icpc | Asia Regionals SHU

上海大学
SHANGHAI UNIVERSITY

2021 ICPC

The 46th ICPC Asia Regional Programming Contest Shanghai Site
第46届ICPC国际大学生程序设计竞赛亚洲区域赛（上海）

2021年11月27−28日

正式赛试题册

Problem A. Strange Functions

Input file:	standard input
Output file:	standard output
Time limit:	2 seconds
Memory limit:	256 megabytes

Given n functions $f_1(x), f_2(x), \cdots, f_n(x)$, where

$$f_i(x) = \begin{cases} |\arctan(k_i \sec(x - a_i))| & (x \neq a_i + (k + \dfrac{1}{2})\pi \, (k = 0, \pm 1, \pm 2, \cdots)) \\ \dfrac{\pi}{2} & (x = a_i + (k + \dfrac{1}{2})\pi \, (k = 0, \pm 1, \pm 2, \cdots)) \end{cases}$$

For each function $f_i(x)$, determine if there is an x_i that $\forall j \in \{1, 2, \cdots, i-1, i+1, \cdots, n\}, f_i(x_i) < f_j(x_i)$.
Note that:

- "arctan" is the inverse function of "tan".

- $\sec(x) = \frac{1}{\cos(x)}$.

Input

The first line contains one integer $n \, (1 \leq n \leq 10^5)$, denoting the number of given functions.

Following n lines each contains two integers $k_i, a_i \, (1 \leq k_i \leq 10^5, |a_i| \leq 10^5)$, denoting the given functions.

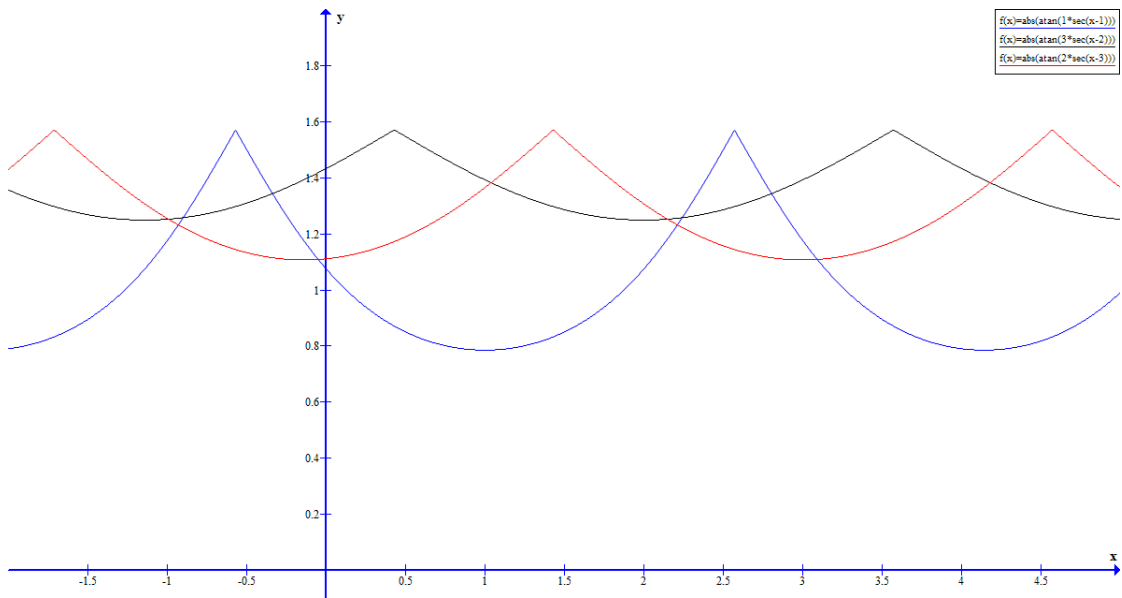It is guaranteed that $\forall 1 \leq i < j \leq n, \ k_i \neq k_j$ or $a_i \neq a_j$.

Output

Output one line containing one 01-string S of length n, where $S_i = 1$ iff such x_i exists, or $S_i = 0$.

