4th Virginia Tech High School Programming Contest

Dec 9, 2017

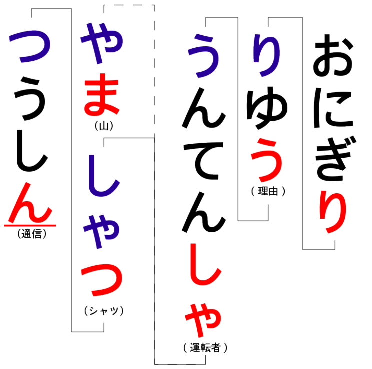
As a reminder, here are the key rules under which this contest is conducted:

- Teams may not communicate with another human during the contest about the problems.
- Teams may not use more than 1 computer.

Enjoy!

This page is intentionally left blank.

Problem A
Shiritori



The original version of Shiritori is played using Japanese hiragana, katakana, or kanji characters. Source WikiMedia

The Japanese game of Shiritori is the perfect game for a long car ride. The rules are simple: the first player picks any word to say, then the second player must choose a new word that begins with the last letter of the word that the first player just said. Your job is to determine if the game was played according to these rules, given a history of the words used in a particular game. In a game, player 1 always starts first.

Input

Input consists of one test case that begins with an integer N ($2 \leq N \leq 100\,000$) on a single line. Each of the following N lines contains 1 word. The words are presented in the order in which the players called them out, starting with player 1. All words consist of between 1 and 120 lowercase English letters.

Output

If the game was played according to the rules, output "`Fair Game`". Otherwise, find out which player first violated the rules of the game. That player lost the game, so output "`Player <i> lost`". For example, if player 1 violated the rules first, output "`Player 1 lost`".

Sample Input 1

```
5
apple
ear
real
letters
style
```

Sample Output 1

```
Fair Game
```

Sample Input 2

```
3
apple
extra
apple
```

Sample Output 2

```
Player 1 lost
```

Sample Input 3

```
2
apple
neat
```

Sample Output 3

```
Player 2 lost
```

Sample Input 4

```
5
apple
east
team
meat
team
```

Sample Output 4

```
Player 1 lost
```

Problem B
StopCard

Created with Imgflip GIF Generator

Jacob is playing a very odd solo card game called StopCard. In this game, the deck consists of n cards where every card has one unique integer written on one side of it. The deck is shuffled randomly before the game is played. During each turn of the game, Jacob can choose to either take a card off the top of the deck and read that value, or not take a card off the deck and end the game. His final score for the game is the value of the last card taken from the deck. The deck will always have at least one card, and the first turn of the game must always be to draw the top card of the deck.

Jacob has a basic understanding of optimal stopping theory, so he devises a plan to play the game somewhat effectively. His plan is to keep drawing cards for a predetermined number (c) of times, then to keep going until he sees a card that is larger than all the cards he previously saw, then to stop. His score will be the value written on that card. If he never sees a larger value on a card than the first cards he skipped, he would continue drawing until the deck runs out and would be left with a score equal to the number on the last card in the deck.

What is Jacob's expected score under this strategy?

Input

The input consists of a single test case. The first line contains two integer numbers n and c denoting the number of cards in the deck ($0 < n \leq 64$) and c is the number of cards Jacob will definitely draw ($0 \leq c < n$). The second line contains n distinct integer numbers a_i ($0 \leq a_i \leq 10\,000$) - the numbers written on the cards in the deck, which may be given in any order.

Output

Output the expected score under Jacob's strategy. Your answer will be considered correct if its absolute or relative error does not exceed 10^{-5}.

Sample Input 1	Sample Output 1

`2 1`
`0 1`
	`0.500000`

Sample Input 2	Sample Output 2

`4 2`
`0 4 8 6`
	`5.500000`

Sample Input 3

```
15 7
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
```

Sample Output 3

```
11.266667
```

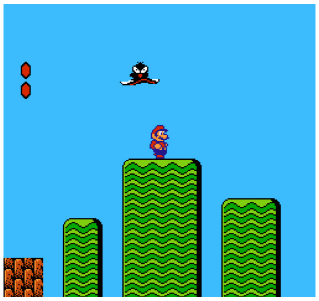
Sample Input 4

```
5 0
10000 9832 3242 2 42
```

Sample Output 4

```
4623.6
```

Problem C
Saving Princess Peach



Source: WikiMedia Commons

Mario is trying to save his beloved Princess Peach! However, in order to do that, Mario must jump over many obstacles in order to save Princess Peach. Thus, he makes a grand plan to infiltrate Bowser's castle.

