# You need a CAT Scan

After a long day at the recording studio Ke$ha is walking home with her friends when they notice an old man staring at them. This got Ke$ha thinking about how old someone must be to be classified as a dinosaur. Ke$ha settles on if a person was born before 1975 they are a dinosaur, and from 1975 and after they’re still sexy. The one exception to this is Ke$ha’s good friend Mick Jagger, whom she will always regard as sexy. Your job is to write a program where Ke$ha can plug in a person’s age and get a definitive answer on whether or not they’re a dinosaur.

Input: The input for this problem will begin with an integer N (0 < N < 100), the number of data sets. Each data set will contain a single line containing a person’s name between single quotes (and not containing quotes) followed by their birthday in the format “Month day, year” presented in AD (CE).

Output: For each data set, the program should print out a single line, stating the Person’s name, and whether or not they are a DINOSAUR.

### Sample Input

5

'Mick Jagger' July 26, 1943

'Mark Lewis' May 14, 1974

'Justin Bieber' March 1, 1994 'Ke$ha' March 1st 1987

'Keith Wedelich' February 14, 1992

### Sample Output

Mick Jagger is SEXY!

Mark Lewis is a DINOSAUR! Justin Bieber is SEXY!

Ke$ha is SEXY!

Keith Wedelich is SEXY!

# Ke$ha Sucks at StarCraft

Ke$ha plays StarCraft II with an AI that Boudreaux and Thibodeaux wrote. Ke$ha always plays Zerg, while Boudreaux and Thibodeaux always choose Terran for their race. Ke$ha sucks at

$tarCraft and the AI is programmed (obviously), so each has a predefined set of rules that they use to move their units. The layout of the map is given as input in the following format in a rectange of a given side length:

**Map Layout**

$ = Ke$ha’s Base

X = Boudreaux and Thibodeaux Base

\* = empty square

# = blocked square

Boudreaux and Thibodeaux’s AI always follows the following rules:

### Base Information and Unit Production:

The bases have 100 health points to start with.

When a base produces a unit, the unit is placed around the base in the following way: first, try the space directly above the base; if this space is blocked try east, south, and west, i.e. in clockwise order. The diagram below gives the order

1 1

4 X 2 4 $ 2

3 3

If all spots are blocked, then no units are produced that turn. These rules are the same for both the AI and Ke$ha

Each base makes one unit in the first move. Ke$ha’s base continues to make a unit every turn. The AI only makes a unit every other turn from that point on. (You can think of it as even or odd turns. If a turn is skipped because all adjacent squares are occupied, it will still be two more turns before the AI base can try again.) Ke$ha’s base goes before the AI base in each turn. Bases will never be on the edge of a map.

### About the Units:

---\*---

--\*\*\*--

-\*\*\*\*\*-

\*\*\*M\*\*\*

-\*\*\*\*\*-

--\*\*\*--

---\*---

To keep things simple, both sides know where everything is on the map at all times. Units act in order of their age, so the unit that has been in existence longest is moved first.

Ke$ha only builds Zerglings (35 hp, 5 dmg, 1 attack range, 2 moves/turn). The AI only builds Marines (45 hp, 6 dmg, 3 attack range, 1 move/turn). The Marine attack range is shown to the right, \* for in range, - for out of range. Zerglings can only attack adjacent squares up, down, left, and right.

What each unit does during a turn is specified by the following algorithm.

* Acquire an enemy target. If one or more are in range, select the oldest. If none are in range, select the one with the highest quotient of (priority/distance) for distance along the shortest path. In case of a tie, the oldest is the target. (Note that the bases are older than all units as each side starts with only a base.)
  + Priority is 5.0 for the base and 1.0 for units.
  + Distance is in allowed moves for locations that are open along the optimal path. That means they aren’t blocked on the map and they aren’t currently occupied. If a target can’t be reached, use a distance of 1,000,000.
  + The AI, being a program, will check for a new target every turn. Ke$ha, being a significantly slower human, will only re-evaluate targets every 5 turns. If the current target dies, the unit that was targeting it will do nothing and Ke$ha will not be able to give the unit a new target until the end of the 5 turns.
* If the target is in attack range, attack it.
  + Marines can shoot anything for which the sum of the distance in x and y is 3 of less. It is possible to shoot over other units or blocked squares.
* Otherwise, move toward the current target along the shortest path.
  + Moves are in x or y direction, not diagonal.
  + If there are two possible directions with a tied length to the selected object, pick the direction given the same order of priority for placing units around the base.
  + Zerglings can do this process twice each turn. If the first move puts them adjacent to their target, they will not use the second move. They can not attack as the second half of their turn.
* If nothing is reachable (nothing is in range and all possible paths are blocked to all enemy units and the enemy base so no target can be acquired), the unit will do nothing.