Example

standard input	standard output
3 1 1 3 2 2 3	101

Note

Here is an illustration for the sample case.

f(x)=abs(atan(1*sec(x-1))))
f(x)=abs(atan(3*sec(x-2))))
f(x)=abs(atan(2*sec(x-3))))

Problem B. Strange Permutations

Input file:	standard input
Output file:	standard output
Time limit:	2 seconds
Memory limit:	256 megabytes

Given a permutation P of $\{1, 2, \cdots, n\}$, determine the number of $\{1, 2, \cdots, n\}$ permutations Q satisfying that $\forall i \in \{1, 2, \cdots, n-1\}, Q_{i+1} \neq P_{Q_i}$. Output the number modulo 998244353.

Input

The first line contains one integer $n\,(1 \leq n \leq 10^5)$, denoting the size of given permutation.

The second line contains n integers $P_1, P_2, \cdots, P_n\,(1 \leq P_i \leq n)$, denoting the given permutation.

It is guaranteed that $\{P_1, P_2, \cdots, P_n\} = \{1, 2, \cdots, n\}$.

Output

Output one line containing one integer, denoting the answer number modulo 998244353.

Example

standard input	standard output
4 3 4 1 2	8

Note

The 8 permutations are:

- $\{1, 2, 3, 4\}$

- $\{1, 4, 3, 2\}$

- $\{2, 1, 4, 3\}$

- $\{2, 3, 4, 1\}$

- $\{3, 2, 1, 4\}$

- $\{3, 4, 1, 2\}$

- $\{4, 1, 2, 3\}$

- $\{4, 3, 2, 1\}$

Problem C. Strange Matrices

Input file:	standard input
Output file:	standard output
Time limit:	1 second
Memory limit:	256 megabytes

As for a 01-matrix(whose entries are all either 0 or 1, similarly hereinafter) M of size $n \times m$, an index set S of the matrix is considered good if following two conditions are satisfied:

1. $M_{u,v} = 0$, for all $(u, v) \in S$

2. For each entry $M_{i,j} = 0 (1 \le i \le n, 1 \le j \le m)$, there exists an index $(u, v) \in S$ satisfying the following two conditions at the same time:

   - $i = u$ or $j = v$
   - $M_{x,y} = 0$ for all x, y such that $(x - i)(x - u) \le 0$ and $(y - j)(y - v) \le 0$

Moreover, the value of a 01-matrix is the minimum size among all of its good index sets.

Now given a 012-matrix, you should replace all the 2 entries to 0 or 1, and determine the minimum possible value among all replacing schemes. As can be seen, there are totally 2^{cnt_2} replacing schemes, where cnt_2 denotes the number of 2 entries in the given matrix.

Input

The first line contains two integers n, m $(1 \le n, m \le 8)$, denoting the size of given 012-matrix.

Following n lines each contains one 012-string of length m, where the j-th character in the i-th line among the n lines denotes entry $M_{i,j}$.

Output

Output one line containing one integer, denoting the minimum possible value after replacing.

Example

standard input	standard output
3 7 0001101 0201020 0001101	3

Note

One possible replacing scheme is:

$$0001101$$
$$0101000$$
$$0001101$$

One possible good set of the minimum size is $\{(1, 1), (2, 6), (3, 3)\}$

Problem D. Strange Fractions

Input file:	standard input
Output file:	standard output
Time limit:	1 second
Memory limit:	256 megabytes

Given a positive fraction $\frac{p}{q}$, you should find two positive integers a, b that $\frac{p}{q} = \frac{a}{b} + \frac{b}{a}$. If no such integers, report it.

Input

The first line contains one integer T $(1 \leq T \leq 10^5)$, denoting the number of test cases.

For each test case:

Input one line containing two integers p, q $(1 \leq p, q \leq 10^7)$, denoting the given fraction.

Output

For each test case:

If solution exists, output one line containing two integers a, b $(1 \leq a, b \leq 10^9)$, or print two zeros in one line if no solution.

Example

standard input	standard output
2	1 2
5 2	0 0
5 1	

Note

For the first case, $\frac{5}{2} = \frac{1}{2} + \frac{2}{1}$ holds. So one possible solution is $a = 1, b = 2$.