But first, he needs to practice. His brother Luigi makes a practice course for Mario to train on. On this course, Mario practices looking for all the possible obstacles that could take away his life. But Mario is sloppy and misses some obstacles, counts some obstacles more than once, and generally screws up the order of the obstacles he does find when he lists them!

Write a program so that Luigi can tell his brother which obstacles he's missed!

Input

The first line contains 2 values. The first value N $(0 < N \leq 100)$ is the total number of obstacles. Obstacles are numbered $0 \ldots N - 1$.

The second value Y $(0 \leq Y \leq 200)$ represents how many obstacles Mario said he's found on his practice run. The next Y lines each list a single integer k $(0 \leq k < N)$ which is the number of an obstacle Mario says he's found.

Output

First, output the obstacles that Mario missed in increasing order, each on a separate line. On the last line, print `Mario got X of the dangerous obstacles.` where X is the number of distinct obstacles Mario found.

Sample Input 1

```
20 4
5
10
12
16
```

Sample Output 1

```
0
1
2
3
4
6
7
8
9
11
13
14
15
17
18
19
Mario got 4 of the dangerous obstacles.
```

Problem D
Swish

You and your friends are playing a card game called “Swish.” In Swish, each player tries to collect as many cards as possible from the table by collecting them in groups called “swishes.” Almost all the cards from the table have been collected so far, and so you wonder if there is a way to group the remaining cards into swishes so that there are no cards left over.

Each card has exactly 1 dot and 1 ring on it. The cards are transparent and rectangular, so you can place them on top of each other and see the rings and dots on the cards underneath. A ring can enclose a dot if they share the same position. Each dot/ring pair can be in one of exactly 12 positions on the card, which are numbered as shown in the Figure below:

Swish (TM) is published by ThinkFun.com

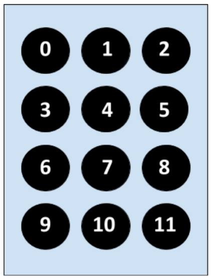


Figure D.1: Possible positions for rings and dots on each card.

A swish is formed by choosing an arbitrary card to be the start of a pile, then adding more cards to the pile so that a new card has a ring which encloses the previous card’s dot, and so that the dot of the added card does not overlap any ring/circle pairs already placed. Cards can be flipped and/or rotated in any way when they are being added to a swish, but the cards must lay exactly on top of each other and must maintain a “portrait” orientation (being higher than it is wide). The last card chosen must have a dot which is enclosed by the first card’s ring. By forming swishes in this way, a valid swish cannot contain a smaller swish if all the card orientations are kept the same. A valid “swish” can contain anywhere from 2 to 12 cards.

Is it possible to group all the cards from the table into swishes, and if so, what is the minimum number of swishes necessary to do so?

Input

The input consists of a single test case. The first line of input contains a single integer n, where $1 \leq n \leq 20$ is the number of cards on the table. The next n lines will describe each card. Each line will contain two integers r and d, where $0 \leq r, d \leq 11, r \neq d$ are the positions of the ring and dot on that card, respectively.

The positions of the rings and the dots are given in row order, as shown in Figure D.1. The positions are symmetric, so that a ring or a dot that is on position 0 may be rotated and/or flipped so that the position changes to $2, 9$, or 11. Similarly, a ring or dot on position 4 may be flipped and/or rotated so that the position changes to 7. The other positions may also be changed by rotating and/or flipping the card.

Output

If it is not possible to arrange the cards in swishes so that each card is a part of exactly one swish, then output `-1`. Otherwise, output an integer denoting the minimum number of swishes so that each card is a part of exactly 1 swish.