Units do nothing in the turn in which they are built. If an older unit kills a younger unit during a turn, the younger unit does not get an action. Instead, it is taken off the map immediately and the square it was on is open for other units younger than the killer to occupy.

**Input:** The input starts with a line that has a single number, 0 < N < 100, for the number of boards you have to consider. Each board starts with a number for how many rows, 4 < R < 51, are in the board. That will be followed by R lines, all of the same length, with the initial map for the board.

Each line will be 50 or fewer characters in length.

### Output:

For each map you will output one line. If Ke$ha wins in the first 1000 turns print “Ke$ha wins!” If the AI wins in the first 1000 turns print “AI wins!” If neither has won by 1000 turns, print “Let’s call it a draw.”

**Example:** Due to the complexity of this problem, we will run through the beginning of an example to make it clear what happens. Consider the following map.

\*\*\*\*\*\*\*

\*X\*\*\*\*\*

\*\*\*##\*\*

\*\*\*\*\*#\*

\*#\*\*$\*\*

\*\*\*\*\*\*\*

**Step 1:** During the first turn, Ke$ha’s base will make a Zergling, then the AI base will make a Marine. After the first turn the play area might look like this:

\*m\*\*\*\*\*

\*X\*\*\*\*\*

\*\*\*##\*\*

\*\*\*\*z#\*

\*#\*\*$\*\*

\*\*\*\*\*\*\*

The ‘z’ is the Zergling and the ‘m’ is the marine.

**Step 2:** In the second turn, the the order of events becomes important. First, Ke$ha’s base builds another Zergling. Because the first one is in the first choice position, the new Zergling goes to the right of the base. After that the first Zergling acquires the AI base as a target and moves left two

squares. Last, the first Marine acquires Ke$ha’s base as a target and moves right one square. This leaves the playing field looking like the following:

\*\*m\*\*\*\*

\*X\*\*\*\*\*

\*\*\*##\*\*

\*\*z\*\*#\*

\*#\*\*$z\*

\*\*\*\*\*\*\*

**Step 3:** For the third step, the order of taking action is Ke$ha’s base, the AI base, the first Zergling, the first Marine, and the second Zergling. Ke$ha’s base makes another Zergling at the 1st position. The AI base makes a Marine at the 1st position. The first Zergling moves up twice to be next to the target it had acquired in the previous step. There are two things to note here. First, even if there had been Marines in the way, the Zergling would still keep going for the base because it can’t acquire a new target until step 7, five steps after it originally acquired the base. Second, there are many routes that get the Zergling next to the base in two steps. However, the priority is to consider up, right, down, then left. Instead of up, up, the Zergling could go left, up or up, left, but it will not consider any left moves unless that is superior to up, right, and down.

The first Marine now sees the first Zergling as being in range. Remember that Marines are under the AI and they can acquire a new target every turn. The Zergling being in range makes it the priority and the first Marine fires on it, taking the first Zergling down to 29 hp. The second Zergling acquires the AI base as a target and moves down and left. It can’t move on a diagonal and the blocked square above it makes the shortest path 9 steps to the base. There would be an equal length path going right that would be higher priority, but the first Zergling is blocking it. The third Zergling and the second Marine do nothing as it is their first turn. The board looks like this.

\*mm\*\*\*\*

\*Xz\*\*\*\*

\*\*\*##\*\*

\*\*\*\*z#\*

\*#\*\*$\*\*

\*\*\*\*z\*\*

**Step 4:** Ke$ha’s base makes a Zergling to the right. The AI base does nothing. First Zergling attacks AI base taking it down to 95 hp. First Marine fires on first Zergling taking it down to 23 hp. Second Zergling moves left then up. Third Zergling move left twice. Second Marine acquires the first Zergling and fires on it, taking it down to 17 hp. The board looks like this:

\*mm\*\*\*\*

\*Xz\*\*\*\*

\*\*\*##\*\*

\*\*z\*\*#\*

\*#\*z$z\*

\*\*\*\*\*\*\*

### Sample Input:

2

5

\*\*\*\*#

\*\*$#\*

\*\*#\*\*

\*#\*X\*

\*\*\*\*\* 8

\*\*\*\*\*

\*\*\*\*\* ##X##

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*$\*\*

\*\*\*\*\*

### Sample Output:

AI wins!

Ke$ha wins!

# Glitter on the floor

Ke$ha’s manager has to prepare the next venue for a concert, and glitter is an absolute necessity. Unfortunately, there is a glitter shortage in the current economy, and he must choose the largest possible venue such that the floor can be covered with his current supply of glitter.

