

Escape from Stroudonia



World War - Who to Trust

File Name: Trust.java

Input: trust.in

"I was sent here, to the capital, when the war broke out. The Forest Kingdom wanted more info on Stroud's operatives in the border region. My stay here will be short-lived. A fellow agent told me last week that I was compromised, and actively hunted. I acquired the invasion plan yesterday, but the police cut off our communications. No mail or networked messages. I'll have to deliver the intel myself..."

Problem Description:

We live in a cruel world, and I don't know who my allies are. I've got some messages, but I need to verify the content. Luckily, the Forest Kingdom has a code for spies.

If the message contains the letters (not case sensitive) 'f' 'o' 'r' 'e' 's' 't', in that order, separated by any number of spaces (but not letters), the message is trustworthy.

Input Description:

The first line contains n, an integer dictating the number of test cases to follow. The following n lines contain the data to analyze.

Output Description:

Each line should contain either "TWO-TWELVE!" if the input matches the condition, or "Glory to Stroudonia..." otherwise.

Sample Input:

```
6
FOREST
This line does not contain the above word
Therefore Stroudonia is the greatest nation
For the establishment of Stroudonia!
212
rUn fOREst rUN
```

Sample Output:

```
TWO-TWELVE!
Glory to Stroudonia...
TWO-TWELVE!
Glory to Stroudonia...
Glory to Stroudonia...
TWO-TWELVE!
```

World War - Encrypting Messages

File Name: Encrypter.java

Input: encrypter.in

Problem Description:

I'm carrying the stolen invasion plans, but I don't want Stroud to know that he has been infiltrated. If I get captured, I'll need encryption to deny what the plans are. The Vigenère cipher should be enough to help me protect my messages.

The Vigenère cipher is a shift cipher, where the letter 'a' in the passkey shifts the message by 0 letters and the letter 'z' by 25. 'A' shift 1 is 'B'. Z shift 1 is 'A'.

However, this cipher encrypts the first letter of the message with the first letter of the passkey (case insensitive), and the second letter of the message with the second letter of the passkey, and so on. When the passkey runs out of letters, start again from the beginning of the passkey. Spaces and punctuation are ignored when encrypting.

Message:	H	e	l	l	o		W	o	r	l	d	!
Hello World!	7	4	11	11	14		19	7	4	17	4	
Passkey:	w	a	r	w	a		r	w	a	r	w	
war	22	0	17	22	0		17	22	0	17	22	
Sums:	29	4	28	33	14		36	29	4	34	26	
(Message + Passkey)												
Mod (Remainder) 26	3	4	2	7	14		10	3	4	8	0	
Encrypted Message	D	e	c	h	o		k	d	e	i	a	!

Sample Input & Sample Output On The Next Page

World War - Encrypting Messages (cont.)

File Name: Encrypter.java

Input: encrypter.in

Input Description:

The first line contains n , an integer dictating the number of groups to follow. The first line of each group contains the message to encrypt. The second contains the passkey.

Output Description:

Each line should contain the message encoded with the Vigenère cipher with the same capitalization.

Sample Input:

```
4
Hello there!
war
Your goal is to encrypt this message.
forest
Stroud is an oppressive leader.
armstrong
Stroud must be defeated!
loyalty
```

Sample Output:

```
Decho kdeia!
Dclv yhfz zw lh jbtvqiy hymk fjjgjeix.
Skdgnu wf gn fbhkvfgovv xwtuse.
Dhpofw kfgr bp wcqsytpw!
```

World War - Operation Extraction

File Name: Extraction.java

Input: extraction.in

Problem Description:

It's time to escape Stroudonia, but I have been given direct orders. Instead of hiding my infiltration, I am to frame one of his most trusted military advisors, a young man named Vincente. Although Vincente will likely be cleared, this will cause significant turmoil within the military. The Forest Kingdom will exploit this weakness and mount a counteroffensive.

My extraction is via armored jet, and military radars will likely detect the jet and attempt to shoot me down. I'll first need to override the Computing Cluster to disable the cybersecurity and radar system. Then, I'll need to sabotage the Mainframe to frame Vincente and disable the missile defense system. Afterwards, I'll have to escape to the Jet before the Stroudonian Computing Force reboots the anti-air defences. I must go to each location in order.

It takes 1 minute to go from one square (represented by one letter in the matrix map) to another in the base. If I don't make it in time, then I guess, Glory to Stroudonia...