Sample Input 1

```
4
9 4
9 1
4 9
1 9
```

Sample Output 1

```
1
```

Sample Input 2

```
5
3 6
7 2
11 0
9 4
0 6
```

Sample Output 2

```
-1
```

Sample Input 3

```
12
6 5
4 0
7 11
5 8
1 6
10 4
11 9
8 9
2 8
3 4
0 1
2 10
```

Sample Output 3

```
1
```

Problem E
Through the Grapevine

Pieter Brueghel the Younger, Web Gallery of Art

According to Wikipedia, to hear something "through the grapevine" is to learn of something informally and unofficially by means of gossip or rumor. In this problem, you are tasked with determining how many people will hear about a particular rumor "through the grapevine" after a certain number of days.

Rumors are always started by a single person. On any given day, a person who knows the rumor can spread it by telling the people that they know. Upon hearing of the rumor, that person must wait until the following day before they can begin to spread it themselves. Furthermore, some people are skeptical and will only spread the rumor once they've heard it from a number of distinct sources. However once a person has heard the rumor from enough people, they will always try to spread the rumor to as many people as possible.

Input

The first line will contain three integers: $0 < n \leq 100\,000$, $0 < m \leq 100\,000$, and $0 \leq d \leq 10\,000$, where n is the number of people, m is the number of connections, and d is the number of days that elapse.

The next n lines will each consist of a unique string s and an integer $0 \leq t \leq 1000$ where s is the name of a person and t is their level of skepticism. In other words, person s must hear the rumor from t distinct other people before s will begin spreading the rumor.

This is followed by m lines each consisting of two strings u and v which indicates that person u and person v know each other. Each of these lines represents a unique pair of persons.

The final line will contain a single string r, the name of the person that the rumor originates from. Note that r is the only person with skepticism $t = 0$. All strings are between 1 and 20 characters long and consists only of letters and digits.

Output

Output a single integer: the number of people (not including person r) that have heard the rumor after d days.

Sample Input 1	Sample Output 1
3 2 1 Alice 0 Bob 1 Carol 1 Alice Bob Bob Carol Alice	1

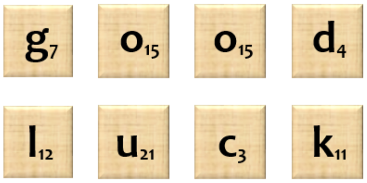
Sample Input 2

```
5 5 3
Alice 0
Bob 1
Carol 1
Dan 3
Erin 1
Alice Bob
Alice Carol
Bob Dan
Carol Dan
Dan Erin
Alice
```

Sample Output 2

```
3
```

Problem F
The Weight Of Words



Justin likes to play a game called “Weight of Words.” In this game, a player must find words of a given length and a given weight. The weight of a word is the sum of the weights of its letters. Letters are English lowercase characters from ‘a’ to ‘z’ and have weights from $1 \ldots 26$.

You are asked to write an AI that can play this game!

Input

The input consists of a single line with two integers l $(0 < l \leq 40)$ and w $(0 < w \leq 1000)$.

Output

Output a string of length l with weight w consisting of lowercase English letters if it exists, otherwise output `impossible`. If more than one string exists, you may output any of them.

Sample Input 1	Sample Output 1
`11 131`	`programming`

Sample Input 2	Sample Output 2
`7 96`	`contest`

Sample Input 3	Sample Output 3
`26 677`	`impossible`

This page is intentionally left blank.

Problem G
RA Duty Scheduler

Students hanging out with their resident advisors at Virginia Tech. Source: VT News

Resident Advisors (or RAs) at Virginia Tech are student leaders who are responsible for helping freshmen transition smoothly to college life. Mr. Friedel is trying to design a system to assist him in the process of scheduling Resident Advisors (RAs) to be on duty every month. On each day there are exactly 2 RAs on duty. Unfortunately for Mr. Friedel, each RA is available only on specific days and can only be scheduled then. Before the end of each month, each RA submits a list of days with their availability to Mr. Friedel.

Your goal is to help him write a program that will take a list of RAs and their availabilities and assign two RAs for duty on each day.