### Input

The first line of the input will contain a single number indicating the number of input sets, 0 < N <

1. Each input set will consist of one number that indicates the number of pieces of glitter he has. Notice that, although there is a glitter shortage, these inputs may be large enough that they will not fit in a long and only a Java BigInteger can hold some. Each piece of glitter is one millimeter squared.

### Output

For each data set, output a single line containing the largest number so that its square doesn’t exceed the number of square millimeters that can be covered by the current glitter supply. Note that this number may also be bigger than the maximum long.

### Sample Input

5

5

38

10002

1270397

1238940234

68476938463658697049374768765949372126265475

### Sample Output

2

6

100

1127

35198

8275079362982490275505

### Appendix

To input a Java BigInteger:

Scanner sc = new Scanner(System.in); BigInteger a = sc.nextBigInteger();

Java BigInteger library supports some basic integer operations: BigInteger a = new BigInteger(“5”); // a is 5 a = a.add(new BigInteger(“2”)); // a is now 7

a = a.subtract(new BigInteger(“3”)); // a is now 4 a = a.multiply(new BigInteger(“4”)); // a is now 16 a = a.divide(new BigInteger(“2”)); //a is now 8

Finally, one can use a.compareTo(b) to compare BigInteger a with BigInteger b.

# Kick ‘em to the curb

Some of you may remember our good ol’ friends, Boudreaux and Thibodeaux. Turns out, Ke$ha also knows them! What a coincidence! Small world, huh? Anyways, unfortunately Ke$ha only has two tickets to the Rolling Stones concert, so she has to choose which of her friends to take. How is Ke$ha going to decide who to take, you ask? Well obviously she’s going to bring the one who looks the most like Mick Jagger!

### Input

The first line of the input will contain a single number indicating the number of input sets. Each input set will consist of three lines. The first line of each set will be a space separated list of words that describe the qualities of Mick Jagger. Each quality will not contain spaces. The next line will be a list of Boudreaux’s qualities, and the third line will be a list of Thibodeaux’s qualities. Boudreaux’s and Thibodeaux’s qualities will be in the same format as Mick Jagger’s qualities. There will be no more than twenty qualities on each line to describe each person. All qualities will be lower-case.

### Output

For each data set, output either “Boudreaux” or “Thibodeaux”, indicating which of the pair has the most qualities like Mick Jagger. In the case of a tie, output “Sneak one in!”

### Sample Input

3

bigmouthed singer songwriter songwriter singer

bigmouthed

unsatisfied shelter-seeker tall shelter-seeker unsatisfied

rich sexy dinosaur

rich friends-with-keith-richards dinosaur rich

**Sample Output** Boudreaux Sneak one in! Thibodeaux

# I want to ride my Golden Bicycle

Like most good and sanitary Unicorn loving people, Ke$ha brushes her teeth every morning. Then, being the eco-friendly person that she is, Ke$ha rides her golden bike to work, or the club, or wherever she goes on a daily basis. However, it’s probably a good idea if Ke$ha waits a little bit between brushing her teeth and trying to operate a moving vehicle, so your job is to help her determine the minimum amount of time she needs to wait before leaving for the night and not coming back.

### Input

The first line of the input will consist of a single number, N, indicating the number of input sets to process. The next N lines will contain a number, the threshold, followed by a space, then a formula with one variable, t, indicating time in seconds. The formula will not contain any spaces. The only operations in these formulas will be addition (+), subtraction (-), multiplication (\*), and division (/). All formulas will be linear in t and will not include parentheses. A preceding minus sign can denote negation. There will never be a preceding plus sign.

### Output

The output will contain a single number indicating the lowest, non-negative value of t, in seconds, for which the answer to the formula is less than or equal to the threshold value. The output must be accurate to within 0.0001 seconds, and the value of t will not exceed 3600.

### Sample Input

2

5.5 -5\*t+17

95 t+9\*-t+15\*8.3

### Sample Output

2.3

3.6875

# Glitter Glitter Everywhere!

As you probably know, Ke$ha loves glitter. One thing about glitter, though: it gets EVERYWHERE! Given that lots of people frequent her favorite club, Ke$ha wants to know to how many people she will transmit glitter to during her night.

### Input

The first line of the input will consist of a single number, 0 < N < 100, indicating the number of input sets to process. The next N data sets will have lines as follows:

* + M R, where 0 < M < 100 is the number of people in the club (with the people indexed from 0 to M-1) and 0 < R < 1000 is the number of pairs of people that will be dancing with each other that night,
  + R lines as follows:
    - X Y, where X and Y are either integers between 0 and M-1 or a dollar sign ($) to denote Ke$ha.