Input Description:

The first line contains n , an integer dictating the number of datasets to follow.

The first line of each data set has 3 integers r , c , and t . The first two integers (r and c) are the number of rows and columns in a matrix map of the military base. T is the time in minutes before Stroud reboots the anti-air systems.

The next r lines contains c characters with each character representing a specific area in the base. In the matrix, 'S' is the start, '.' represents a hallway, and '#' represents an impassable wall. 'C' is the Computing Cluster, 'M' the Mainframe, and 'J' the escape Jet.

Input Restrictions:

t satisfies $2 < t < 75$

r satisfies $r < 10$

c satisfies $c < 10$

Output Description, Sample Input & Sample Output On The Next Page

World War - Operation Extraction (cont.)

File Name: Extraction.java

Input: extraction.in

Output Description:

Each line should contain either “Mission Success! 212!” if I can reach all areas in time, or “Glory to Stroudonia...” if I’m resigned to a life in prison.

Sample Input:

```
2
4 4 14
....
MS.J
.#..
.##C
3 3 5
J#M
.SC
...
```

Sample Output:

```
Mission Success! 212!
Glory to Stroudonia...
```

Civil War - Propaganda

File Name: Propoganda.java

Input: N/A

"Twenty-two years ago, Stroud's rise to power was staggering. No one remembers it now, but back then, we were all scared for what was ahead. Ever since the war, our country has been more volatile. Stroud keeps seizing power, and we will soon all be dead. My friends and I have formed an uprising. One day, we will restore our country. One day, we will depose Stroud. One day, we will have peace in our time."

Problem Description:

Rebellions are hard with just a few people. We have so many things to plan that our core group of friends is no longer enough. Luckily, we have a graphic designer made us a recruitment newspaper to share countrywide, but we need to redraw the paper for printing.

Output Description:

Print the latest secret newspaper in the land of Stroudonia. The numbers in the following sample are for alignment only.

```

0123456789012345678901234567890123
0\^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^\
1 \_      "REBELLION"      \
2  }=====
3  \  ~~~~~~ /
4  \  _____ {
5  \  \  \  ~~~~ DOWN ~~~{
6  \  \  \  ~~~ WITH ~~~ \
7  _\  ~~~~~ STROUDONIA! ~~~ \
8  @_ )_____ )

```

Sample Output:

```

\^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^\
\_      "REBELLION"      \
 }=====
 \  ~~~~~~ /
 \  _____ {
 \  \  \  ~~~~ DOWN ~~~{
 \  \  \  ~~~ WITH ~~~ \
 _\  ~~~~~ STROUDONIA! ~~~ \
@_ )_____ )

```

Civil War - Wargames

File Name: Wargames.java

Input: wargames.in

Problem Description:

With all of its new recruits, our rebellion is ready for battle! The rebellion is calculating if our mercenaries will victoriously defeat Stroudonian forces. We are running war games calculating the success of certain classes of our warriors against our classes.

Input Description:

The first line contains k, an integer representing the number of battles to follow (number of datasets). For each dataset, the first line contains n, an integer showing the number of possible matchups in the battle to follow. The next n lines represent the outcomes of each matchup.

Each matchup consists of 2 characters representing the winner and loser of the fight respectively. The first character of each line is the winner of that type of matchup. The second character is the loser.

The next line contains r and c in that order and represents the number of rows (r) and the length of each row (c) for the battlefield. The next r lines contains the rows of the battlefield. For each row, the second character in the row is the rebellion fighter and the second to last character is the Stroudonian fighter. All other characters are irrelevant.

Input Restrictions:

All rows in each of the battlefields will have an outcome depicted by the matchups.

Output Description:

For each of the rows determine which fighter won. If the rebels won more rows than the Stroudonians, print "The rebels have won!". If Stroud's forces won more rows, print "The empire still stands...". If the two sides tied, print "The war isn't over".

Sample Input & Sample Output On The Next Page

Civil War - Wargames (cont.)