While there is no limit for how many days each RA can be on duty, Mr. Friedel wants to reduce unfairness by limiting the imbalance between the number of days each RA is assigned.

Towards that goal, your program needs to minimize the maximum number of days to which any RAs may be assigned.

Input

The input consists of a single test case. The first line contains 2 numbers m ($2 \leq m \leq 60$) and n ($28 \leq n \leq 31$) where m is the number of RAs and n is the total number of days this month. This is followed by m lines where each line consists of the name of the RA followed by an integer d ($1 \leq d \leq n$) denoting the number of days this RA is available. The remainder of the line consists of d unique integers i ($1 \leq i \leq n$) denoting the days on which this RA is available. Each RA's name is a string between 1 and 30 characters consisting only of letters. You may assume that there is at least one possible assignments of RAs!

Output

On the first line, output the maximum number of days on which any RA is assigned. Following that, you should output n lines starting with `Day k:` where k is the number of the day ranging from 1 to n (inclusive and in order), followed by a space, followed by the names of two RAs scheduled for that day, separated by spaces. The two RAs scheduled for a given day can be listed in any order.

If there are multiple valid assignments, you may output any of them!

Sample Input 1

```
20 30
Katrina 11 1 3 4 8 10 12 13 15 17 27 29
Pawl 13 3 4 5 6 8 10 11 12 13 15 17 26 27
Sydney 14 1 2 3 4 6 7 8 11 13 14 15 16 28 30
Kylie 15 1 2 4 5 6 8 9 10 12 13 15 16 17 18 27
Meredith 15 1 2 4 7 8 9 11 12 13 25 26 27 28 29 30
Amanda 16 1 2 7 8 9 10 11 14 15 16 17 18 19 28 29 30
Akshay 16 1 2 3 4 6 7 8 12 13 14 15 16 17 28 29 30
Chelsea 17 1 2 3 4 5 7 8 9 11 15 16 17 25 27 28 29 30
Zachary 18 1 3 4 5 6 7 8 9 12 13 14 15 16 17 26 27 29 30
Alex 19 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 17 28 29 30
Tariq 19 1 3 4 5 6 8 10 11 12 13 18 19 20 22 24 25 26 27 29
Schlake 21 1 3 4 6 10 11 12 13 15 17 18 19 20 21 22 23 24 25 26 27 29
Joel 23 3 4 5 6 7 8 9 10 11 12 13 14 18 19 20 21 22 23 24 25 26 27 28
Benjamin 23 1 2 3 4 5 6 7 8 9 10 12 13 14 15 18 19 20 21 22 23 24 28 30
Collin 24 1 2 3 4 5 6 7 8 9 10 11 13 14 15 16 17 18 19 20 21 27 28 29 30
Austin 24 1 2 3 5 6 7 8 9 10 12 13 14 15 16 17 18 19 20 21 22 23 24 28 29
Shahmir 24 1 2 3 4 6 8 12 13 14 15 16 17 19 20 21 22 23 24 25 26 27 28 29 30
Harrison 25 1 2 3 6 7 8 9 10 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30
Josh 27 2 3 4 5 6 7 8 9 10 11 13 14 15 16 17 18 20 21 22 23 24 25 26 27 28 29 30
Amy 28 2 3 4 5 6 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30
```

Sample Output 1

```
3
Day 1: Benjamin Katrina
Day 2: Amanda Alex
Day 3: Alex Katrina
Day 4: Akshay Zachary
Day 5: Collin Chelsea
Day 6: Collin Schlake
Day 7: Akshay Sydney
Day 8: Pawl Meredith
Day 9: Joel Kylie
Day 10: Alex Amanda
Day 11: Josh Meredith
Day 12: Akshay Meredith
Day 13: Schlake Sydney
Day 14: Shahmir Amy
Day 15: Kylie Katrina
Day 16: Sydney Amanda
Day 17: Amy Harrison
Day 18: Collin Kylie
Day 19: Harrison Tariq
Day 20: Austin Benjamin
Day 21: Amy Austin
Day 22: Tariq Shahmir
Day 23: Austin Joel
Day 24: Benjamin Josh
Day 25: Joel Tariq
Day 26: Schlake Pawl
Day 27: Harrison Pawl
Day 28: Shahmir Josh
Day 29: Chelsea Zachary
Day 30: Zachary Chelsea
```

