Tehran Site Qualifications 2018 - Round 1

Isfahan University of Technology

November 2018

Organizers:

Sponsors:

A. AI the Redeemer

In a far far land called IUT (Issac United Territories), There is a king who has ruled over the kingdom for decades. Due to his numerous mistakes, The people now live in misery.

"Someday, Someone will come along and will put an end to the poverty and misery", People say. The king got mad once his council told him that "The Redeemer is in fact not a human, He is a robot, an artificial intelligence!". So the mad king ordered to destroy any AI in the world; "Burn them all", He said.

Your job is to find that how many cities in the kingdom do not have an AI to rescue the city. Don't let the people down.

Input

The first line of input will contain a single integer n ($1 \leq n \leq 100$), the number of cities in the kingdom. Then follow n lines each with a sequence of k_i ($1 \leq k_i \leq 1000$) characters, each of which are either 'A', 'I' or 'X'.

Output

Output the number of cities with no AIs, assuming that an AI lives where an 'A' is followed by an 'I' immediately.

Sample Input

```
3
IAXXX
IXIXAI
AXI
```

Sample Output

```
2
```

B. Caravan Robbers

Long ago there were two great cities named "Isfahan" and "Sedeh" and The Great Caravan Road between them. Many robber gangs "worked" on that road.

By an old custom, the i-th band robbed all merchants that dared to travel between a_i and b_i miles of The Great Caravan Road. The custom was old, but a clever one, as there were no two distinct i and j such that $a_i \le a_j$ and $b_j \le b_i$. Still when intervals controlled by two gangs intersected, bloody fights erupted occasionally. Gang leaders decided to end those wars. They decided to assign each gang a new interval such that all new intervals do not intersect (to avoid bloodshed), for each gang their new interval is subinterval of the old one (to respect the old custom), and all new intervals are of equal length (to keep things fair).

You are hired to compute the maximal possible length of an interval that each gang would control after redistribution.

Input

The first line contains n ($1 \le n \le 100\ 000$) - the number of gangs. Each of the next n lines contains information about one of the gangs - two integer numbers a_i and b_i ($0 \le a_i \le b_i \le 1\ 000\ 000$). Data provided in the input file conforms to the conditions laid out in the problem statement.

Output

Output the maximal possible length of an interval in miles as an irreducible fraction p/q.

Sample Input

```
3
2 6
1 4
8 12
```

Sample Output

```
5/2
```

C. Mega Inversions

The n^2 upper bound for any sorting algorithm is easy to obtain: just take two elements that are misplaced with respect to each other and swap them. Alireza conceived an algorithm that proceeds by taking not two, but three misplaced elements. That is, take three elements $a_i > a_j > a_k$ with $i < j < k$ and place them in order a_k, a_j, a_i. Now if for the original algorithm the steps are bounded by the maximum number of inversions $\frac{n(n-1)}{2}$, Alireza is at his wits' end as to the upper bound for such triples in a given sequence. He asks you to write a program that counts the number of such triples.

Input

The first line of the input is the length of the sequence, $1 \leq n \leq 10^5$. The next line contains the integer sequence $a_1, a_2, \ldots, a_n$. You can assume that all $a_i \in [1, n]$.

Output

Output the number of inverted triples.

Sample Input 1

```
3
1 2 3
```

Sample Output 1

```
0
```

Sample Input 2

```
4
3 3 2 1
```

Sample Output 2

```
2
```

D. The Elevator to the Rahmena College

You are on your way to the Rahnema college to present your final project presentation, and you are already late. The presentation is in a skyscraper and you are currently in floor s, where you see an elevator. Upon entering the elevator, you learn that it has only two buttons, marked "UP u" and "DOWN d". You conclude that the UP-button takes the elevator u floors up (if there aren't enough floors, pressing the UP-button does nothing, or at least so you assume), whereas the DOWN-button takes you d floors down (or none if there aren't enough). Knowing that the presentation will be at floor g, and that there are only f floors in the building, you quickly decide to write a program that gives you the amount of button pushes you need to perform. If you simply cannot reach the correct floor, your program halts with the message "`use the stairs`".