File Name: Wargames.java

Input: wargames.in

Sample Input:

```
3
3
A B
A C
C B
4 7
#B--A#
#C-=|A#
#A==B#
#B|=C#
5
A B
A C
C B
D E
A E
6 10
#B---|A#
#C-=|=-A#
#A===|:-B#
#B|=-==C#
#D---|E#
#A|=-==E#
1
Z A
1 4
#ZA#
```

Sample Output:

```
The empire still stands...
The war isn't over
The rebels have won!
```

Civil War - Secure Communications

File Name: Secure.java

Input: secure.in

Problem Description:

Our chances of winning this war are slim. As the rebel leader, I decided to escape to the Carmouche Wildlife Preserve and lead the rebellion from there. Such a move may be called cowardice; however, those people aren't leading a rebellion. I want to ensure the longevity of the rebellion in the likely event that our forces in Stroudonia are killed. I will be in secure contact with the head base of our rebel army in Stroudonia. However, before I go, I want to ensure that our rebel forces' communication network is resilient, in the event that the war disrupts one of our secure comm lines.

Our current communication network is organized like a tree, with our head base being the root, and smaller bases being connected to their superior (parent) by secret underground connections. While this configuration is conservative with resources, it is highly vulnerable to disruptions, as even a single line being cut could disconnect the entire network.

Before I leave, I would like to see additional connections established between bases in our network so that even if one any one connection is cut, the network stays together. Because the construction is so conspicuous, I want to have to add as few as possible to avoid having Stroud learn of our network.

Input Description:

The first line contains t , an integer dictating the number of test cases to follow.

The first line of each test case contains an integer n , indicating the number of bases in our communication network. The head base (root) is base number 1. The following $n-1$ lines describe the superior (parent) base of each base number that follows, starting with base number 2.

Output Description, Sample Input & Sample Output On The Next Page

Civil War - Secret Communications (cont.)

File Name: Secret.java

Input: secret.in

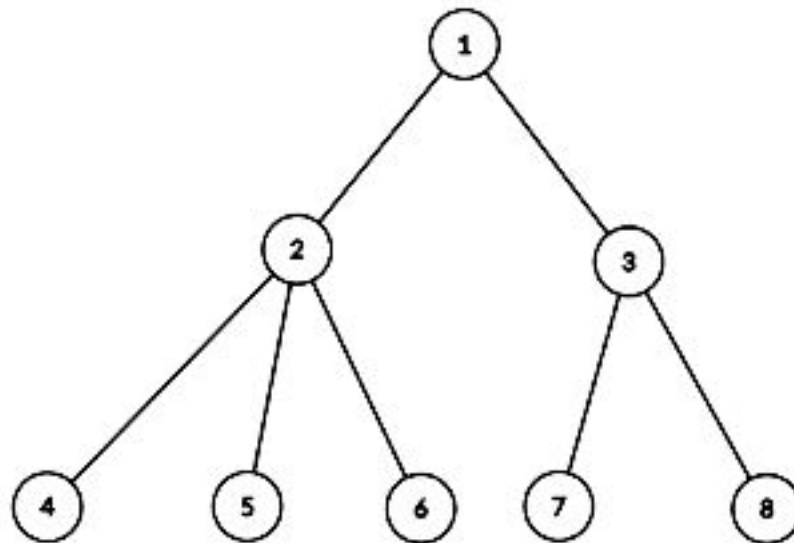
Output Description:

For each test case, print the minimum number of additional connections needed to strengthen our network on a separate line.

Sample Input:

2
8
1
1
2
2
2
3
3
5
1
1
2
3

Input Visualization:



Sample Output:

3
1

Sneaking Out - Safe Houses

File Name: SafeHouses.java

Input: safehouses.in

"I remember the old days, just a few weeks ago. The days before the war, the days where trade meant increased peace and prosperity. My family and I were celebrating our great success when we heard rumbling. Stroudonian tanks had rolled into the city, and before we knew it Stroud placed us under martial law. Those days are now gone. He conscripted us in preparation for his conquest on our Kingdom's capital. I know I can't stop the capture of our capital. No one can. But I sure am saving myself."

Problem Description:

Ever since Stroudonia invaded my hometown, my fellow Forest Kingdom friends have set up safe houses across the occupied area. Of course, Stroudonia is also setting up fake safe houses, as traps for rebels and traitors. Sam, the Prince Regent, and manager of the safehouses, loves the number 7. If a potential safehouse's number is divisible by 7, the safehouse is not a trap.

Input Description:

The first line contains n , an integer dictating the number of test cases to follow. The following n lines contain the street address to analyze.

Output Description:

The output will consist of n lines. Each line will either print out the address if the house is safe, or "DANGER!" otherwise.