Problem E. Strange Integers

Input file:	standard input
Output file:	standard output
Time limit:	1 second
Memory limit:	256 megabytes

Given n integers $A_1, A_2, \cdots, A_n$ and a parameter k, you should choose some integers $A_{b_1}, A_{b_2}, \cdots, A_{b_m}$ $(1 \le b_1 < b_2 < \cdots < b_m \le n)$ so that $\forall 1 \le i < j \le m, |A_{b_i} - A_{b_j}| \ge k$. Determine the maximum number of the integers you can choose.

Input

The first line contains two integers n, k $(1 \le n \le 10^5, 0 \le k \le 10^9)$, denoting the number of given integers and the given parameter.

The second line contains n integers $A_1, A_2, \cdots, A_n$ $(1 \le A_i \le 10^9)$, denoting the given integers.

Output

Output one line containing one integer, denoting the maximum number of the integers you can choose.

Example

standard input	standard output
11 2 3 1 4 1 5 9 2 6 5 3 5	4

Note

One possible scheme is to choose $\{A_3 = 4, A_6 = 9, A_7 = 2, A_8 = 6\}$.

Problem F. Kaiji!

Input file:	standard input
Output file:	standard output
Time limit:	2 seconds
Memory limit:	512 megabytes

Itou Kaiji, a smart gambler, lost all his money once again and owed Hyodo Kazutaka a huge sum of money, who is very cruel and proposes a game to bet on Kaiji's four fingers!

Hyodo has a box with n nearly identical balls in it, where the i-th ball has an integer number $a_i (1 \le i \le n)$ on it, which is the only difference among the balls. At the first, Kaiji will be blindfolded, which means that he cannot see anything, then Hyodo will **arbitrarily** choose two close balls(Formally, assuming that the two chosen balls are i and j, there is no k that $(a_i - a_k)(a_j - a_k) < 0$), take them out of the box and put them into Kaiji's two hands. After that, Kaiji has to say "left hand" or "right hand", and Hyodo will tell him the number on the ball in the claimed hand. Finally, the most important part, Kaiji will answer the size relationship(greater than, less than, or equal to) between the told number and the number on the ball in the other hand. If Kaiji's answer is correct, he will be forgiven the debt, or he will lose his life.

Now, Kaiji needs your help, he wants to know the highest winning probability that can be guaranteed no matter what Hyodo's strategy is.

Note Kaiji will know all the numbers on balls and the whole rule before he is blindfolded.

Input

The first line contains one integer $T (1 \le T \le 1000)$, denoting the number of test cases.

For each test case:

Only one line contains 6 integer $n (2 \le n \le 10^7), a_0, A, B, C (0 \le a_0, A, B, C \le 10^9), M (1 \le M \le n)$, denoting the number of balls and generate parameters of a, where

$$a_i = (A \cdot a_{i-1}^2 + B \cdot a_{i-1} + C) \mod M + 1 \ (1 \le i \le n)$$

It is guaranteed that $\sum n \le 10^7$ among all test cases.

Output

For each test case:

Output one line containing one integer, denoting the winning probability modulo 998244353.

Note that for a rational number $\frac{p}{q}$ and an integer $k (0 \le k < P)$, if $kq \equiv p \pmod{P}$ holds, we say that $\frac{p}{q}$ modulo P equals k. In this problem, you can assume that there will be exactly one integer which equals the winning probability modulo 998244353.

Example

standard input	standard output
2	499122177
4 1 0 1 0 2	665496236
4 1 0 1 0 4	

Note

For the first test case:

$n = 4, \{a_1, a_2, a_3, a_4\} = \{2, 1, 2, 1\}$, one possible optimal strategy for Kaiji is:

Choose left hand or right hand randomly. Then:

- if the heard number is 1, guess two numbers are equal with a 50% probability, number on ball in the other hand is greater than number has been told with a 50% probability.

- if the heard number is 2, guess two numbers are equal with a 50% probability, number on ball in the other hand is less than number has been told with a 50% probability.

Under this strategy, Kaiji will have a 50% probability of winning, no matter how Hyodo chooses balls.