Notice that the R lines are placed in order of when people danced, so if 0 dances with 1 before 1 is glitter-fied, then 0 may not be glitter-fied yet.

### Output

The output should be a series of N lines denoting the number of people with glitter on them by the end of the night (not including Ke$ha)

### Sample Input

3

|  |  |
| --- | --- |
| 2 | 3 |
| $ | 0 |
| 0 | 1 |
| $ | 1 |
| 3 | 4 |
| 0 | 1 |
| $ | 2 |
| 0 | 2 |
| 0 | $ |
| 9 | 9 |
| $ | 0 |
| 0 | 1 |
| 1 | 3 |
| 3 | 2 |
| 2 | 4 |
| 6 | 7 |
| 4 | 5 |
| 5 | 8 |
| 8 | $ |

### Sample Output

2

2

7

# Is This Place About To Blow?

The Owner of Club Cannibal, Mr. D Saur is about to open up for a Friday night when it filters through the rumor mill that Ke$ha will be at the club tonight. This causes the Mr. Saur to worry about the possibility of his club getting “blown.” He has asked you to develop an program that will analyze the playlist for the night and the expected number of people dancing per song to determine when the club would be in danger of getting blown. He has also provided you a Hottness Factor Table to determine when the club may blow as shown below:

|  |  |
| --- | --- |
| Hottness | Number of People Required to Blow |
| AWESOME SAUCE! | 50 |
| Pretty Good | 150 |
| Alright | 300 |
| Lame | 500 |
| Only For Dinosaurs | 1000 |

### Input

The first line of the input will consist of a single number, 0 < N < 100, indicating the number of input sets to process. Each input set will consist of three parts:

* + It begins with a line tell you how many songs the DJ has, 0 < S < 100, and how many songs are on the playlist for the night, 0 < P < 100.
  + The second part will S lines of the songs that the DJ has, each with a hotness factor.
  + The third part will have P lines of songs and the anticipated change in people on the dance floor for each song (ex. a number 30 would imply when that song plays 30 people join the dance floor, whereas -10 would imply 10 people leave the dance floor during the song)

Song names will be in double quotes and can contain spaces or other characters, but not double quotes.

### Output

The output should be a series of N lines denoting for each set during which song the club will “blow” or if the club is not in danger of being blown simply print a line stating “False Alarm”

### Sample Input

5

3 5

"Tik-Tok" Alright

"Back in Black" Pretty Good

"Strength of Strings" Only For Dinosaurs "Tik-Tok" 40

"Back in Black" 40 "Strength of Strings" 40 "Back in Black" 40

"Tik-Tok" 40

5 5

"Sexy and I Know It" Alright "Tears from the Moon" Alright "Womanizer" Pretty Good "Precious" Pretty Good "Hangover" Lame

"Sexy and I Know It" 100 "Tears from the Moon" -15 "Womanizer" 10

"Precious" -50

"Hangover" -5

**Sample Output** Back in Black False Alarm

# Ke$ha’s Purse!

Ke$ha is getting ready for a night of fun at her favorite club with her friends Boudreaux and Thibodeaux, but the problem is, she can’t figure out what she needs to pack in her purse for the night.

Each item that she might want to put in her purse has a specific weight and a numeric importance. The importance is measured on a scale of 1 to 100, with 100 being highly important and 1 being unimportant. For example, for Ke$ha, things like glitter have very high importance level of 100 while things like her house keys do not, and would have an importance value of 1.

Your job is to figure out what Ke$ha should bring with her in her purse, making sure that you fill up the maximum cumulative importance in items. The list should be presented in ASCIIbetical order by item names. (This is what you get by default when you compare Strings.) If there is a tie in cumulative importance, the list with more total items take priority. No inputs will require a tie breaker beyond that.

### Input

The input for this problem will begin with an integer N (0 < N < 100), the number of data sets. Each data set will contain the following:

* + A line containing an integer X (0 < X < 15), which will be the number of items Ke$ha has to choose from.
  + A line containing an integer Y (0 < Y < 50) representing the number of pounds her purse can hold.
  + X lines containing the weight (in pounds) of the item, the importance value of the item, and the name of the item. Weights and importance values are integers.

### Output

The output for each data set will consist of a sentence saying:

“Ke$ha will carry W in her purse”

With W being all the items she will carry, separated by commas. The items should appear in ASCIIbetical order.

