the type of advertisement. It takes 1 second to refresh the page.

By looking up the documentation, Justin obtains the probabilities of each type of video advertisement appearing. The probabilities stay constant even if you refresh the page.

As a self-proclaimed PuiLaTube expert, Justin would adopt an advertisement refreshing strategy that minimizes the expected time of waiting before he can finally watch the video. Of course, he would skip any Skippable video ads immediately after 5 seconds if he decides to not refresh.

Can you find the minimized expected time?

Input

The first line includes 5 integers, $p_0, p_5, p_{15}, p_{20}, p_6$ ($0 < p_0 \leq 100$, $0 \leq p_5, p_{15}, p_{20}, p_6 \leq 100$).

$p_0\%$ represents probability of no advertisements are shown. $p_5\%$ represents probability of Skippable video ads is shown. $p_{15}\%$ and $p_{20}\%$ represents probabilities of 15 seconds and 20 seconds Non-skippable video ads is shown respectively. $p_6\%$ represents probability of Bumper ads is shown.

It is guaranteed that $p_0 + p_5 + p_{15} + p_{20} + p_6 = 100$, and at least one of p_{15} and p_{20} is 0.

Output

Output a single line containing a single real number, the minimized expected waiting time before Justin could watch the video (in seconds).

Your answer is considered correct if its absolute or relative error, whichever is less, is not greater than 10^{-6}.

Examples

input

```
50 0 50 0 0
```

output

```
1.0
```

input

```
15 25 0 35 25
```

output

```
4.625000
```

Note

Consider the first example. Only 15 seconds of Non-skippable video ads may be shown with a 50% probability.

If you choose not to refresh, the expected time is $50\% \times 15 = 7.5$ seconds.

If you choose to refresh one time, the expected time is $50\% \times (1 + 50\% \times 15) = 4.25$ seconds.

If you choose to refresh two times, the expected time is $50\% \times (1 + 50\% \times (1 + 50\% \times 15)) = 2.625$ seconds.

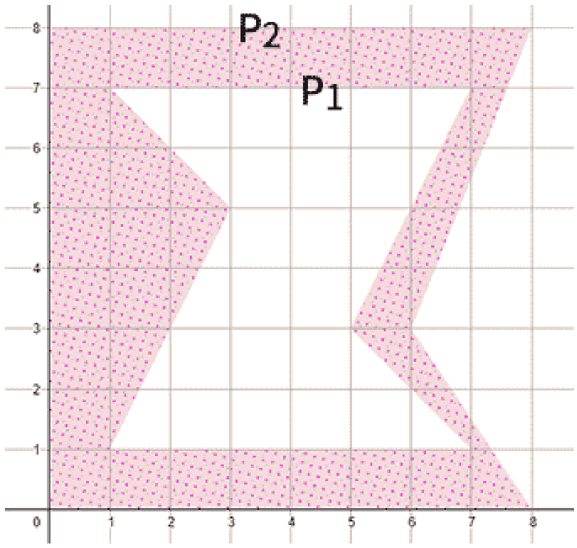
If you choose to refresh until there is no advertisement, the expected time is $50\% \times (1 + 50\% \times (1 + \ldots)) = 1$ second, which is the minimum expected time.

K Kario Mart

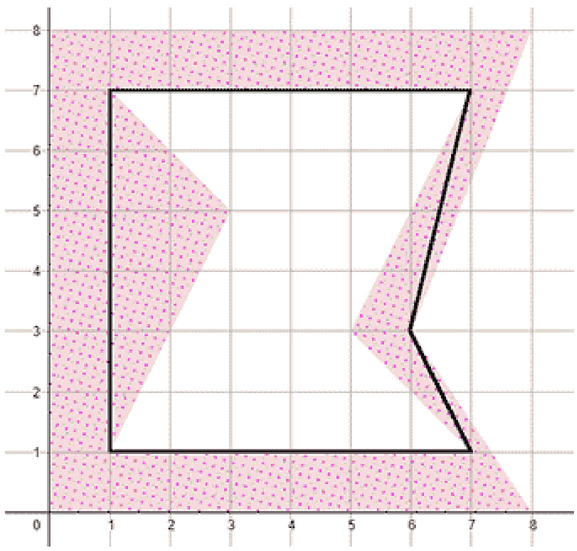
Time Limit: 0.5 seconds
Memory Limit: 256 MB
Input: standard input
Output: standard output

A new supermarket "Kario Mart" has opened in the neighbourhood! As a promotion event to attract customers, the supermarket is giving out 1000 dollars coupon to the first customer who completes a lap around the supermarket while pushing a shopping trolley.