Here $499122177 \times 2 \equiv 1 \pmod{998244353}$ and 499122177 is the only integer which equals $\frac{1}{2}$ modulo 998244353, so the answer integer is 499122177.

Problem G. Edge Groups

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Given an undirected connected graph of n vertices and $n-1$ edges, where n is guaranteed to be odd. You want to divide all the $n-1$ edges to $\frac{n-1}{2}$ groups under following constraints:

- There are exactly 2 edges in each group

- The 2 edges in the same group share a common vertex

Determine the number of valid dividing schemes modulo 998244353. Two schemes are considered different if there are 2 edges that are in the same group in one scheme but not in the same group in the other scheme.

Input

The first line contains one integer $n\,(3 \le n \le 10^5)$, denoting the number of vertices.

Following $n-1$ lines each contains two integers $u, v\,(1 \le u < v \le n)$, denoting that vertex u, v are undirectedly connected by an edge.

It is guaranteed that n is odd and that the given graph is connected.

Output

Output one line containing one integer, denoting the number of valid dividing schemes modulo 998244353.

Example

standard input	standard output
7 1 2 1 3 1 7 4 7 5 7 6 7	3

Note

The 3 schemes are:

- The 3 edge groups are $\{1 \leftrightarrow 2, 1 \leftrightarrow 3\}, \{1 \leftrightarrow 7, 4 \leftrightarrow 7\}, \{5 \leftrightarrow 7, 6 \leftrightarrow 7\}$

- The 3 edge groups are $\{1 \leftrightarrow 2, 1 \leftrightarrow 3\}, \{1 \leftrightarrow 7, 5 \leftrightarrow 7\}, \{4 \leftrightarrow 7, 6 \leftrightarrow 7\}$

- The 3 edge groups are $\{1 \leftrightarrow 2, 1 \leftrightarrow 3\}, \{1 \leftrightarrow 7, 6 \leftrightarrow 7\}, \{4 \leftrightarrow 7, 5 \leftrightarrow 7\}$

Problem H. Life is a Game

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Life is a game.

The world can be regarded as an undirected connected graph of n cities and m undirected roads between the cities. Now you, the life game player, are going to play the life game on the world graph.

Initially, you are at the x-th city and of k social ability points. You can earn social ability points by living and working. Specifically, you can earn a_i social ability points by living and working in the i-th city. But in this problem, you cannot earn social ability points duplicatedly in one city, so you want to travel the world and earn more social ability points. However, the roads are not easy. Specifically, there is an ability threshold w_i for the i-th road, you should be of at least w_i social ability points to go through the road. Moreover, Your social ability point will not decrease when passing roads but just need to be at least w_i if you want to go through the i-th road.

So as you can see, the life game is just living, working and traveling repeatedly. There are q game saves. For each game save, the initial city and social ability point is given and the player has not lived or worked in any city. Now you, the real life game player, need to determine the maximum possible number of social ability points you can have in the end of the game and output it for each given game save.

Input

The first line contains three integers n, m, q $(1 \le n, m, q \le 10^5)$, denoting the number of cities, roads and game saves respectively.

The second line contains n integers $a_1, a_2, \cdots, a_n$ $(1 \le a_i \le 10^4)$, denoting the bonus social ability points for the cities.

Following m lines each contains three integers u, v, w $(1 \le u < v \le n, 1 \le w \le 10^9)$, denoting that cities u, v are undirectedly connected by a road of ability threshold w.

Following q lines each contains two integers x, k $(1 \le x \le n, 1 \le k \le 10^9)$, denoting the game saves.
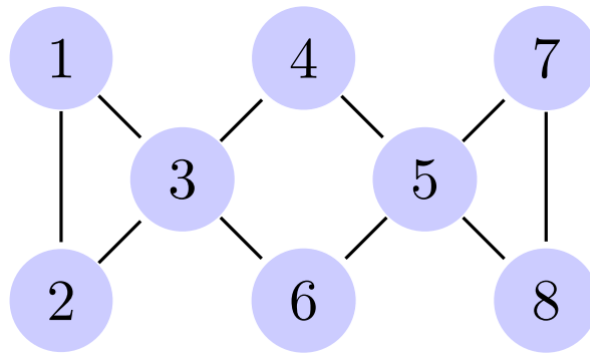
Output

For each game save, output one line containing one integer, denoting the maximum possible number of social ability points you can have.