### Sample Input

2

2

5

3 90 cowboy boots

5 1 book 6

10

3 100 bag of glitter

2 60 breathmints 1 40 $20 cash

4 80 baby pig

1. 50 toothbrush
2. 1 house keys

### Sample Output

Ke$ha will carry cowboy boots in her purse

Ke$ha will carry baby pig, bag of glitter, breathmints, toothbrush in her purse

# Until the Police Shut Us Down

At the club, Ke$ha is gonna fight ‘til she sees the sunlight unless it gets too crunk and the police

shut the place down. Fortunately, she is pretty familiar with the club’s back door, so if she can get out of the club fast enough, the party won’t stop. The only thing getting in her way are the tables. The police are hindered by the tables, as well, but one officer will attempt to make it to the back door in time to stop Ke$ha. (**NOTE**: Both Ke$ha and the officer can only move up, down, left, or right.

There are no diagonal moves, and it is possible that either person can be blocked in such a way that it is impossible to reach the back door.)

**Input:** The input for this problem will begin with an integer N (0 < N < 100), the number of data sets. Each data set will contain the following:

* + A line containing two integers, X and Y, the length and width of the rectangular club building (X and Y are both less than 20)
  + X lines containing Y characters, to describe a map of the club. The characters denote the layout of the club, as well as the location of the police officer headed toward the back door. Areas of the club are denoted as follows:
    - a dash, - , will denote a clear space in the club.
    - a capital “B” will denote the back door of the club
    - a dollar sign, $, will denote Ke$ha’s initial position.
    - a lowercase “o” will denote the initial position of the officer headed toward the back door
    - a lowercase “x” will denote the location of tables and other obstructions such as incapacitated clubgoers

**Output:** For each data set, the program should print out a single line, stating whether Ke$ha made it out of the club. If Ke$ha exits through the back door before the police officer, or the officer can’t get to the exit, print “The party don’t stop”. If the police officer gets to the exit before she does, or if they reach the exit at the same time, print “Put your hands up”.

## Sample Input

2

5 5

B--x-

-x--x

x$---

o x

6 7

$-x-B-x

-x-x---

------x

xx--o--

----xx-

xx-xx--

## Sample Output

The party don’t stop Put your hands up

Traveling Salesperson Pete (TSP) always makes sure to give his customers receipts for their purchases. Unfortunately, since he’s always in a hurry to get to his next destination, TSP’s copies of the receipts are all out of order. He has until tomorrow to get the receipts in order before he turns them in to his boss, Manager Mike. Mike wants to know what items TSP made the most money on. Write a catalog of the receipts for Mike to look at so he can see the total amount of money made from the sale of each kind of item.

**Input:** The first line of input will contain a single integer, 0< N ≤ 10, indicating the total number of receipts. Each receipt will contain a single integer, 0 < K ≤ 10, indicating the number of items. There will then be K lines, each containing an item. Item lines will have name, price and quantity sold, in that order and separated by spaces. Each item name will have no white space.

**Output:** A list using the same datatype that has the items in order (largest to smallest) based on the total revenue (before factoring in cost). The output of this problem should consist of a line for each item with the name, price ($), number of items sold, total revenue earned for that item. Item totals will be unique. There should be a final line containing the total revenue for all items.

## Sample Input:

2

3

Leg\_Lamp 7.40 5

Daisy\_Red\_Ryder\_Air\_Rifle 64.14 2

Bunny\_Suit\_Pajamas 120.00 1

2

Deluxe\_Fruitcake 34.19 1

Red\_and\_Green\_Foil\_Gift\_Wrap 19.99 4

## Sample Output:

Daisy\_Red\_Ryder\_Air\_Rifle 64.14 2 Revenue: 128.28

Bunny\_Suit\_Pajamas 120.00 1 Revenue: 120.0

Leg\_Lamp 7.40 5 Revenue: 37.0

Total Revenue: 285.28

Red\_and\_Green\_Foil\_Gift\_Wrap 19.99 4 Revenue: 79.96

Deluxe\_Fruitcake 34.19 1 Revenue: 34.19

Total Revenue: 114.15

Traveling Salesperson Pete (TSP) is a serious businessman, with a serious agenda. He doesn’t have time to deal with such petty matters such as speed limits and other traffic laws as his trips to and from places of business often need to be very fast. In order to avoid speeding tickets, in his tiny town of Toontown, USA, he has talked to an old buddy of his with a police radio scanner, who has, for an inconsequential fee, agreed to let TSP in on where the current speed traps are, so that he can avoid them. Using this information TSP wants to know if he can finish his route in time to make it home to watch his favorite television show without getting a traffic ticket. TSP’s car can travel up to 120 mph, and accelerates instantaneously to his required speed (being a salesman for ACME has it’s perks).

