

Seven Lakes Kickoff

GLORY TO STROUDONIA



Welcome to Stroudonia

0. Table of Contents

1. Glory to Stroudonia!
2. A Study in Stroud's Likeness
3. Hello, Citizen!
4. The First Census
5. Doubleplusgood
6. The Second Census
7. Stroudship the First
8. Equal Citizens
9. Stroudship the Second
10. Dotetracontranary
11. Stroud / Stroud 2014
12. Stroudonia Citizen Engagement
13. The Third Census
14. May His Brilliance be Seen from the Highest Hill
15. More Glory to Stroudonia!
16. Rock, Worker, Bureau, Citizen, Stroud
17. The Bureau of Message Consistency
18. To War!

You may solve these problems in any order.
Do not write to any files. Always print to Standard Out.

Glory to Stroudonia!

1. Glory to Stroudonia!

Input File: None

Problem Description:

There's nothing quite like a little patriotism for your country, especially if it's absolutely mandatory. As part of your daily required display of affection to your homeland, the Bureau of Citizen Engagement has asked you to make a digital version of the flag. It's probably better for your health and longevity to comply.

Input Description:

None.

Output Description:

Output the flag as shown, to appease the officials.

Sample Input:

None

Sample Output:

```
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXX                                                                    X
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXX                                                                X
XXXXXXXXXXXXXX                  XX                                       X
XXXXXXXXXXXXXX                  XXXXXX                                   X
XXXXXXXXXXXXXX                  XXXXXX                                   X
XXXXXXXXXXXXXX                  XX                                       X
XXXXXXXXXXXXX                                                            X
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXX                                                                    X
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

2. A Study in Stroud's Likeness

Input File: stroudslikeness.in

Problem Description:

The Stroudonian Loyalty Society is producing small statues of His Brillianceness Stroud. The Loyalty Society has a very nice production system for creating these statues, but they are looking to optimise their production, which operates under the influence of the following:

- Each month, a certain number of loyal citizens will purchase the statues of His Greatness Stroud. We will call the number of statues ordered in one month the *demand for month n* , or \mathbf{D}_n .
- Statues of His Perfectness Stroud fluctuate in cost to produce based on the availability and grade of materials being used. We will call the price of one statue in a given month the *production cost for month n* , or \mathbf{C}_n .
- Space is limited in the Loyalty Society's warehouse. Therefore, it is undesirable to store overflow statues in some cases. However, sometimes it is good to produce statues before they are ordered. The cost to store one statue between months n and $n+1$ is called the *holding cost for month n* , \mathbf{H}_n . You may NOT produce statues before the first month, nor may you hold statues past the last month. Therefore, the last month will not have a holding cost.
- Sometimes, it may be necessary to deliver statues in months after they are ordered. Of course, customers will receive a discount, which is considered a cost for the Loyalty Society. This is called a backlog. We will call the price of backlogging one statue between months n and $n+1$ the *backlog cost for month n* , or \mathbf{B}_n . You may NOT backlog statues to before the first month, nor may you fulfil orders past the last month. Therefore, the last month will not have a backlog cost.

The Stroudonian Future Society has predicted the demand, production cost, holding cost, and backlog cost for one statute for the next several months. It is now your job, which you may not refuse, to write a program that determines the **optimal policy** for producing statues; in other words, what is the minimum cost of producing statues?

Input Description:

The first line of the input will represent the number of data sets to follow.

Each data set will begin with a single integer n , which represents the number of months for which you will have to calculate the optimal policy. The next n lines will contain four space separated positive integers in the following order: \mathbf{D}_i , \mathbf{C}_i , \mathbf{H}_i , \mathbf{B}_i , where i represents the month. The values of \mathbf{H} and \mathbf{B} for the final month will always be -1 and should be ignored.

Output Description:

For each data set, output the cost for satisfying all orders using the optimal policy.

Sample Input:

```
2
5
3 5 9 10
4 3 4 10
5 8 1 10
6 9 6 2
7 3 -1 -1
3
1 1 1 1
1 1 1 1
1 1 -1 -1
```

Sample Output:

```
113
3
```

3. Hello, Citizen!

Input File: hellocitizen.in

Problem Description:

In the glorious land of Stroudonia, there is exactly one proper greeting to use when meeting a fellow citizen:

Hello, Citizen <name>! Glory to Stroudonia!

Failure to use this greeting may result in a point deduction from your team for your lack of loyalty to Stroudonia.

There is one exception: If you ever encounter Stroud, this is the proper greeting to use:

Hello, Stroud! May your brilliance guide our computer scientists forever!

Input Description:

The input will contain an unknown number of lines, with each line being the name of a citizen.

Output Description:

Greet each citizen with the proper Stroudonian greeting.

Sample Input:

```
Beck
Lorena
Jacob Van Geffen
Stroud
Hicks
```

Sample Output:

```
Hello, Citizen Beck! Glory to Stroudonia!
Hello, Citizen Lorena! Glory to Stroudonia!
Hello, Citizen Jacob Van Geffen! Glory to Stroudonia!
Hello, Stroud! May your brilliance guide our computer scientists forever!
Hello, Citizen Hicks! Glory to Stroudonia!
```

4. The First Census

Input File: census1.in

Problem Description:

The glorious People's Republic of Stroudonia, after learning about the wonders of statistics, have decided to have its first official census! The officials in charge aren't quite sure what a census is, however, so they've decided to start off simple: by counting the number of people in Stroudonia.

Input Description:

There will be an unknown number of lines in the input. Each line will contain the name of one faithful citizen of Stroudonia.

Output Description:

Output the total number of citizens of Stroudonia, based on the list described in the input description.

Sample Input:

```
Geffen, Jacob  
Stroud, His Excellence  
Tully, Jason
```

Sample Output:

```
3
```

5. Doubleplusgood

Input File: doubleplusgood.in

Problem Description:

This is a doubleplusgood program. Bellyfeel this program or become an unperson. All you are asked to do is rewrite the oldspeak sentence into goodthink CompSpeak®.

| The following phrase... | should be re-written as... |
|-------------------------|----------------------------|
| very | double plus |
| bad | ungood |
| nonsense | duckspeak |
| computer science | joycamp |

All of these phrases will appear in lowercase only, and will never appear in the middle of other words.

Input Description:

The first line of the input will be a single integer n , representing the number of data sets to follow. Each data set will be a single line and will contain a sentence or paragraph of Oldspeak that must be translated to CompSpeak®.

You may assume that none of the phrases you are searching for will appear as a substring of any other word in the sentences.

Output Description:

For each data set, print out the complete CompSpeak® sentence.

Sample Input:

```
3
I think that computer science is a very good class.
However, my friend says that computer science is nonsense.
We all know that this is a very bad thing to say!
```

Sample Output:

```
I think that joycamp is a double plus good class.
However, my friend says that joycamp is duckspeak.
We all know that this is a double plus ungood thing to say!
```

6. The Second Census

Input File: census2.in

Problem Description:

The glorious People's Republic of Stroudonia has realized that instigating a census to only determine how many people live in the nation is not a very effective or useful census. Always ones to persevere and improve, the new officials in charge of the census (after the "re-education" of the old ones) have decided to actually gather some important information: the gender and age of each citizen in addition to their name.

Input Description:

There will be an unknown number of lines in the input. Each line will contain a name in the format of "Last, First", followed by a space and the gender of the citizen as "Male" or "Female", followed by another space and the age of the person as an integral number of years. There will be no spaces in the first or last name.

Output Description:

Output the following statistics as shown in the sample output, in the format shown in the sample output: Number of Citizens, Number of working-age citizens, Male Percentage, Female Percentage, Oldest Person. The percentages and average age should be output to 3 decimal places. The number of working-age citizens is the number of citizens who have an age between 16 and 65, inclusive.

Sample Input:

```
Geffen, Jacob Male 50
Gates, Bill Male 59
Smith, Annabelle Female 15
```

Sample Output:

```
Number of Citizens: 3
Number of Working-Age Citizens: 2
Male Percentage: 66.667%
Female Percentage: 33.333%
Oldest Person: Gates, Bill - 59
```


7. Stroudship the First

Input File: stroudship1.in

Problem Description:

You are working in the Stroudship Office, which is in charge of assessing people and determining if they are worthy to become citizens of Stroudonia. The Stroudship Office currently has some very simple policies: the first n applicants will be processed each day. Given this very simple policy, you have been tasked with writing a program to determine who will be processed for Stroudship today, and announcing to everyone who those lucky few are.

Input Description:

The first line of data will be a single integer n , representing the number of applicants who will be processed today. The rest of the input will contain an unknown number of lines, representing the people who are standing in line at the Stroudship Office. The people will be listed one person per line in the order that they are standing in line.

Output Description:

Begin the output with the phrase, "The people who will be processed today are:".

Output the people who will be served, in the order they are given, one person per line.

At the end of the output, display "Come back tomorrow!" so that everyone knows that they may try again.

Sample Input:

```
3
Luke
Mister Brennan
Claire
Colby Brown
Scott
Miguel
Jacob
Mr. Schniggles
```

Sample Output:

```
The people who will be processed today are:
Luke
Mister Brennan
Claire
Come back tomorrow!
```

8. Equal Citizens

Input File: equalcitizens.in

Problem Description:

In the perfect society of Stroudonia, all citizens are created equal. However, this causes many logistical problems, so Stroudonia has declared that some citizens may be more equal than others. How equal a citizen is determined in the following manner:

- Citizens are first sorted based on their **Loyalty Index**, with citizens with higher indexes being more equal than citizens with lower indexes.
- Citizens are then sorted based on the **Computer Merit Score**, with citizens with higher scores being more equal than citizens with lower scores.
- Citizens are then sorted based on their **Sufficiency Penalty**, which is a cumulative record of citizen's transgressions against Stroudonia. A citizen with a smaller sufficiency penalty is more equal than a citizen with a greater penalty.
- Citizens are finally sorted on their **Monetary Gifts**, which encompass taxes and other monetary factors. A citizen who has given more monetary gifts is more equal than a citizen who has given less monetary gifts.

Remember, a citizen named "Stroud" will ALWAYS be more equal than his fellow citizens.

Given the current Stroudonian Registry, sort the citizens from most equal to least equal.

Input Description:

The first line of the input will be a single integer n , representing the number of citizens in the registry. The next n lines will contain the citizen's information, one citizen per line, with the following space separated information:

- Name, without spaces
- Computer Merit Score, an integer ranging from 0 to 100 inclusive.
- Loyalty Index, a floating point number with two decimal places ranging from -100 to 100 inclusive.
- Sufficiency Penalty, an integer ranging from 0 to 99,999 inclusive.
- Monetary Gifts, a real number with up to two decimal places, ranging from 0 to 99,999.

Output Description:

Output the name of the citizens, one citizen per line, in the sorted order from most equal to least equal, with the most equal citizen being printed first. An empty line at the end of the output is allowed.

Sample Input:

```
5
Comrade_Scott 75 90 0 1000
Mister_Brenan 99 84 0 250
Stroud 0 0 0 0
Luke 75 90 40 1000
Miguel 98 84 5 400
```

Sample Output:

```
Stroud
Comrade_Scott
Luke
Mister_Brenan
Miguel
```

9. Stroudship The Second

Input File: stroudship2.in

Problem Description:

You are working in the Stroudship Office, which is in charge of assessing people and determining if they are worthy to become citizens of Stroudonia. There have been a lot of complaints in recent weeks that the office is not processing enough people, and that people who arrive late in the day have no chance of being processed. Therefore, the Stroudship Office is implementing a new policy: The first n applicants will be processed in addition to the last n applicants. This way, all applicants have equal potential of being approved for full Stroudonian citizenship!

Input Description:

The first line of data will be a single integer n , representing the number of applicants who will be processed today. The rest of the input will contain an unknown number of files, representing the people who are standing in line at the Stroudship Office. The people will be listed one person per line in the order that they are standing in line.

Output Description:

Begin the output with the phrase, "The people who will be processed today are:".

Output the people who will be served, in the order they are given, one person per line.

At the end of the output, display "Come back tomorrow!" so that everyone knows that they may try again.

In the event that there are fewer than $2n$ people applying today, then output the list of people in the same order that it is given to you without modification.

Sample Input:

```
3
Luke
Mister Brenan
Claire
Colby Brown
Scott
Miguel
Jacob
Mr. Schniggles
```

Sample Output:

```
The people who will be processed today are:
Luke
Mister Brenan
Claire
Miguel
Jacob
Mr. Schniggles
Come back tomorrow!
```

10. Dotetracontranary

Input File: dotetracontranary.in

Problem Description:

Stroudonia is developing a new encryption system to send secret messages that can not be intercepted by the enemy. They have come up with an encryption system that takes a letter and shifts it 42 positions forwards in the alphabet, wrapping the letter z around to the letter a. Messages to be written in dotetracontranary will only ever contain lowercase letters; the uppercase letters and punctuation have already been removed for you.

Your mission, which you can not refuse under violation of loyalty to Stroudonia, is to write a program that takes a message written in English and converts it to its dotetracontranary counterpart.

Input Description:

The first line of the input will be a single integer n , representing the number of messages to be encoded. The next n lines will contain one message per line, which will only contain lowercase letters and spaces.

Output Description:

For each message, output the message encrypted into dotetracontranary on its own line. An empty line after the last message is allowed.

Sample Input:

```
3
glory to stroudonia
the metropolis of computer science
may stroudonia stand forever
```

Sample Output:

```
wbeho je ijhektedyq
jxu cujhefebyi ev secfkjuh isyudsu
cqo ijhektedyq ijqdt vehuluh
```

11. Stroud / Stroud 2014

Input File: stroudstroud2014.in

Problem Description:

The glorious People's Republic of Stroudonia is holding a vote for their new President. The incumbent President, His Gloriousness Stroud, is running for re-election. Everyone already knows that Stroud, given his infallible leadership, is guaranteed to win the election. Therefore, it is only reasonable to assume that any other result is due to cheating.

After the votes are collected and counted, we will assume that some of the candidates in the election have cheated and should thus be captured and imprisoned and all of their votes discounted. The discounted votes will not count towards the total number of votes. However, it is only reasonable to assume that as few candidates cheated as possible (for reasons of economy, of course). As Chief Executive Counter of the Votes of the Ministry of Elections, it is your job to determine which candidates cheated in the election.

You are to accuse as few candidates as possible of cheating such that, when cheating candidates' votes have been discounted, Stroud has 50% **or more** of the votes.

Stroudonia uses a "first past the post" system, which means that any candidate who receives 50% or more of the votes is the winner.

Input Description:

The first line in the input will be the number of data sets to follow. Each data set will start with a single integer n representing the number of candidates in the election. The next n lines will contain the name of the candidate, without spaces, followed by the number of votes that they received. Stroud will be one of the candidates, but not necessarily the first.

Output Description:

For each data set, output the names of the candidates who cheated in alphabetical order on the same line. Assume that the fewest number of candidates cheated as possible; the case with the minimal amount of cheaters will always be unique (in other words, there will never be any ambiguity about who should be considered cheating). If Stroud won the election without any cheaters, then display "Glory to Stroudonia!" instead.

Sample Input:

```
2
3
Stroud 100
McGee 40
Brenan 30
5
Tenant 75
Smith 200
Stroud 125
Capaldi 50
Eccleston 175
```

Sample Output:

```
Glory to Stroudonia!
Eccleston Smith
```

12. Stroudonia Citizen Engagement

Input File: compress.in

Problem Description:

The Bureau of Citizen Engagement has congratulated you for your intense dedication to His Brilliance Stroud and Stroudonia by successfully making a digital version of the flag. To spread your great work of art (which has been framed in the Capitol building), as well as other images of great importance, the Bureau of Citizen Engagement has asked you to write a compression algorithm for simple binary images.

The compression algorithm is defined as follows. Take groups of 8 characters and convert "X" to 1s and "." to 0s. Convert the resulting binary number into a 2 digit hexadecimal number.

```
XXXX.X.X -> 11110101 -> F5
XXXXXXXX -> 11111111 -> FF
.....X -> 00000001 -> 01
XX..... -> 11000000 -> A0
```

Input Description:

The first line of the input will be a single integer n, representing the number of images to be compressed. The next n images will contain an unknown number of lines, each containing some multiple of 8 characters, either "X" representing 1 or "." representing 0. Each image will be separated with an empty line.

Output Description:

Apply the compression algorithm as described for each group of eight characters, output their uppercase hexadecimal equivalent, and separate each group of 8 characters on the same line by a space. Each compressed image should be separated by an empty line. An empty line at the end of the output is acceptable.

Sample Input:

```
3
XXXX....

X.X.X.X.X.X.X.X.X.....
.X.X.X.X.X.X.X.X.X.X....

XX.....
.XX.....
XX.....
```

Sample Output:

```
F0

AA AA A0
55 55 50

C0
60
C0
```

13. The Third Census

Input File: census3.in

Problem Description:

The glorious People's Republic of Stroudonia has finally got a hang of the "census", and is now eager to put out it's most comprehensive census yet. In order to improve the people's standard of life, the government is taking a comprehensive economic census, grouped by on the various districts in the country. Thanks to your excellent work on the previous censuses, you have again been *randomly* selected to create the analyzing program.

The program must be able to conduct statistical analysis, namely means and standard deviations. The mean (also known as the average) can be found by dividing the sum of the elements by the number of elements. Standard Deviation is slightly more complicated:

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \mu)^2}$$

While this formula may be familiar to anyone who has taken statistics, the summation and notation may not be familiar to some people. In laymans terms, there are 4 steps to find the population standard deviation:

1. Find the mean of the population (the group of people you're finding the standard deviation for).
2. For each number, subtract the mean and square the result.
3. Find the average of each of **those** numbers.
4. Finally, take the square root of the average that you found.

You will need to make use of your newfound mathematical knowledge to find the average and standard deviation of age and income for each district.

Input Description:

There will be an unknown number of districts, whose start is denoted by "DISTRICT "[NAME]" on it's own line, where [NAME] is the name of the district. Each district will have an unspecified number of citizens in the form "[Last], [First] [Gender] [Age] \$[Income]". Last and First name will only have alphanumeric characters, Gender will be one of "Male" or "Female", and Age and Income will both be integral values which can be stored in an int type.

Output Description:

You will be producing two different rankings, the first of age and the second of income, based on mean and standard deviation, sorted by decreasing mean and then increasing standard deviation as a tiebreaker. All numbers should be output to 2 decimal places, and monetary amounts should be prefixed with a dollar sign and comma-separated.

13. The Third Census (cont.)

Sample Input:

```
DISTRICT "Azstortzka"  
Smith, Roger Male 50 $20,000  
Gates, Steve Male 40 $14,000  
Smith, Annabelle Female 15 $0  
DISTRICT "Kolechia"  
Berman, George Male 35 $27,000  
Prasana, Ayuush Male 17 $12,000  
Romanov, Alexis Male 8 $0  
Stroud, Paul 42 $68,000
```

Sample Output:

```
DISTRICTS BY AGE  
Azstortzka 35.00 14.72  
Kolechia 25.50 13.61  
  
DISTRICTS BY INCOME  
Kolechia $179,750.00 $288,977.83  
Azstortzka $11,333.33 $8,379.87
```


14. May His Brilliance Be Seen From The Highest Hill (cont.)

Sample Input:

```
1
10 13
. . . . .
. ^ ^ ^ ^ ^ ^ ^ ^ ^
. ^ ^ ^ ^ . . . . . ^
. ^ ^ . . ^ . . . . . ^
. ^ ^ . . ^ . . . . . ^
. ^ ^ . . ^ ^ ^ ^ ^ ^ ^
. ^ ^ . . . . . . . ^ ^
. ^ ^ . . . . . . . ^ ^
. ^ ^ ^ ^ ^ ^ ^ ^ ^ ^ ^
. ^ ^ ^ ^ ^ ^ ^ ^ ^ ^ ^
```

Sample Output:

```
. . . . .
. ^ ^ ^ ^ ^ ^ ^ ^ ^
. ^ ^ ^ ^ . . . . . ^
. ^ ^ S S ^ . . . . . ^
. ^ ^ S S ^ . . . . . ^
. ^ ^ S S ^ ^ ^ ^ ^ ^ ^
. ^ ^ S S S S S S S S ^ ^
. ^ ^ S S S S S S S S ^ ^
. ^ ^ ^ ^ ^ ^ ^ ^ ^ ^ ^
. ^ ^ ^ ^ ^ ^ ^ ^ ^ ^ ^
```

15. More Glory to Stroudonia!

Input File: None

Problem Description:

Today, like any other glorious day, you are to read the national anthem at 7:00 AM sharp. Failure to do so will result in a friendly visit from the Bureau of Citizen Engagement. The national anthem is as follows:

Stroudonia

Stroudonia greatest country in the world.

*If you want to get in
you will need to bring your programs
and your skill.*

If you forge, if you lie, if you cheat you will be detained.

*You can not, I repeat, you can not
fool us here on judge.*

Next.

Stroudonia

Stroudonia better than everyone else.

*We have supercomputers, we have PCs, we have phones
we have glory too.*

*There is no cowardly enemy that can bother us.
Compute.*

Stroudonia

Stroudonia medicine is best in world.

*If you're sick, if you sneeze, or you need another heart.
Or a lung.*

*We have doctors who will make you feel almost no pain.
If you scream, then they know that you are very much alive.
Ouch.*

Stroudonia

Stroudonia have best football team in world.

*They kick balls, they kick dirt, they kick everything you see.
They are great.*

*They can run, they can jump, they can score so many goals,
other teams should despair, there is no true hope to win.
Goal!*

15. More Glory To Stroudonia (cont.)

Stroudonia

*Stroudonia have best apartments in the world.
They are big, they are grey, they are square and glorious.
So great.
You can eat, you can sleep, you can even just relax
and watch all the other countries be really jealous.
Ha-ha.*

Stroudonia

*Stroudonia economy is best in world.
Every man, every month can afford to eat one loaf.
Mmm, bread.
Our industry is heaviest in world
we melt steel, we burn coal, we make air look really cool.*

Glory to Stroudonia!

Input Description:

None

Output Description:

Output the 8th verse of the national anthem.

Sample Input:

None

Sample Output:

Glory to Stroudonia!

16. Rock, Worker, Bureau, Citizen, Stroud

Input File: rwbcs.in

Problem Description:

It is time to decide who next has the honor of leading the daily national anthem! How about a quick game of Rock, Worker, Bureau, Citizen, Stroud? This modified version of Rock, Paper, Scissors, Lizard, Spock has been approved by His Brilliance Stroud as the definitive method for the resolution of all arguments.

Unfortunately, the game can sometimes take a long time as everyone picks Stroud every single time, so the Bureau of Citizen Independent Thought has tasked you with creating a program which plays the game for citizens, so they don't have to!

The game of Rock, Worker, Scissors, Citizen, Stroud is defined as followed:

Stroud beats Rock, Worker, Bureau, Citizen
Citizen beats Worker
Bureau beats Citizen, Worker, Rock
Worker beats Rock
Rock beats Citizen

Input Description:

The first line of the input will be a single integer n , representing the number of data sets to follow. Each data set will be Player 1's choice and Player 2's choice.

Output Description:

For each data set output who wins or "Tie" if the players choose the same.

Sample Input:

```
5
Stroud Citizen
Rock Rock
Rock Citizen
Rock Bureau
Bureau Stroud
```

Sample Output:

```
Player 1
Tie
Player 1
Player 2
Player 2
```

17. The Bureau of Message Consistency

Input File: bureau.in

Problem Description:

Today, you have been assigned to work at the recently formed Bureau of Message Consistency. The bureau was recently formed as a subdivision of the Ministry of Citizen Media, and has been tasked specifically with determining if given messages have the proper amount of Stroudness. Your task, if you choose to accept it (which you already have, as a loyal citizen of Stroudonia would *never* refuse a government order), is to write a program to do the Bureau's work.

The program, given a message, should determine if the message has enough of Stroudness. For a message to have enough "Stroudness", the message must have more of the letters in his loyalty's name than the letters in the rest of the alphabet (ignoring punctuation and other characters). The case of the letters doesn't matter.

| | | |
|---------------------|------------------|--------------------------|
| - Ex: Stroud. | 100% Stroudness | Approved |
| - Ex: This word. | 62.5% Stroudness | Approved |
| - Ex: Strown | 50% Stroudness | Approved |
| - Ex: Hello, world. | 30% Stroudness | Not up to Specifications |

Input Description:

The first line of the input will be a single integer n , representing the number of messages to be checked. The next n lines will contain one message per line.

Output Description:

For each line, report whether the batch is "Approved" or "Not up to Specifications". An empty line after the last report is allowed.

Sample Input:

```
3
Glory to Stroud. Glory to Stroud. Glory to Stroud. Glory to Stroud.
Glory to Stroud. Glory to Stroud. Glory to Stroud. (this will appear on one line
in the input)
This word. Plus this word.
This will not be approved.
```

Sample Output:

```
Approved
Approved
Not up to Specifications
```

18. To War!

Input File: war.in

Problem Description:

Stroudonians, war is upon us! Commissar McGee of the Glorious People's Green Army of our Glorious People's Republic of Stroudonia has mobilized all troops to overtake and integrate the democratic republic of Tullyland for the capital offense of spreading the lies of "freedom" and "liberty" and misleading our very own citizens! While His Supremeness Stroud is certain of the victory of our Glorious People's Green Army, Commissar McGee is more cautious and has requested the Ministry of War to create a program to determine his chance of success in a given engagement. The Ministry of War, after a great amount of deliberation, has created a glorious formula which infallibly determines the percentage of success of a given conflict:

$$\% \text{ Success} = \frac{o}{o+e}(|r - i|)^t$$

In the equation, "o" represents our troop amount, "e" represents their troop amount, "r" represents the rest of our troops as a number between 0 and 100, "i" represents the enemy initiative as a number between 0 and 100, and "t" is the time of the engagement in hours.

The Ministry of War assures us that any results we get with this formula are **absolutely correct**.

Input Description:

The first line of the input will be a single integer n, representing the number of data sets which will follow. The following n lines will contain 5 numbers: "o" as an integer, "e" as an integer, "r" as an integer, "i" as an integer, and "t" as a decimal value. All values will be less than 100000.

Output Description:

For each line, output the probability of victory as a percentage to 2 decimal places.

Sample Input:

```
3
10000 50 100 0 1.00
1000 1000 75 25 2.00
800 2000 30 50 0.75
```

Sample Output:

```
99.50%
1250.00%
2.70%
```