Example

standard input	standard output
8 10 2	16
3 1 4 1 5 9 2 6	36
1 2 7	
1 3 11	
2 3 13	
3 4 1	
3 6 31415926	
4 5 27182818	
5 6 1	
5 7 23333	
5 8 55555	
7 8 37	
1 7	
8 30	

Note

Following is a illustration of the given graph.

- For the first game save, you can reach 4 cities $\{1, 2, 3, 4\}$ and have $7 + 3 + 1 + 4 + 1 = 16$ social ability points in the end

- For the second game save, you can only reach the initial city $\{8\}$ and have $30 + 6 = 36$ social ability points in the end

Problem I. Steadily Growing Steam

Input file:	standard input
Output file:	standard output
Time limit:	1 second
Memory limit:	512 megabytes

Alice enjoys playing a card game called Steadily Growing Steam (as known as SGS).

In this game, each player will play different roles and have different skills. Players get cards from the deck and use them to play the game. Each card has a numeric label t_i, the point number. In addition, each card has a value v_i.

Now Alice is playing this game with Bob. According to the skill of Alice's role, she can have Bob display n cards from the top of the deck. After that, Bob must choose some cards from the n cards and split the chosen cards into two sets that the sum of the cards' point numbers in the two sets are equal. In other words, if one of the sets is S and another is T, $S \cap T = \emptyset$ and $\sum_{i \in S} t_i = \sum_{j \in T} t_j$ (Note that $S \cup T = \{1, 2, \cdots n\}$ is not necessary). Then, Alice gets all of the cards in set S and Bob gets the cards in set T.

However, according to the skill of Bob's role, before choosing the two sets, he can choose at most k different cards and double their point numbers. In other words, he can choose a sequence $\{a_1, a_2, \cdots, a_r\}, (1 \leq a_1 < a_2 < \cdots < a_r \leq n, 0 \leq r \leq k)$ and for each $i \, (1 \leq i \leq r)$, change t_{a_i} into $2t_{a_i}$. After that he can continue choosing the two sets.

Alice and Bob are partners in this game. Now given the n cards from the deck, they want to know the maximum possible sum of the values of the cards they finally get. In other words, determine the maximum $\sum_{i \in S \cup T} v_i$ among all valid schemes(choose cards to double their point numbers, then choose cards and split them into two sets S, T of the same point number sum) and output it.

Input

The first line contains two integers $n \, (1 \leq n \leq 100)$ and $k \, (0 \leq k \leq n)$, denoting the number of the displayed cards and the maximum number of cards that Bob can choose to double their point numbers, respectively.

The $i + 1$ line contains two integers $v_i \, (|v_i| \leq 10^9)$ and $t_i \, (1 \leq t_i \leq 13)$, denoting the value and the point number of the i-th card, respectively.

Output

Output one line containing one integer, denoting the maximum sum of the value of the cards that Alice or Bob can get.

Example

standard input	standard output
4 1	21
10 1	
-5 3	
5 1	
6 1	

Note

One possible scheme:

Double t_1 and choose that $S = \{1\}, T = \{3, 4\}$, where the point number sum are both 2, and the sum of the card values is $10 + 5 + 6 = 21$.

Problem J. Two Binary Strings Problem

Input file:	standard input
Output file:	standard output
Time limit:	1 second
Memory limit:	512 megabytes

Given an integer n and two binary strings $a_1 a_2 \cdots a_n$ (denoted by A) and $b_1 b_2 \cdots b_n$ (denoted by B) of length n.

Define function:

$$f(l, r) = \begin{cases} 1, & \text{if } \sum_{i=l}^{r} a_i > \frac{r-l+1}{2} \\ 0, & \text{otherwise} \end{cases}$$

We say an integer k is lucky, iff for each $i\,(1 \le i \le n)$, $f\big(\max(i - k + 1, 1), i\big) = b_i$ holds.

For each integer $k\,(1 \le k \le n)$, determine if it is lucky.

