NCPC 2014

October 4, 2014

acm International Collegiate Programming Contest

IBM event sponsor

Problems

A Amanda Lounges
B Basin City Surveillance
C Catalan Square
D Dice Game
E Opening Ceremony
F Particle Swapping
G Outing
H Clock Pictures
I How many squares?
J Road Work
K Train Passengers

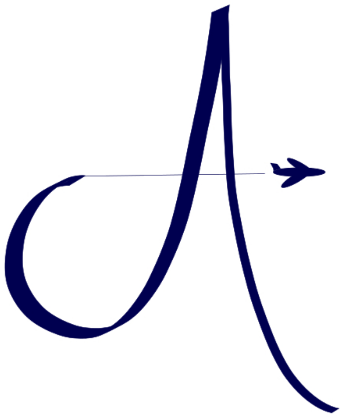
Do not open before the contest has started.

Advice, hints, and general information

- Your submissions will be run multiple times, on several different input files. If your submission is incorrect, the error message you get will be the error exhibited on the first input file on which you failed. E.g., if your instance is prone to crash but also incorrect, your submission may be judged as either "wrong answer" or "run time error", depending on which is discovered first.
- For problems with floating point output, we only require that your output is correct up to either a relative or absolute error of 10^{-6}. For example, this means that
  - If the correct answer is 0.05, any answer between 0.049999 and $.050001$ will be accepted.
  - If the correct answer is 50, any answer between 49.99995 and 50.00005 will be accepted.

  Any reasonable format for floating point numbers is acceptable. For instance, "17.000000", "0.17e2", and "17" are all acceptable ways of formatting the number 17. For the definition of reasonable, please use your common sense.

Problem A
Amanda Lounges
Problem ID: amanda


H. A. Hansen, cc-by-sa

AMANDA AIR has routes between many different airports, and has asked their most important frequent flyers, members of the AA Frequent Flyer program, which routes they most often fly. Based on this survey, Amanda, the CEO and owner, has concluded that AMANDA AIR will place lounges at some of the airports at which they operate.

However, since there are so many routes going between a wide variety of airports, she has hired you to determine how many lounges she needs to build, if at all possible, given the constraints set by her. This calculation is to be provided by you, before any lounges are built. Her requirements specifies that for some routes, there must be lounges at both airports, for other routes, there must be lounges at exactly one of the airports, and for some routes, there will be no lounges at the airports.

She is very economically minded and is demanding the absolute minimum number of lounges to be built.

Input

The first line contains two non-negative integers $1 \leq n, m \leq 200\ 000$, giving the number of airports and routes in the Amanda Catalog respectively. Thereafter follow m lines, each describing a route by three non-negative integers $1 \leq a, b \leq n$ and $c \in \{0, 1, 2\}$, where a and b are the airports the route connects and c is the number of lounges.

No route connects any airport with itself, and for any two airports at most one requirement for that route is given. As one would expect, 0 is a request for no lounge, 1 for a lounge at exactly one of the two airports and 2 for lounges at both airports.

Output

If it is possible to satisfy the requirements, give the minimum number of lounges necessary to do so. If it is not possible, output `impossible`.

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

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

Sample Input 3	Sample Output 3
4 5 1 2 1 2 3 0 2 4 1 3 1 1 3 4 1	2

Problem B
Basin City Surveillance
Problem ID: basincity

BASIN CITY is known for her incredibly high crime rates. The police see no option but to tighten security. They want to install traffic drones at different intersections to observe who's running on a red light. If a car runs a red light, the drone will chase and stop the car to give the driver an appropriate ticket. The drones are quite stupid, however, and a drone will stop before it comes to the next intersection as it might otherwise lose its way home, its home being the traffic light to which it is assigned. The drones are not able to detect the presence of other drones, so the police's R&D department found out that if a drone was placed at some intersection, then it was best not to put any drones at any of the neighbouring intersections. As is usual in many cities, there are no intersections in BASIN CITY with more than four other neighbouring intersections.