**Input:** The first line of the input will contain a single integer, 0 < N ≤ 20 indicating the number of input sets. Each input set will first contain a single integer, 0 < K ≤ 12 indicating the number of different roads TSP must visit on his route. The next line will be another integer with the time in minutes that TSP needs to reach the end location. Then, for the next K lines, there will be a “C” if there is a cop on this road, or a “N” if there is not. On the same line will also be the distance TSP has to travel and the name of the road which will contain no spaces and the integer speed limit on the road which will be greater than 0.

**Output:** for each input set you must return Yes if TSP is able to complete the route in time, or No if he cannot.

## Sample Input:

2

4

60

C 20.0 Cedar 60

N 10.5 Straightshot 30

N 20.0 Powderkeg 45

N 10.8 Charleston 30

1

20

C 30.0 Charleston 30

## Sample Output:

Yes

No

On his way out for a sale, Traveling Salesperson Pete (TSP) spots the car of his biggest competition Bandit Keith on the side of the road with a smoking radiator. After pulling over TSP sees that the car is abandoned, obviously Bandit Keith has been forced to walk to town to get help. Seeing an opportunity to expand his profits, TSP quickly looks through Bandit Keith’s trunk for items of value. Knowing that he has a limited amount of time until Keith returns and limited space in his trunk, TSP needs to figure out how much of a profit he can make given the items in Bandit Keith’s car.

**Input:** The first line of the input will contain the number of inputs, 0 < N ≤ 10. The first line of each input contains three integers, S, T, and I. S indicates the amount of free space in TSP’s trunk. T indicates the amount of time, in minutes, that Pete has until Bandit Keith returns. 0 < I ≤ 15 indicates the number of items in Bandit Keith’s trunk. Each item will be a new line consisting of the name of the item, the value of the item, the space the item takes up and the time, in minutes, it will take to move the item.

**Output:** For each input set the output should be the maximum value that TSP can get from the items he moves from Bandit Keith’s car given the time and space restraints.

## Sample Input:

2

5 10 4

Kerrigan\_Dolls 2 2 1

PowerPuff\_Vitamin\_Pills 50 1 1

Yogi\_Ties 10 1 2

|  |  |
| --- | --- |
| Bunny\_Slipper\_Socks 3 | 2 3 |
| 30 30 2  Chinese\_Checkers 2 50 | 14 |
| Tungsten\_Steel\_Pliers | 30 1 3 |
| **Sample Output:** |  |
| 63 |  |
| 30 |  |

Traveling Salesperson Pete must go through a neighborhood with a train track running across its length. Pete needs to get to the next city on his schedule as soon as possible; he’s running late due to an inefficient program that wrote down his itinerary. Luckily, Pete is an amazing judge of speeds and can tell how fast a train is moving just by looking at it. Given that knowledge and the distances that both he and the train have to travel, he needs to figure out if he will make it across the tracks or not.

**Input:** Input will first consists of a number 0<N<100, this is the number of input sets that you will be provided. Each input set will consists of 4 lines, each with a single, positive integer value. The first line will contain the speed of Pete’s car in mph, the second line will contain the distance from Pete’s car to the train tracks in miles, the third line will contain the speed of the train in mph, and the fourth line will contain the distance from the head of the train to the intersection of the tracks and the road in miles.

**Output:** Output should be Yes if Pete will reach the tracks before the train and No if the train will reach the road before Pete. The train wins ties.

## Sample Input:

2

50

100

60

300

30

100

50

50

## Sample Output:

Yes

No

Traveling Sales Pete used an inefficient path to sell his goods in various cities and ended up driving for a long time. He realized today that he needed to drive less and spend more family time. He said out loud, "I can finally go home, nothing is going to delay me from seeing my family." Then the heavy rain started.

The rain is making it difficult for Pete to read the exit signs. He needs to pick the right exit to get home to his family as soon as possible to set things right. The tears and the rain make it difficult to see every character in the road signs. As he drives near each exit, he manages to only catch a few characters of the signs.

Given the characters Pete sees of each sign of the exits he takes and his map information, determine if it is possible that Pete can go home safely and see his family.

**Input:** The first line of the input will contain a single number, 0 < N < 20 indicating the number of input sets. The The first line of each input set will have two city names representing where Pete is and where home is. That will be followed by a line with a number, 0 < R ≤ 40, for the number of roads on the map. The following R lines will each have two words for two cities connected by a highway. Note that these roads are all two way. These will be followed by another number, 0 < S ≤ 8, telling how many signs Pete saw on his trip. The input for each set ends with S lines giving what he saw. Characters he couldn’t make out are shown as hyphens. Signs that are seen will always be possible road combinations.