Given input f, s, g, u and d (floors, start, goal, up, down), find the shortest sequence of button presses you must press in order to get from s to g, given a building of f floors, or output "`use the stairs`" if you cannot get from s to g by the given elevator.

Input

The input will consist of one line, namely `f s g u d`, where $1 \leq s, g \leq f \leq 1000\,000$ and $0 \leq u, d \leq 1000\,000$. The floors are one-indexed, i.e. if there are 10 floors, s and g will be in $[1, 10]$.

Output

You must reply with the minimum numbers of pushes you must make in order to get from s to g, or output use the stairs if it is impossible given the configuration of the elevator.

Sample Input 1

```
10 1 10 2 1
```

Sample Output 1

```
6
```

Sample Input 2

```
100 2 1 1 0
```

Sample Output 2

```
use the stairs
```

E. Private Space

People are going to the movies in groups (or alone), but normally only care to socialize within that group. Being Iranian, each group of people would like to sit at least one space apart from any other group of people to ensure their privacy, unless of course they sit at the end of a row. The number of seats per row in the cinema starts at X and decreases with one seat per row (down to a number of 1 seat per row). The number of groups of varying sizes is given as a vector $(N_1, \ldots, N_n)$, where N_1 is the number of people going alone (poor singles!), N_2 is the number of people going as a pair, etc. Calculate the seat-width, X, of the widest row, which will create a solution that seats all (groups of) visitors using as few rows of seats as possible. The cinema also has a limited capacity, so the widest row may not exceed 12 seats.

Input

The first line of input contains a single integer n ($1 \leq n \leq 12$), giving the size of the largest group in the test case. Then follows a line with n integers, the i-th integer (1-indexed) denoting the number of groups of i persons who need to be seated.

Output

A single number; the size of the smallest widest row that will accommodate all the guests. If this number is greater than 12, output "`impossible`" instead (without quotes).

Sample Input 1

```
3
0 1 1
```

Sample Output 1

```
3
```

Sample Input 2

```
3
2 1 1
```

Sample Output 2

```
4
```

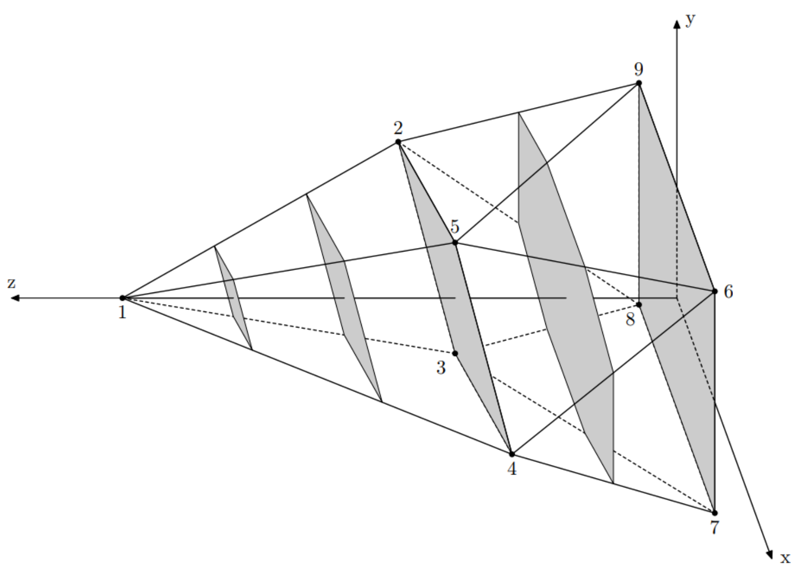
F. Aerodynamics

Maryam is working in a secret laboratory. She is developing missiles for national security projects. Maryam is the head of the aerodynamics department.