The drones are government funded, so the police force would like to buy as many drones as they are allowed to. Being the programmer-go-to for the BASIN CITY POLICE DEPARTMENT, they ask you to decide, for a given number of drones, whether it is feasible to position exactly this number of drones.

Input

The first line contains an integer k ($0 \leq k \leq 15$), giving the number of drones to position. Then follows one line with $1 \leq n \leq 100\,000$, the total number of intersections in BASIN CITY. Finally follow n lines describing consecutive intersections. The i-th line describes the i-th intersection in the following format: The line starts with one integer d ($0 \leq d \leq 4$) describing the number of intersections neighbouring the i-th one. Then follow d integers denoting the indices of these neighbouring intersections. They will be all distinct and different from i. The intersections are numbered from 1 to n.

Output

If it is possible to position k drones such that no two neighbouring intersections have been assigned a drone, output a single line containing `possible`. Otherwise, output a single line containing `impossible`.

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

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

Problem C
Catalan Square
Problem ID: catalansquare


Photo by Wikimedia Commons user Rauenstein

Last weekend you and your friends went to visit the local farmer's market at the town square. As you were standing around in a circle talking, you couldn't help overhearing two of your friends musing over what sounded like an interesting problem: They were considering the number of ways in which you could all shake hands, such that everyone in the circle simultaneously shaked hands with one other person, but where no arms crossed each other.

After a few seconds' thought you decided to join your two friends, to share (with them) the solution to their problem. "If we are $2n$ persons", you said, "pick any particular person, and let that person shake hands with somebody. That person will have to leave an even number of people on each side of the person with whom he/she shakes hands. Of the remaining $n-1$ pairs of people, he/she can leave zero on the right and $n-1$ pairs on the left, 1 on the right and $n-2$ pairs on the left, and so on. The pairs remaining on the right and left can independently choose any of the possible non-crossing handshake patterns, so the count C_n for n pairs of people is given by:

$$C_n = C_{n-1}C_0 + C_{n-2}C_1 + \ldots + C_1C_{n-2} + C_0C_{n-1},$$

which, together with the fact that $C_0 = C_1 = 1$, is just the definition of the Catalan numbers." By consulting your handy combinatorics book, you find out that there is a much more efficient formula for calculating C_n, namely:

$$C_n = \frac{\binom{2n}{n}}{n+1}.$$

After a collective groan from the group, your particularly cheeky friend Val called out "Well, since we are at the town square, why don't you try to square your Catalan numbers!". This was met with much rejoicing, while you started to think about how to square the *Catalan sequence*…

Task

Let C_n be the nth Catalan number as defined above. By regarding the sequence $(C_n)_{n\geq 0}$ of Catalan numbers, we can define a sequence $(S_n)_{n\geq 0}$, corresponding to "squaring the Catalan sequence", by considering the Cauchy product, or discrete convolution, of $(C_n)_{n\geq 0}$ with itself, i.e.,

$$S_n = \sum_{k=0}^{n} C_k C_{n-k}.$$

Your task is to write a program for calculating the number S_n.[1]

[1] To see why $(S_n)_{n\geq 0}$ could be said to correspond to the square of the Catalan sequence we could look at Cauchy products of power series. Suppose that $p(x) = \sum_{n=0}^{\infty} a_n x^n$ and $q(x) = \sum_{n=0}^{\infty} b_n x^n$, then $p(x) \cdot q(x) = \sum_{n=0}^{\infty} c_n x^n$ where $c_n = \sum_{k=0}^{n} a_k b_{n-k}$.

Input

The input contains one line containing one non-negative integer: n, with $0 \leq n \leq 5\,000$.

Output

Output a line containing S_n.

Sample Input 1	Sample Output 1
0	1

Sample Input 2	Sample Output 2
59	1583850964596120042686772779038896

Problem D
Dice Game
Problem ID: dicegame

Photo by JD Hancock

Gunnar and Emma play a lot of board games at home, so they own many dice that are not normal 6-sided dice. For example they own a die that has 10 sides with numbers $47, 48, \ldots, 56$ on it.