**Output:** The output, for each input set, should be “Maybe home.” if it is possible that the set of exits Pete took got him home and “Definitely lost.” otherwise. Note that Pete sees a sign for every exit he takes. So he will always see a sign in going from one city to another.

## Sample Input:

2

Austin Houston 4

Austin Dallas Dallas Waco Houston Temple Waco Temple

2

D­ll­s

W­c­

Austin Houston 2

Austin Houston Austin Halifax 1

H­­­­­­

**Sample Output:**

Definitely lost.

Maybe home.

Traveling Salesman Pete ends up in a confrontation with his rival, Bandit Keith. Like all sane businessmen, they decide to settle their dispute in the only way they know how. A children’s card game. Each player is dealt 5 cards. Each card is either a monster card, with an attack strength, or a trap card. The winner is the player with the highest total monster strength in their hand, however each trap card will cancel the strength of the opponents strongest remaining card. Pete, not one to leave things to chance, has a card up his sleeve when he plays these games. He has the option of substituting one of his cards for his hidden card if it will help him not lose.

**Input:** The first number of the input will be the number of sets. The next 5 lines will contain the name of the card, no space between words, and a number, the attack strength, if it’s a monster card or the word “Trap” if it’s a trap card. The next card will be Pete’s hidden card. The last 5 cards in each input set will be the 5 cards in Bandit Keith’s hand, following the same format as the other cards.

**Output:** The output should be “Win” if Pete will win with his original hand, “Lose” if Bandit Keith will win with his original hand even if Pete cheats, “Tie” if Pete will tie against Keith without cheating and “Cheat” if Pete will not lose ­ win or tie ­ if he switches one of his cards with the card in his sleeves.

## Sample Input:

2

The\_Dark\_Magician 500

Toy\_Soldier 100 Prison\_of\_Bone Trap Heaven\_Cannon 300

Magician’s\_Apprentice 200 Sandpit Trap Goblin\_Archer 400

Goblin\_Techies 600

Golden\_Warrior 100

Panda\_Monk 400 Sarlak Trap Doctor\_Hooligan 300

Mister\_Freeze 300

Dark\_Bell\_Curve 100

Mythical\_Hair 100

Rambunctious\_Rhino 700

Dull\_Secret\_Agent 100

Swiper\_Fox 100 Knapsack\_Of\_Holding Trap Jumping\_Magic\_Beans Trap Lukewarm\_Water Trap Ornery\_Owl 200

## Sample Output:

Cheat

Lose

Traveling Salesperson Pete (TSP) is happily married man. Well usually. Right now his wife is upset that he spends so much time on the road and is demanding that he take her on vacation to Las Vegas.

Since he values his life, TSP is doing whatever he can to make this happen. Given a list of TSP’s inventory, the monthly orders that he gets, his monthly expenses, and the cost of the vacation you need to determine the number of months that it will take TSP to save up enough money to take the vacation. It is also possible that he will be unable to save enough money or it will take too long and his wife’s patience will expire.

**Input:** The first line will contain a single number, 0 < N < 20 indicating the number of input sets. The first line of each input set contains an integer, 0 < M ≤ 10 indicating the number of items in TSP’s catalog. The next M lines will be the items that TSP sells in alphabetical order by the name of each item, and the cost to TSP to order the item. There will then be a number of sales, 0 < K ≤ 10, of items each month. This will be followed by K lines of sales. Each item sale will consist of the name of the item sold, the number of items sold, and the value of each item individually. This list will not be in a specific order and can contain multiple orders for the same item or no orders for some items. The next line will be the expenses that TSP needs to pay each month. The final line will be the cost of the trip.

**Output:** The number of months it will take to earn the required amount. If TSP is unable to take a vacation, or if it will take longer than 12 months return “R.I.P.” instead.

## Sample Input:

2

2

Hearing Aid $50 Stuffed Penguin $30 2

Hearing Aid 5 $200 Hearing Aid 3 $150

$400

$5000 1

Jumbo Jet $30 1

Jumbo Jet 1 $15

$200

$1000

## Sample Output:

8

R.I.P.

Pete is addicted to Pizza. He is so addicted that he can only travel for so long without eating a slice. His wife has begged him to go to a professional but he refuses. He leaves his house in a puff and doesn’t get to plan his travel route as he sells his goods like he usually does.

Given his itinerary, along with the distance between pizza stores and the number of pizzas his car can hold, will TSP be able to complete his trip or will he give up before finishing due to a lack of pizza? Each pizza box contains 8 slices of pizza and TSP needs a slice of pizza every so often.