Problem H
Coconut Splat

Source: WikiMedia

Coconut Splat is one of Theta's favorite counting-out games. It goes like this: initially, all players stand in a circle with their hands folded together (like an intact coconut). In clockwise order, one player touches the hands of the other players and says the rhyme: "Co-co-nut, Co-co-nut, Co-co-nut, Splat!" At each syllable, the player touches a hand, and the player whose hand or hands is touched last takes one of the following actions:

- If the player's hands are still folded, they are split into two fists (the coconut is cracked in two halves). The next round starts with the fist that is the first half of the coconut, then the second half, then going to the next player.
- If a fist is touched last, the hand is turned palm down (the milk spills out). The next round starts with the next hand in clockwise order, which could be the other hand of the same player, or it could be the hand or folded hands belonging to the next player.
- If a hand that is already turned palm down is touched last, the player to whom it belongs puts the hand behind their back and this hand won't be counted in the following rounds. The next round starts with the next hand in clockwise order as in the previous case.
- If a player has put both of their hands behind their back, that player is out of the game. The game ends when there is only one player left.

The hand or hands of the player doing the counting are taken into account (for instance, the counting player touches their thigh when it would be her turn to be touched).

There are variations of this game, for instance, some kids say "Coconut, coconut, crack your nut!" instead, which has only 9 instead of 10 syllables.

There are n players, and counting always starts with the folded hands of player 1. For instance, in the first round, if the rhyme has 3 syllables, player 3 would be the one to first crack their coconut into two fists.

Write a program that determines the winner of the counting-out game based on the number of players and based on the number of syllables in the rhyme that is used!

Input

The input consists of a single test case with two numbers s ($0 < s \leq 100$) and n ($2 \leq n \leq 100$) denoting the number of syllables in the rhyme and the number of players, respectively.

Output

Output a single integer p ($1 \leq p \leq n$), the number of the player who is left.

Sample Input 1

```
10 2
```

Sample Output 1

```
2
```

Sample Input 2

```
10 10
```

Sample Output 2

```
7
```

Sample Input 3

```
1 2
```

Sample Output 3

```
2
```

Problem I
Alien Codebreaking

You've intercepted encrypted communications between Martian diplomats. Since Martian diplomats are often spies, you decide to decrypt the messages. While the Martians have skilled rocket tech, they lag behind in number theory considerably, which compromises their encryption protocol.

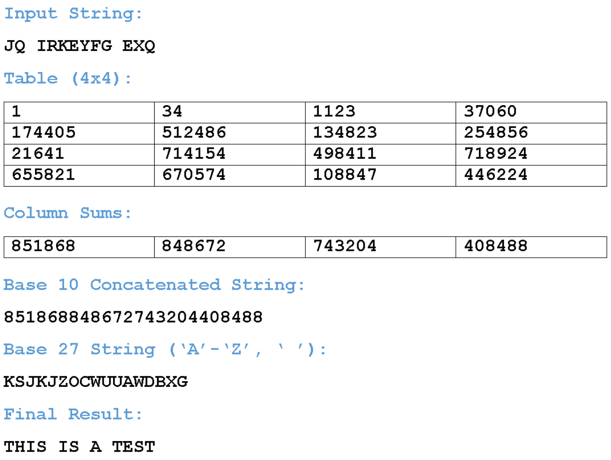
Fortunately for you, spies friendly to you have reverse engineered the Martian protocol. It turns out that the Martians are using a shift-based cipher combined with a very long one-time pad. More specifically, the decryption procedure works as follows:

Step 1: Define the function $f(x) = (33x + 1) \mod 2^{20}$.

Further define $f^1(x) = f(x)$, $f^2(x) = f(f(x))$, $f^3(x) = f(f(f(x)))$, and so on.