There has been a big storm in Stockholm, so Gunnar and Emma have been stuck at home without electricity for a couple of hours. They have finished playing all the games they have, so they came up with a new one. Each player has 2 dice which he or she rolls. The player with a bigger sum wins. If both sums are the same, the game ends in a tie.

Task

Given the description of Gunnar's and Emma's dice, which player has higher chances of winning?

All of their dice have the following property: each die contains numbers $a, a+1, \ldots, b$, where a and b are the lowest and highest numbers respectively on the die. Each number appears exactly on one side, so the die has $b - a + 1$ sides.

Input

The first line contains four integers a_1, b_1, a_2, b_2 that describe Gunnar's dice. Die number i contains numbers $a_i, a_i + 1, \ldots, b_i$ on its sides. You may assume that $1 \leq a_i \leq b_i \leq 100$. You can further assume that each die has at least four sides, so $a_i + 3 \leq b_i$.

The second line contains the description of Emma's dice in the same format.

Output

Output the name of the player that has higher probability of winning. Output "`Tie`" if both players have same probability of winning.

Sample Input 1	Sample Output 1
1 4 1 4 1 6 1 6	Emma

Sample Input 2	Sample Output 2
1 8 1 8 1 10 2 5	Tie

Sample Input 3	Sample Output 3
2 5 2 7 1 5 2 5	Gunnar

This page is intentionally left (almost) blank.

Problem E
Opening Ceremony
Problem ID: ceremony

From Wikipedia under Creative Commons licence, by G Laird

For the grand opening of the algorithmic games in NlogNsglow, a row of tower blocks is set to be demolished in a grand demonstration of renewal. Originally the plan was to accomplish this with controlled explosions, one for each tower block, but time constraints now require a hastier solution.

To help you remove the blocks more rapidly you have been given the use of a Universal Kinetic / Incandescent Energy Particle Cannon (UKIEPC). On a single charge, this cutting-edge contraption can remove either all of the floors in a single tower block, or all the x-th floors in all the blocks simultaneously, for user's choice of the floor number x. In the latter case, the blocks that are less than x floors high are left untouched, while for blocks having more than x floors, all the floors above the removed x-th one fall down by one level.

Task

Given the number of floors of all towers, output the minimum number of charges needed to eliminate all floors of all blocks.

Input

The first line of input contains the number of blocks n, where $2 \leq n \leq 100\,000$. The second line contains n consecutive block heights h_i for $i = 1, 2, \ldots, n$, where $1 \leq h_i \leq 1\,000\,000$.

Output

Output one line containing one integer: the minimum number of charges needed to tear down all the blocks.

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

Sample Input 2	Sample Output 2
5 1 1 1 1 10	2

This page is intentionally left (almost) blank.

Problem F
Particle Swapping
Problem ID: particles

From flickr under Creative Commons licence, by Tom Fassbender

The research team of prof. Feynmansson is preparing a new groundbreaking experiment in particle physics. On a special plate they have prepared a system consisting of a number of nodes connected via wires[2]. In the beginning of the experiment a pair of particles appears at two different nodes of the system: one normal particle of matter appears at some node A, and one corresponding particle of antimatter appears at some node B. The goal of the experiment is to swap these particles, i.e., to arrive at a state where the normal particle is at node B and the antiparticle is at node A. This state should be reached by a sequence of *moves*, where each move consists of transmitting one of the particles from its current location to a neighbouring node via a wire.

As you probably remember from popular science TV programmes, playing with matter and antimatter is usually not that safe. In particular, if particles of matter and antimatter get too close to each other, they will annihilate each other blowing up the whole experiment. Therefore, the research team would like to swap the locations of the particles in such a manner that the minimum Euclidean distance between them during the experiment is as large as possible. This minimum distance is called the *safeness* of the experiment. For simplicity, we assume that while a particle is transmitted via a wire we do not consider its location; in other words, the only risky moments during the experiment are when both particles are at some nodes. You may assume that it is always possible to swap the particles with positive safeness, that is, so that the particles are never placed at the same node during swapping.