**Input:** The first line will contain a single number, 0 < N < 20 indicating the number of input sets. The first line of each input set will have four integer values. These represent the number of pizza boxes Pete’s car can hold, the number of miles TSP can travel for each slice of pizza he eats, the number of pizza stores on his route (0 < S ≤ 10), and the total distance he has to go on his trip. Then the S lines after that will have the name of each pizza store followed by distance Pete needs to travel to get to it from the previous location.

The first pizza store will always have a distance of 0. Names of pizza stores will contain no spaces.

**Output:** Each line of the output should say “BLISS” or “AGONY” depending on if Pete manages to obtain enough pizza for each input set. “BLISS” if there is enough pizza so that he can finish his itinerary or “AGONY” if he stops trying because he runs out of pizza.

## Sample Input:

2

4 10 2 520

Pizza\_Palace 0

Papas\_Pizza 240

3 2 4 110

Lisa\_Pizza 0

Little\_Ceaser 20

Testing\_Pizza\_Palace 50

Johns\_Dine\_And\_Dash 30

## Sample Output:

BLISS

AGONY

**Pete’s Peddlers**

Pete’s empire is expanding and he is finding it hard to personally make deliveries to all of the different cities. The items he sells all arrive in his home city and it would really help to just get things closer to where they need to end up going. Pete is far too cheap to hire real delivery drivers to do this. Instead he has come up with the idea of paying High School students sub­minimum wage to ride their bikes and transport items for him. Even with this low cost approach, Pete still trying to further optimize his costs.

Pete pays the students per mile that they ride between cities. He doesn’t really care how far the items go when he isn’t driving them, he just needs to be able to get items from the source city to every other city. So he wants you to find the roads to include in order to spans all the cities, but with the minimum total distance. For this problem you need only tell him the length of the sum of the roads in miles.

**Input:** The first line of the input will contain a single number, 0 < N < 20 indicating the number of input sets. Each input set will contain a single number, 0 < K ≤ 20 indicating the number of cities TSP needs to connect. There will then be K lines, each containing the name of one city followed by the names and distances from that city to all nearby cities that can be ridden to on a bike. The distances will be integer values and will be symmetric. The first city will always be his hometown, and each city name will only be a single word.

**Output:** The output for this problem will be one line per input sets with the minimum number of miles of road that he needs to pay the students to cover.

|  |  |  |
| --- | --- | --- |
| **Sample Input:** |  | |
|  |
| 2 |
| 3 |
| Toontown Westford 3 | Georgetown | 5 |
| Westford Toontown 3 | Georgetown | 4 |
| Georgetown Toontown | 5 Westford | 4 |
| 5 |  |  |

Toontown Paton 4 Westend 4

Paton Toontown 4 Anterson 3 Paulson 5

Westend Toontown 4 Paulson 10 Anterson 5

Anterson Paulson 8 Paton 3 Westend 5

Paulson Paton 5 Westend 10 Anterson 8

## Sample Output:

7

16

**Distribution Difficulties**

Pete has been expanding his courier network and he needs to know if the current network can handle the orders he is getting. All of his orders come in at one city and go out to the others. He is paying kids to bike things between cities and he knows how much weight can go between each pair of cities in his network each day. He doesn’t care exactly how long it takes for things to get from the delivery spot to their destination, as long as the courier network can keep up so that in a steady state there aren’t items piling up in any given city.

You are given the roads he has couriers working on and how many pounds of inventory can be carried down each road per day. These have a direction as Pete has arranged for couriers to only carry inventory one way and just ride back not carrying anything. You also have how many pounds of items are to be sold in each city each day. Assume the number of pounds received in Pete’s home town is equal to the total orders. Your program needs to tell Pete if the courier network can handle the orders.

**Input:** The first line of the input will contain a single number, 0 < N < 20 indicating the number of input sets. Each input set will start with an integer, 0 < K ≤ 20, indicating the number of cities in the courier network.

There will then be K lines, each containing the name of one city followed by the number of pounds of items that city need each day, 0 < P < 1000, and connection information. The connection information is in the form of city names and courier capacities from that city to all nearby cities with connecting roads. The first city will always be his hometown and it will have zero pounds of orders to consider. Each city name will only be a single word. There will never be couriers running in opposite directions between two cities.

**Output:** Each input set will have a single line of output. If the network of couriers can handle the flow of products print “SUCCESSFUL”. Otherwise you should print “TOO HEAVY”.

## Sample Input:

2

3

Toontown 0 Westford 3 Georgetown 5

Westford 3 Georgetown 4

Georgetown 2

5

Toontown 0 Paton 4 Westend 4

Paton 6 Anterson 3

Westend 2 Paulson 10 Anterson 5

Anterson 4 Paulson 8

Paulson 3 Paton 5

**Sample Output:**

SUCCESSFUL

TOO HEAVY