Input

The first line contains one integer $T\,(1 \le T \le 50000)$, denoting the number of the test cases.

For each test case:

The first line contains one integer $n\,(1 \le n \le 50000)$, denoting the length of the two binary strings.

The second line contains one binary string A.

The third line contains one binary string B.

It is guaranteed that $\sum n \le 50000$.

Output

For each test case:

Output one line containing a binary string $c_1 c_2 \cdots c_n$ of length n, where $c_i = 1$ iff i is lucky while $c_i = 0$ iff i is not lucky.

Example

standard input	standard output
2	01000
5	00001100
11010	
11000	
8	
11110000	
11111100	

Problem K. Circle of Life

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Your friend Lucky is a little elf. Recently, Lucky discovered a magical creature, she named them "Twinkle". Lucky has a magical chain of n vertices and $n-1$ edges, where the vertices are connected by the edges one by one. We assume that the vertices are numbered 1 to n from left to right, and the i-th edge connects vertex i and vertex $i+1$. She found that Twinkles could live in the magical chain.

Lucky can put some Twinkles on some vertices simultaneously with her magic, but there should be at most one Twinkle in one vertex since the near Twinkles will perish together. Then, every second, each Twinkle will split into two Twinkles simultaneously and they will dash in the opposite directions to the two adjacent vertices.

Formally, a Twinkle in vertex u splits into two Twinkles, the left splited Twinkle and the right splited Twinkle.

- The left splited Twinkle will dash toward the left and reach the vertex $u-1$.

- The right splited Twinkle will dash toward the right and reach the vertex $u+1$.

But unluckily, if one side has no vertex, the Twinkle will dash out of the chain and die out(for the left splited Twinkle in vertex 1 and the right splited Twinkle in vertex n). Much more unfortunately, if two Twinkles dash to have a head-on collision (i.e. meet in the same vertex or edge), they will perish together with a tearing crash! Specifically, a crash will happen in two situations:

1. The right splited Twinkle in vertex i and the left splited Twinkle in vertex $i+2$ will crash in vertex $i+1$(assuming that vertex $i+1$ lives no Twinkle).

2. The right splited Twinkle in vertex i and the left splited Twinkle in vertex $i+1$ will crash in the i-th edge.

Lucky hopes there's always some Twinkle living on the chain. In addition, in order to be more convenient for Lucky to check the validity, there should be duplicated configurations within $2n$ seconds. Specifically, a configuration can be denoted by a binary string S of length n, where $S_i = 1$ iff vertex i lives a Twinkle, and C_i denotes the configuration after i seconds. Your task is to find an initial configuration C_0 so that for each $i\,(0 \le i \le 2n), C_i \ne 00\cdots00$ and that there exist two integers $i, j\,(0 \le i < j \le 2n), C_i = C_j$.

Input

Input one line containing one integer $n\,(2 \le n \le 123)$, denoting the length of the chain.

Output

If there is no solution, just output "Unlucky"(without quotes) in one line.

Otherwise, output one line containing one binary string, denoting the initial configuration you found.

Examples

standard input	standard output
2	10
4	1000

Note

For the first case, the configurations will change like this:

$$10 \to 01 \to 10$$

, where $C_0 = C_2$ holds.

For the second case, the configurations will change like this:

$$1000 \to 0100 \to 1010 \to 0001 \to 0010 \to 0101 \to 1000$$

, where $C_0 = C_6$ holds.

Problem L. Three,Three,Three

Input file:	standard input
Output file:	standard output
Time limit:	1 second
Memory limit:	256 megabytes

A poor little penguin has been thrown by a killer whale for a distance of one hundred and eight thousand li. The poor little penguin found itself on a mysterious island then. On the ground there are many graphs. The poor little penguin found that all the vertices are of degree three. "The Dao produced One; One produced Two; Two produced Three; Three produced All things", the poor little penguin recalled this mystical sentence. Suddenly, a quill-pen came into view. A sound said, "Every time you dip the quill-pen in the sea, you can one-touch-draw three edges. That is, you can start at a vertex, pass three different edges to a vertex." It is a parrot with three golden feathers on top of its head, "If all edges of a graph could be drawn once and only once by this way, then we call it a Three-throwable graph. You should find all the Three-throwable graphs, and show me how to throw three! As soon as you're done, I will bestow you the Produced-All-Things Power and send you back to avenge yourself on the killer whale!"