One surprising fact of aerodynamics is called Whitcomb area rule. An object flying at high-subsonic speeds develops local supersonic airflows and the resulting shock waves create the effect called wave drag. Wave drag does not depend on the exact form of the object, but rather on its *cross-sectional profile*.

Consider a coordinate system with OZ axis pointing in the direction of object's motion. Denote the area of a section of the object by a plane $z = z_0$ as $S(z_0)$. Cross-sectional profile of the object is a function S that maps z_0 to $S(z_0)$. There is a perfect aerodynamic shape called Sears-Haack body. The closer cross-sectional profile of an object to the cross-sectional profile of Sears-Haack body, the less wave drag it introduces. That is an essence of Whitcomb area rule.

Maryam's department makes a lot of computer simulations to study missile's aerodynamic properties before it is even built. To approximate missile's cross-sectional profile one takes samples of $S(z_0)$ for integer arguments z_0 from z_{min} to z_{max}.



Your task is to find the area $S(z_0)$ for each integer z_0 from z_{min} to z_{max}, inclusive, given the description of the missile. The description of the missile is given to you as a set of points. The missile is the minimal convex solid containing all the given points. It is guaranteed that there are four points that do not belong to the same plane.

Input

The first line of the input file contains three integer numbers: n, z_{min} and z_{max} ($4 \le n \le 100$, $0 \le z_{min} \le z_{max} \le 100$). The following n lines contain three integer numbers each: x, y, and z coordinates of the given points. All coordinates do not exceed 100 by their absolute values. No two points coincide. There are four points that do not belong to the same plane.

Output

For each integer z_0 from z_{min} to z_{max}, inclusive, output one floating point number: the area $S(z_0)$. The area must be precise to at least 5 digits after decimal point.

Sample Input

```
9 0 5
0 0 5
-3 0 2
0 -1 2
3 0 2
0 1 2
2 2 0
2 -2 0
-2 -2 0
-2 2 0
```

Sample Output

```
16.00000
14.92000
10.08000
4.48000
1.12000
0.00000
```

G. Enemy Division

Captain Mehran has to make a difficult decision. It is year 2118 and there is a big war in the world. His soldiers have been together since the war started, two years ago, and some of them have become enemies. Luckily, each soldier has at most 3 enemies.

They need to attack another country soon, and Mehran is worried that soldiers who are enemies might not cooperate well during the battle. He has decided to divide them into groups such that every soldier has at most one enemy in his group. He also wants to make it simple, so he wants to use as few groups as possible. Can you divide the soldiers into groups for him?

Input

On the first line there are two integers n and $m, 2 \leq n \leq 100\ 000, 0 \leq m \leq \frac{3n}{2}$, where n is the number of soldiers and m is the number of enemy pairs. Then follow m lines, each containing two space separated integers a_i, b_i, denoting that soldiers a_i and b_i are enemies, where $1 \leq a_i < b_i \leq n$. You can assume that all soldiers have at most 3 enemies

Output

The first line of output contains the minimal number of groups of soldiers k. Each of the next k lines contains a space separated list of a soldiers in a unique group.

Sample Input

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

Sample Output

```
2
1 3
2 4
```

H. Odd, Even, Imaginary!

Odd and Even have had their share of fun times playing the good old prime game:

They start with an arbitrary natural number, and take turns either adding 1 or dividing by a prime (assuming the result is still a natural number), and the one to reach 1 is the winner.

However, now that they have a new friend, Imaginary, they have decided to expand the rules of the game to allow for three-player action:

Instead of determining a winner for each round of play, they instead score points; the lowest number each of them has claimed during the round is the amount of points they get. (If any of them did not have the opportunity to claim any numbers, the starting number will be their score for that round.) At the end of the day, the player with the fewest points wins. And to avoid bad blood between themselves, they have all agreed that each of them only will focus on minimizing their own scores, and that whenever a player can choose different numbers that will result in the same score, that player will choose the lowest of those numbers. They have also agreed on a fixed order of play: Odd → Even → Ingmariay → Odd → ... , but they alternate who gets to start.