Another catch is that the physicists do not know precisely where the particles will appear. They have made a list of potential pairs of initial locations (A, B), and for each of them they would like to know the maximum possible safeness of swapping the particles. Help them in this task.

Input

The first line of the input contains a single integer n ($1 \leq n \leq 500$), denoting the number of nodes in the system. Then follow n lines, each containing two integers x, y ($-10\,000 \leq x, y \leq 10\,000$); the numbers in the i-th line denote the coordinates on the plate of the i-th node. No two nodes are located at the same point.

The next line of the input contains a single integer m ($0 \leq m \leq 2\,000$), denoting the number of wires in the system. Then follow m lines; each line contains a description of a wire as a pair of integers a, b ($1 \leq a, b \leq n$, $a \neq b$), denoting the indices of the nodes that are connected by the wire. You may assume that no two nodes are connected by more than one wire, and no wire connects a node with itself.

[2]The wires may cross each other on the plate.

The next line of the input contains a single integer ℓ ($1 \leq \ell \leq \binom{n}{2}$), denoting the length of the list of potential initial positions prepared by the physicists. Then follow ℓ lines, each containing two integers a, b ($1 \leq a, b \leq n$, $a \neq b$), denoting the indices of the initial nodes A and B, respectively.

Output

Output exactly ℓ lines. The i-th line of the output should contain a single floating point number, being the maximum possible safeness for the i-th pair of initial positions listed by the physicists. Absolute or relative errors of value at most 10^{-6} will be tolerated.

Sample Input 1

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

Sample Output 1

```
1.00000000
3.16227766
2.23606798
1.41421356
3.16227766
```

Problem G
Outing
Problem ID: outing

Organising a group trip for the elderly can be a daunting task... Not least because of the fussy participants, each of whom will only make the trip on condition that some other participant also comes.

After some effort, you have taken from each of your participants a number, indicating that this participant will refuse to join the excursion unless the participant with that number also joins– the less choosy simply give their own number. This would be easy enough to resolve (just send all of them) but the bus you are going to use during the trip has only a fixed number of places.

From Wikipedia under Creative Commons licence, by Tom Page

Task

Given the preferences of all participants, find the maximum number of participants that can join.

Input

The first line of input contains two integers n and k ($1 \leq k \leq n \leq 1\,000$), where n denotes the total number of participants and k denotes the number of places on the bus.

The second line contains n integers x_i for $i = 1, 2, \ldots, n$, where $1 \leq x_i \leq n$. The meaning of x_i is that the i-th participant will refuse to join the excursion unless the x_i-th participant also joins.

Output

Output one integer: the maximum number of participants that can join the excursion, so that all the participants' preferences are obeyed and the capacity of the bus is not exceeded.

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

Sample Input 2	Sample Output 2
12 3 2 3 4 5 6 7 4 7 8 8 12 12	2

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

This page is intentionally left (almost) blank.

Problem H
Clock Pictures
Problem ID: clockpictures

You have two pictures of an unusual kind of clock. The clock has n hands, each having the same length and no kind of marking whatsoever. Also, the numbers on the clock are so faded that you can't even tell anymore what direction is up in the picture. So the only thing that you see on the pictures, are n shades of the n hands, and nothing else.

You'd like to know if both images might have been taken at exactly the same time of the day, possibly with the camera rotated at different angles.

Task

Given the description of the two images, determine whether it is possible that these two pictures could be showing the same clock displaying the same time.

Input

The first line contains a single integer n ($2 \leq n \leq 200\,000$), the number of hands on the clock.

Each of the next two lines contains n integers a_i ($0 \leq a_i < 360\,000$), representing the angles of the hands of the clock on one of the images, in thousandths of a degree. The first line represents the position of the hands on the first image, whereas the second line corresponds to the second image. The number a_i denotes the angle between the recorded position of some hand and the upward direction in the image, measured clockwise. Angles of the same clock are distinct and are not given in any specific order.