Too many graphs! So the poor little penguin asks you three for help. Please help it, for Light and Justice.

In short, given a graph of n vertices and m edges, where n is even and all the vertices are of degree 3. So it can be seen that $m = \frac{3n}{2}$ and that m is multiple of 3. You should try to split the graph into $\frac{m}{3}$ chains of length 3. Determine if the graph is Three-throwable graph(such chain spliting scheme exists) and output the splited chains if solution exists.

Input

The first line contains two integers n, m $(2 \leq n \leq 500)$, denoting the number of vertices and edges.

Following m lines each contains two integers u, v $(1 \leq u, v \leq n)$, denoting an edge connecting vertex u and v.

It is guaranteed that n is even, $m = \frac{3n}{2}$ and that all the vertices are of degree 3.

Output

If it is not a Three-throwable graph, output "IMPOSSIBLE"(without quotes) in one line.

Otherwise, output $\frac{m}{3}$ lines each contains four integers a_i, b_i, c_i, d_i, denoting one chain containing three edges $a_i \leftrightarrow b_i, b_i \leftrightarrow c_i, c_i \leftrightarrow d_i$. The multiset of given edges and the multiset of all the chain edges should be the same. If multiple solutions, output any one of them.

Examples

standard input	standard output
2 3 1 2 1 2 1 2	2 1 2 1
4 6 1 1 1 2 2 3 2 3 3 4 4 4	1 1 2 3 2 3 4 4
4 6 1 1 2 2 3 3 1 4 2 4 3 4	IMPOSSIBLE

Problem M. Harmony in Harmony

Input file:	standard input
Output file:	standard output
Time limit:	1 second
Memory limit:	256 megabytes

Haha is a humorous god of harmony. Why do I say that? In fact, Haha is forgetful, but Haha thinks that he has an excellent memory. Haha is responsible for balancing the power of various elements in the world. The power is derived from harmony fruits which grow in Hahiland. There are n tribes in Hahiland, each takes charge of one element, such as fire, water, etc. There is a fertile field in Hahiland, where the soil is suitable for growing the harmony fruits.

Every spring, Haha will divide the field into n parts, and each part is of the same total area. Then every tribe will choose one part to be their territory, then sow the harmony seeds on it, and give some element power to the seeds. Although the element power helps the plants to grow well, when the fruits are ripe, they could only be harvested by the one who has the same kind of element power.

"My memory is very good. I don't need to record my divide plan at all! These are all in my brain clearly!" Haha said with full confidence.

As autumn comes, Haha divides the field into n parts again from his memory. But it's actually not the same as last time. People are afraid to tell the harmony god that he is wrong. So they discuss how to allot the n parts in order to guarantee the minimum area of plants each tribe could harvest to be as large as possible.

The members of Hahiland are wondering how large the minimum area of plants each tribe could harvest could be guaranteed no matter how Haha divides the field, for their future generations. Please calculate it with your wisdom, so that they will be not so anxious.

In short, given n, determine:

$$\min_{S,A}\{\max_{p}\{\min_{i=1}^{n}\{|S_i \cap A_{p_i}|\}\}\}$$

, where S, A denote the field parts(the whole field and the parts can be denoted by some areas in 2D plane) divided in spring and autumn respectively(so $|S_1| = |S_2| = \cdots = |S_n| = |A_1| = |A_2| = \cdots = |A_n|$ holds), p denotes a permutation of $\{1, 2, \cdots, n\}$(so $\{p_1, p_2, \cdots, p_n\} = \{1, 2, \cdots, n\}$ holds).

Moreover, assume the total area of the whole field is 1 and the shape of the whole field doesn't count in this problem as can be seen.

Input

Input one line containing one integer n $(1 \le n \le 500)$, denoting the number of tribes.

Output

Output one line containing one real number round to 9 digits after the decimal point, denoting the answer.

Examples

standard input	standard output
1	1.000000000
2	0.250000000