You recently missed one of their exciting evenings of play, because you had to make problems for the ICPC event. Fortunately for you, they had recorded the numbers and starting players for each round, and told you that since they always play optimally, you could use this to simulate the event for yourself. Oh joy!

As an example round, assume that Even is chosen as the starting player, and with the starting number 15. Then Even claims 16, Imaginary 8, Odd 4, Even 2 and Imaginary 1. Odd gets 4 points, Even 2 and Imaginary 1.

Input

The first line of input contains a single integer n ($1 \leq n \leq 1000$), the number of rounds they played that evening. Then follow n lines each beginning with the first character of the name of the starting player (either 'O', 'E' or 'I'), followed by a space and then the starting number for that round, in the range $[1, 10000]$.

Note: If the starting number is 1, all players receive 1 point for that round.

Output

Output a single line with the score at the end of the day for each of the three contestants, in the order "Odd", "Even", "Imaginary".

Sample Input 1

```
1
O 4
```

Sample Output 1

```
2 1 4
```

Sample Input 2

```
3
O 13
I 14
E 15
```

Sample Output 2

```
6 29 16
```

I. The Ring

Isfahan University of Technology (Hamoon Posht-e-Koohe Khodemoon!) consists of what was once a medieval town with narrow winding streets completely surrounded by a high wall protecting the town against the wild animals! The wall has since been removed and replaced by a system of interconnecting roads completely circumscribing the old part of the university. The roads inside still remains more or less the same as it was in the middle ages, which of course comes in conflict with modern requirements for accessibility by car, resulting in a maze of twisty little one-way streets, all alike, mixed with slightly wider two-way streets.

Making changes to the traffic routes in such a town can easily cause unexpected side effects if you do not plan carefully ahead. The story goes that a prominent member of the university council once submitted a proposal to the council regarding extensive changes to how the traffic should be organized in the town center. The proposal did have the merit that it would be very easy to drive in to the central square, but it would unfortunately also be impossible to drive out again. The council member in question later went on to become minister of justice in the town under the parole that society should be harder on criminals - "it should be easy to go to jail, but difficult to get out again".

To avoid mistakes as the one above, the town planners need you to develop a tool that can help them discover any traffic problems in the planning stage. The planners need to be alerted of two different situations. The first situation is that a street exists in the town center from which you cannot reach the surrounding, circular, system of roads, i.e., you are *trapped* inside the town. The second situation is that a street exists in the town that cannot be reached from the surrounding system of roads, i.e., it is *unreachable*.

Input

The input consists of a description of how streets connect to each other and the surrounding circular road system. Each street (or a segment of a street) within the town center is represented by an arbitrary integer id number ($0 < id < 1000$). The surrounding circular road system is represented by the special id number 0.

First line: An integer giving the number of streets (including the surrounding road system, $0 < streets \leq 1000$).

The following lines: One line for each street (no particular order required and the surrounding road system is included) consisting of a number of integers. First an integer giving the id number of the street. Second, the number of (other) streets that can be reached from this street. Third, a sequence of street id numbers indicating which streets can be reached from this street.

Output

One line for each street on which you would be trapped within the city consisting of the text "`TRAPPED X`" where "X" is replaced by street id number in question.

Then, one line for each street within the town that is unreachable from the surrounding system of roads consisting of the text "`UNREACHABLE X`" where X should be replaced by the street id in question.

If no problems are found, i.e., you are not trapped in any street and every street is reachable, you should print a single line containing the text "`NO PROBLEMS`".

If multiple streets cause you to get trapped - or are unreachable - you should list them in the same order they were entered in the input (within respective category).

Sample Input 1

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

Sample Output 1

```
TRAPPED 3
UNREACHABLE 4
UNREACHABLE 5
```

Sample Input 2

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

Sample Output 2

```
NO PROBLEMS
```