Output

Output one line containing one word: `possible` if the clocks could be showing the same time, `impossible` otherwise.

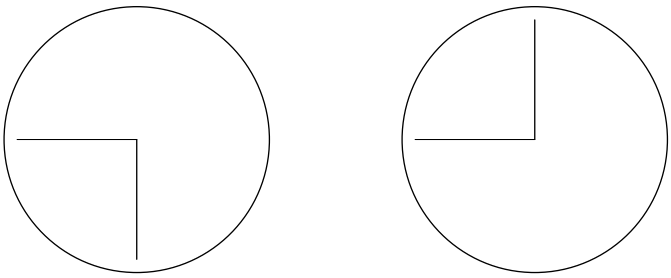


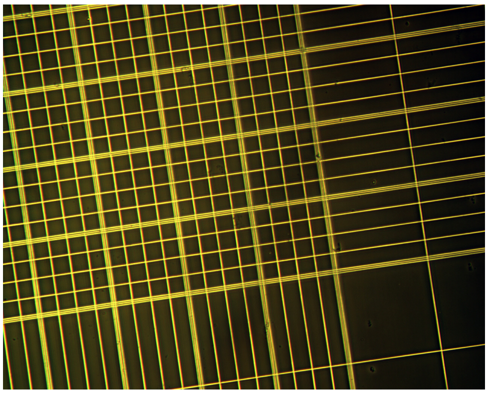
Figure H.1: Sample input 2

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

Sample Input 2	Sample Output 2
2 0 270000 180000 270000	possible

Sample Input 3	Sample Output 3
7 140 130 110 120 125 100 105 235 205 215 220 225 200 240	impossible

Problem I
How many squares?
Problem ID: squares



Hemocytometer, a device for counting blood cells; by Jeffrey M. Vinocur under CC BY 2.5

While browsing the internet, of course using Internet Explorer without any adblocker, you have noticed a number of interesting competitions advertised in the panels on various webpages. In most of these competitions you need to answer a simple question, like how many triangles/squares/rectangles there are in a picture, or even choose the right answer out of three possibilities. Despite the simplicity of the task, it seems that there are many valuable prizes to be won. So there is definitely something to compete for!

In order to increase your chances, you decided to write a simple program that will solve the problem for you. You decided to focus first on the question "How many squares are there in the picture?", and to simplify the problem even more, you assume that the input picture consists only of a number of lines that are infinite in both directions. To be precise, we say that four lines $\ell_1, \ell_2, \ell_3, \ell_4$ in the picture form a square if lines ℓ_1 and ℓ_3 are parallel to each other and perpendicular to ℓ_2 and ℓ_4, and moreover the distance between ℓ_1 and ℓ_3 is the same as the distance between ℓ_2 and ℓ_4.

Input

The first line of the input contains a single integer n ($1 \leq n \leq 2\,000$), denoting the number of lines in the input picture. Then follow n lines, each containing a description of one line in the input picture. The line is given as a pair of distinct points lying on it. That is, the description consists of four integers x_1, y_1, x_2, y_2, each of them of absolute value at most $10\,000$, such that the line passes through points (x_1, y_1) and (x_2, y_2). You may assume that points (x_1, y_1) and (x_2, y_2) are different, and also that all the lines in the picture are pairwise different.

Output

Output exactly one line with one integer, denoting the total number of squares formed by the lines in the picture.

Sample Input 1	Sample Output 1
10 0 0 1 0 0 1 1 1 0 2 2 2 0 0 0 4 1 -1 1 0 2 -2 2 2 1 1 2 2 1 1 0 2 3 1 2 2 1 3 0 2	6

Problem J
Road Work
Problem ID: roadwork

Image from Jocelyn Kinghorn, cropped.