The layout of Kario Mart can be described by two simple polygons, P_1, P_2, of N and M vertices respectively, where P_1 is fully enclosed by P_2. Also, P_1 and P_2 do not touch. The area of the Kario Mart is the area covered in P_2 that is not covered by P_1. For example, the red region in the following diagram is the walkable region.



A complete lap in the market is defined as a sequence of points $p_1, p_2, \ldots, p_k$ that forms a simple polygon that fully encloses P_1. The lap can start at any point in the market. The following diagram shows an example of a lap.



As a broke student, you would like to win the event at all costs. Write a program to find the length of the shortest possible lap.

Input

The first line consists of two integers, N and M ($3 \leq N, M \leq 100$), the number of vertices of P_1 and P_2 respectively.

The following N lines contains N pairs of integer (x, y) representing the coordinates of vertices of P_1 in clockwise order.

The final M lines contains M pairs of integer (x, y) representing the coordinates of vertices of P_2 in clockwise order.

For all input coordinates (x, y), $0 \leq |x|, |y| \leq 5000$.

For each polygon:

- There are no duplicate vertices.
- Exactly 2 line segments meet at each vertex.
- No angle is a multiple of 180 degrees.

Output

The output consists of one floating point number, the minimum length of the lap. Your output should have an absolute error or relative error of at most 10^{-6}.

Examples

input

```
6 4
1 7
7 7
5 3
7 1
1 1
3 5
0 0
0 8
8 8
8 0
```

output

```
24.0
```

input

```
6 5
1 7
7 7
5 3
7 1
1 1
3 5
0 0
0 8
8 8
6 3
8 0
```

output

```
24.359174
```

Note

Sample 2 corresponds to the example in the problem statement.

L Lockout

Time Limit: 0.5 seconds
Memory Limit: 256 MB
Input: standard input
Output: standard output

The long-awaited annual Lockout for Schools using Problems from Codeforces (LSPC) has commenced! For those who are new to Lockout, in each match, two players compete head-to-head in a thrilling test of speed and accuracy. Contestants are reminded to familiarize themselves with the revised rules this year:

- In a match, two contestants (denoted as contestant 1 and contestant 2) solve tasks from a given problemset consisting of 16 tasks arranged in a 4×4 grid. Each task is uniquely identified by a task ID.
- Contestants are allowed to attempt the tasks in any order. The first contestant to solve a task *claims* that task. Once a task is claimed, contestants are not allowed to submit solutions to that task anymore.
- The first contestant who claims 4 tasks that form a *line* or any 9 tasks wins. The match immediately ends when this condition is met. A *line* is defined to be any of the 4 rows, 4 columns, the leading diagonal and the trailing diagonal of the 4×4 grid.
- Matches have a time limit. If the winning condition is not met before the end of the match, the invigilator shall declare the match a *tie*.

Here's a warmup task designed to consolidate your understanding of the rules: given the list of task IDs in an arbitrary order along with the list of accepted submissions (submissions that result in a task being claimed) of a match in chronological order, can you reconstruct the 4×4 grid during the match?

Input

The first line contains 16 task IDs. A task ID consists of an integer between 1 and 9999 (inclusive) followed by an uppercase letter.

The second line contains an integer N ($0 \leq N \leq 16$), representing the number of accepted submissions during the match.

The next N lines each describes an accepted submission: it contains an integer (either 1 or 2) and a task ID, representing the contestant who made the submission and the task that they claimed respectively. These N task IDs are distinct.

It is guaranteed that the input corresponds to the accepted submissions of a Lockout match in chronological order.

Output

Output any possible 4×4 grid of task IDs that could have been used in the match.

Example

input

```
3P 1U 4I 1C 5H 9I 2N 6G 5X 3L 5A 8S 9A 7L 9L 3E
7
1 5H
2 3E
1 5X
2 5A
2 3L
2 1C
2 7L
```

output

```
9I 8S 9L 1C
2N 3P 7L 5H
6G 3E 1U 3L
5A 9A 4I 5X
```

Note

In the Sample Output above, the winning condition was not met after the first 6 accepted submissions. Once contestant 2 claimed task `7L`, they managed to formed a line (specifically along the trailing diagonal) so contestant 2 wins and the match is over.

On the other hand, the output

```
9I 8S 9L 1C
2N 3P 3L 5H
6G 3E 1U 7L
5A 9A 4I 5X
```

will result in a Wrong Answer verdict because once contestant 2 claims task `1C`, a line has been formed so the match should have ended then.