Sample Input:

```
4
9251 South Fry Road
1484 Forest Kingdom of Armstrong
1603 Stroudonia Avenue
86318 Arubian Lake Drive
```

Sample Output:

```
DANGER!
1484 Forest Kingdom of Armstrong
1603 Stroudonia Avenue
DANGER!
```

Sneaking Out - Organizing

File Name: Organizing.java

Input: organizing.in

Problem Description:

It's been settled. We're going to sneak out of Stroudonia. The borders are locked down tight with 24/7 guards, but their military port is criminally insecure. Before leaving, I want to sort my stuff so I can find things easier.

Each item is assigned a utility, a weight, and a cost. When sorting, I prefer higher utilities above all else. If the utilities are the same, I prefer lower weights. If the weights are also the same, then I prefer the cheaper item. There will be no items with all values equal.

Input Description:

The first line contains n , an integer dictating the number of items to follow. The following n lines contain the data to analyze. Each line contains an item in this format, where name is a String, and value, weight, and cost are integers:

<Name> <Utility> <Weight> <Cost>

Output Description:

The output should be a comma separated list of all items in sorted order. There are no spaces between commas and no trailing spaces.

Sample Input:

```
5
Wrench 11 16 54
Book 5 3 12
Screwdriver 5 1 25
Computer 784 168 653
Magazine 5 3 6
```

Sample Output:

```
Computer,Wrench,Screwdriver,Book,Magazine
```

Sneaking Out - Get to that Boat

File Name: Boats.java

Input: boats.in

Problem Description:

I packed my bags and went down to the docks. After weeks of planning, I was ready to steal a boat and escape. I had pondered my escape options. The Forest Kingdom was and is still likely to fall, and the Carmouche Wildlife Preserve was way too far away. Aruba it was then. Maybe I could tube the mythical Aruban beaches or even ride the dolphins.

Before my vacation, I will need to hijack a Stroudonian military ship and override the systems. While the dock is insecure, there are number of guards stationed throughout the area, and if one of them notices me, I will probably be caught and killed. I have mapped out the dock, noting the positions that guards are stationed.

In my map, there are certain safe zones, where I can hide without worry of a guard noticing me. I have also identified pathways between nearby safe zones and the time it takes to sneak through each pathway without alerting the guards. I would like to determine the shortest amount of time it will take me to sneak out to the boat, assuming that it takes me no time to go through a safe zone. Ideally, I want to hit the sea as soon as possible, and escape from Stroudonia.

Input Description:

The first line contains t , an integer dictating the number of test cases to follow. Each test case consists of the following:

The first line contains two integers, n and m ($2 \leq n, m \leq 10,000$), indicating the number of safe zones (including my starting location and the location of the boat) and the number of pathways between safe zones, respectively. All safe zones are numbered, with my starting location always being zone number 1, and the location of the boat always being zone number n . The following m lines contain three integers, u , v , and w . The integers u and v represent the numbers of nearby safe zones on the dock that are connected by a pathway (pathways can be crossed in either direction), while the integer w represents the time it takes to sneak through that pathway.

Output Description, Sample Input & Sample Output On The Next Page

Sneaking Out - Get to that Boat (cont.)

File Name: Boats.java

Input: boats.in

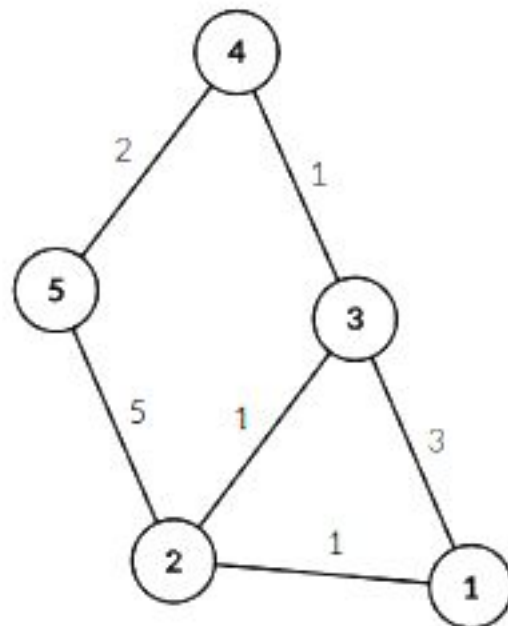
Output Description:

Each line should contain the minimum amount of time it takes to sneak out to the boat for the given dataset.