Step 2: Create a X by X size grid, fill the upper left corner with $f^1(0)$, the next cell to the right with $f^2(0)$, $f^3(0)$ etc. Once the top row is filled, continue to the cell below the upper left cell, and fill with $f^{X+1}(0)$. Continue this process until all rows are filled.

Step 3: Sum all the values in every column, and take those values mod 2^{20}.



Input String:

JQ IRKEYFG EXQ

Table (4x4):

1	34	1123	37060
174405	512486	134823	254856
21641	714154	498411	718924
655821	670574	108847	446224

Column Sums:

851868	848672	743204	408488

Base 10 Concatenated String:

851868848672743204408488

Base 27 String (‘A’-‘Z’, ‘ ’):

KSJKJZOCWUUAWDBXG

Final Result:

THIS IS A TEST

Step 4: Concatenate the base-10 representations of the column sums together, to get a very long base-10 number. For instance, if you had column sums of `10` and `12` for the first and second column, the leftmost four digits of the resulting value would be `1012`.

Step 5: Convert the result of step 4 from base 10 to base 27. This will yield the one-time pad the Martians used.

Step 6: For each letter l of the intercepted message, shift the letter by the amount given by the corresponding digit of step 5, base 27.

You know that the both the encrypted and the decrypted message consist of only uppercase English characters ‘`A`’ through ‘`Z`‘ and spaces, which are assigned values $0 \ldots 26$. (`A = 0`, `B = 1`, ... `Z = 25`, `SPACE = 26`. Shifting means to add the digit at the corresponding position of the pad to the value of the letter in the encrypted message. For instance, if the encrypted message has letter '`D`' in position 3, and the 3rd base-27 digit of the pad is 25, the decrypted letter would be $3 + 25 = 1 \mod 27$ which is ‘`B`’.

Step 7: Output the decrypted message.

Input

The first line of the input contains two positive integers, N $(1 \leq N \leq 10^6)$, and X $(1 \leq X \leq 2.5 * 10^5)$. It is guaranteed that the base 27 result of step 5 will be longer or equal to the length of the intercepted message. The second line of the input contains a string consisting of uppercase letters and spaces of length N, the encrypted text.

Output

Output the decrypted text.

Sample Input 1

```
14 4
JQ IRKEYFG EXQ
```

Sample Output 1

```
THIS IS A TEST
```

Sample Input 2

```
43 100000
BLNAMOTPRRNIXRNMPIWHXDZTRQJXRKIAIEEIIPJLGZP
```

Sample Output 2

```
FRIENDS ROMANS COUNTRYMEN LEND ME YOUR EARS
```

Problem J
Traveling Monk

The following puzzle was popularized by Martin Gardner's book "My Best Mathematical and Logic Puzzles," although it appears first in the monograph "On Problem-Solving" by the Gestalt psychologist Karl Dunker.

Source: pixabay cc0

> *One morning, exactly at sunrise, a Buddhist monk began to climb a tall mountain. The narrow path, no more than a foot or two wide, spiraled around the mountain to a glittering temple at the summit.*
>
> *The monk ascended the path at varying rates of speed, stopping many times along the way to rest and eat the dried fruit he carried with him. He reached the temple shortly before sunset. After several days of fasting and meditation he began his journey back along the same path, starting at sunrise and again walking at variable speeds with many pauses along the way. His average speed descending was, of course, greater than his average climbing speed.*
>
> *Prove that there is a spot along the path that the monk will occupy on both trips at precisely the same time of day!*

You can probably see why this is true - but can you write a program that computes the time at which the monk will be at the same spot during his ascent and descent?

Input

The input consists of a single test case. The first line contains two integers a $(0 < a \leq 5\,000)$ and d $(0 < d \leq 5\,000)$ denoting the number of segments during the ascent and descent, respectively. This is followed by a lines, each containing two integers h $(0 \leq h \leq 1\,000)$ and t $(0 < t \leq 100)$ denoting the positive change in elevation (h) the monk gained during this segment and the time it took (t). If the monk rests and eats during a segment, h will be 0.

This is followed by d lines, each containing two integers h $(0 \leq h \leq 1\,000)$ and t $(0 < t \leq 100)$ denoting the change in elevation (h) the monk descended during this segment and the time it took (t). If the monk rests and eats during a segment, h will be 0.