Per is repairing roads. The job is concentrated on roads with one lane in each direction. Thus, when Per closes down the lane in one direction, all traffic has to go through the other lane. This is done by allowing only one direction of travel at any time. Per is often assigned the task of directing the traffic through this lane.

No car drives before being given a "go" signal from Per, and all the cars drive through the maintained segment at the same speed. Because there is only one lane, cars in one direction must leave the segment before cars in the other direction can enter. For safety reasons, cars driving in the same direction have to keep a distance of at least 3 seconds between each other.

For example, if cars `A` and `B` arrive at the west endpoint at second 10, Per can let them go at earliest second 10 and 13 in the order they arrived. If it, in this example, takes 8 seconds to pass and car `C` arrives at the east endpoint at second 17, then car `C` has to wait 4 seconds until Per lets it go at second 21.

There is a problem of drivers getting irritated with Per; they think they have to stop for too long. Per has been logging how long they can bear to wait before they get irritated. One day, to be able to evaluate his work, Per noted down when the cars arrived at the two endpoints of the segment. Per's question is the following: what is the least number of drivers that can be irritated? We assume that a driver gets irritated if the time between the moment he arrives at the maintained segment and the moment he is actually given the "go" exceeds his irritation time limit.

Input

The first line of the input contains two integers t and n ($4 \leq t \leq 180$ and $1 \leq n \leq 250$), where t is the time in seconds needed for a car to pass the segment under maintenance, and n is the total number of cars arriving at the segment. The following n lines describe the cars. The i-th line contains the description of the i-th car in the following format:

- one character d, being `W` for cars arriving at the west endpoint of the segment, and `E` for the ones that arrive at the east endpoint; and
- two integers a and r ($0 \leq a < 86\,400$ and $0 \leq r \leq 3\,600$), where a denotes the arrival time in seconds after midnight, and r denotes the time in seconds it takes for the driver to get irritated.

The cars arrive in the order specified in the input and they cannot overtake each other. In particular, a car whose driver is already irritated has to stay in the queue until eventually receiving the "go" and passing the maintained segment.

Output

Output one line with the least possible number of irritated drivers.

Sample Input 1	Sample Output 1
8 3 W 10 0 W 10 3 E 17 4	0

Sample Input 2	Sample Output 2
100 5 W 0 200 W 5 201 E 95 1111 E 95 1 E 95 11	1

Problem K
Train Passengers
Problem ID: trainpassengers

Photo by Elmer and Tenney

The Nordic Company of Passing Carriages is losing money at an alarming rate because most of their trains are empty. However, on some lines the passengers are complaining that they cannot fit in the cars and have to wait for the next train!

The authorities want to fix this situation. They asked their station masters to write down, for a given train, how many people left the train at their station, how many went in, and how many had to wait. Then they hired your company of highly paid consultants to assign properly sized trains to their routes.

You just received the measurements for a train, but before feeding them to your optimisation algorithm you remembered that they were collected on a snowy day, so any sensible station master would have preferred to stay inside their cabin and make up the numbers instead of going outside and counting.

Verify your hunch by checking whether the input is inconsistent, i.e., at every time the number of people in the train did not exceed the capacity nor was below 0 and no passenger waited in vain. The train should start and finish the journey empty, in particular passengers should not wait for the train at the last station.

Input

The first line contains two integers C and n ($2 \leq n \leq 100$), the total capacity and the number of stations the train stops in. The next n lines contain three integers each, the number of people that left the train, entered the train, and had to stay at a station. Lines are given in the same order as the train visits each station. All integers including C are between 0 and 10^9 inclusive.

Output

One line containing one word: `possible` if the measurements are consistent, `impossible` otherwise.

Sample Input 1

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

Sample Output 1

```
possible
```

Sample Input 2

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

Sample Output 2

```
impossible
```

Sample Input 3	Sample Output 3

`1 2`
`0 1 0`
`1 0 1`
	`impossible`

Sample Input 4	Sample Output 4

`1 2`
`0 1 1`
`0 0 0`
	`impossible`