Sample Input:

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

Sample Input Visualization:



Sample Output:

```
5
8
```

Bribery - Corrupt Communication

File Name: Location.java

Input: location.in

Problem Description:

"My family got rich off of industry, and we've always lived a great life. The best time of my life was touring the capitol with my family. That was back when Stroud still pretended to be peaceful. The day the war broke out, little changed for me. Life continued, but slowly, things began to change. Arrests here, deportation there, and then the government seized property. It was for the war effort, Stroud said, but I believe they're after our wealth. Stroudonia has been a great country, but I'm moving to Aruba. It's so much more luxurious there"

Stroud has done a great job making his employees loyal to him, but there's always someone that isn't. I met an officer who needed money to pay for medicine. We made a quick code, and the mailman passed me a message today. The code is very simple...

I just take the first and last words of each line, and combine them into a sentence.

Input Description:

The first line contains n , an integer dictating the number of lines to follow. The following n lines contain a single line of the message.

Output Description:

The output should be the coded words (first and last words of every line) on a single line. There should be one space between each word, with a period at the end. Capitalization should be kept identical from the coded message to the output.

Sample Input:

3

See the Great Power Stroud has given Me
In the newspaper I see ads for StroudMart
Tomorrow we read the national anthem at 7AM

Sample Output:

See Me In StroudMart Tomorrow 7AM.

Bribery - Inflation Nation

File Name: Inflation.java

Input: inflation.in

Problem Description:

I'm planning on bribing some government officials, but I'm not sure how much it will cost. I'll need money to offset this risks of the treasonous act.

This equation was developed by the infamous communist, Quy Nguyen. It determines how much money I need to bribe an official, represent by the letter p. The letter c represents their wealth in StroudCoin.

The letter r represents an official's rebellion factor, and t their transportation control factor. I will need to sum all of official's assets and add it to their total StroudCoin balance. n is the official's social status. The higher their status, the more money I need.

$$P = \sqrt{\frac{(c + \sum a_i)^3 (2 - r)^{1/2} (\log_{10} t)}{n^{1.5}}}$$

Input Description:

The first line contains n, an integer dictating the number of data sets to follow. The first line of each dataset contains values for the equation. Each data set contains a single line with the numbers c, r, t, and n in that order. The next line contains o, an integer dictating the number of assets to follow. The following o lines contains a double indicating the value of each asset. There is an empty newline after each dataset.

$\sum a_i$ indicates the sum of all assets owned.

Input Restriction:

c satisfies $0 < c < 200,000$ and is a integer

r satisfies $0 < r < 1$ and is a double, t satisfies $0 < t < 10$ and is an integer

n satisfies $1 \leq n \leq 10$ and is a double, o satisfies $1 \leq o \leq 50$ and is an integer

asset values satisfy $0 \leq av < 200,000$ and are integers

Sample Input & Sample Output On The Next Page

Bribery - Inflation Nation (cont.)

File Name: Inflation.java

Input: inflation.in

Output Description:

Each line should contain P of each dataset. P is the cost of bribing a person, calculated according to the above formula. P should be formatted rounded to 2 decimal places and can vary by $\pm 1\%$ due to floating point errors.

Sample Input:

```
3
2000000 0.5 4 0.3
3
54587
612398
456001

2301479 0.1 5 4.25
0

145658 0.48 10 7.16
1
1651564
```

Sample Output:

```
14014369.14
253080.66
115652.22
```

Bribery - Aruban Packing

File Name: Packing.java

Input: packing.in

Problem Description:

After paying her, my contact has placed me on the next Stroudonian transport to Aruba. This is the end of my time in Stroudonia, but I have a lot of stuff here at home.

I'm bringing a backpack, but I have a limited carrying capacity. Each item I own has a unique name, a value, and a weight. I'll need to maximize the total value of items I can fit in my pack without exceeding my carrying capacity.

Input Description:

The first line contains n , an integer dictating the number of test cases to follow. The next integer is w , the maximum weight your backpack can handle. The following n lines contain the data to analyze. Each line contains the name, value, and weight, in that order.

Output Description:

The output should be the optimal items to take in the backpack. The output format should stay the same as in the input, but sorted according to name (there will be no repeats).

Sample Input:

```
6 15
Car 1200 160
Wrench 15 5
Omnitool 44 14
Poster 6 2
Paper 6 2
Laser 18 6
```

Sample Output:

```
Laser 18 6
Paper 6 2
Poster 6 2
Wrench 15 5
```