Output

Output a single floating point number, the earliest point in time at which the monk occupies the same spot during his climb and his descent. The monk starts his ascent and his descent at time 0 on both days.

Your answer will be considered correct if its absolute or relative error does not exceed 10^{-5}.

Sample Input 1

```
1 1
10 11
10 10
```

Sample Output 1

```
5.238095
```

Sample Input 2

```
3 1
4 2
0 3
6 3
10 7
```

Sample Output 2

```
4.200000
```

Sample Input 3

```
3 3
2 3
0 5
3 1
3 4
0 2
2 2
```

Sample Output 3

```
4.000000
```

Problem K
Provinces and Gold

Jake is learning how to play the card game Dominion. In Dominion, you can buy a variety of treasure, action, and victory point cards - at the end of the game, the player with the most victory points wins!

Each turn, each player draws 5 cards and can use their action and treasure cards to obtain buying power in order to buy more cards. Since Jake is just starting out, he's decided to buy only treasure and victory point cards.

This means the cards he can buy are:

Province (costs 8, worth 6 victory points)
Duchy (costs 5, worth 3 victory points)
Estate (costs 2, worth 1 victory point)

Gold (costs 6, worth 3 buying power)
Silver (costs 3, worth 2 buying power)
Copper (costs 0, worth 1 buying power)

At the start of Jake's turn, he draws 5 of these cards. Given the number of Golds, Silvers, and Coppers in Jake's hand, calculate the best victory card and best treasure card he could buy that turn. Note that Jake can buy only one card.

Input

The input consists of a single test case on a single line, which contains three non-negative integers G, S, C $(G + S + C \leq 5)$ indicating the number of Golds, Silvers, and Coppers Jake draws in his hand.

Output

Output the best victory card and the best treasure card Jake can buy this turn, separated with `" or "`, in this order. If Jake cannot afford any victory cards, output only the best treasure card he can buy.

Sample Explanation

In Sample Input 1, Jake has 1 Silver in his hand, which means he has 2 buying power. This would allow him to either buy an Estate or a Copper.

Sample Input 1	Sample Output 1
`0 1 0`	`Estate or Copper`

Sample Input 2

```
2 1 0
```

Sample Output 2

```
Province or Gold
```

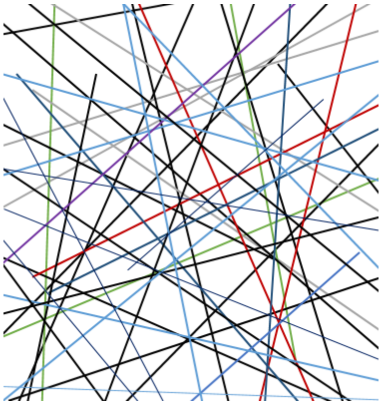
Sample Input 3

```
0 0 1
```

Sample Output 3

```
Copper
```

Problem L
Criss-Cross



Peter is bored during his operating systems class, so he draws doodles on a sheet of paper. He feels like drawing abstract art using his ruler: he draws line segments by choosing two points in the plane and connecting them. Lots of them.

Can you write a program that counts the number of distinct points at which the line segments he drew intersect or touch?

Input

The first line in the input contains an integer n ($1 \leq n \leq 1\,000$) which is the number of lines. The following n lines contain four integers x_0 y_0 x_1 y_1 ($-1\,000\,000 \leq x_0,\ y_0,\ x_1,\ y_1 \leq 1\,000\,000$. Lines have non-zero length, i.e., the two points will be distinct: $x_0 \neq x_1$ or $y_0 \neq y_1$ or both.

Output

Output the number of distinct points for which there is at least one pair of line segments that intersects or touches at this point.

Sample Input 1

```
3
1 3 9 5
2 2 6 8
4 8 9 3
```

Sample Output 1

```
3
```

Sample Input 2

```
3
5 2 7 10
7 4 4 10
2 4 10 8
```

Sample Output 2

```
1
```

Sample Input 3

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

Sample Output 3

```
1
```

This page is intentionally left blank.