Frisco FirstBytes Programming Contest

October 15, 2016

NOVICE PROBLEM SET

DO NOT OPEN THE PACKET UNTIL TOLD TO DO SO.

The final Novice winner will be decided by the total number of Novice and Advanced problems solved. Novice teams can receive credit for solving Advanced Problems.

G Old Traditions Never Die!

Data File: G.txt

It's not really talked about but StarFleet Academy had a football team. And of course Captain Kirk was on the team, as the quarterback (what else?). Scotty was the equipment manager, Dr McCoy was the trainer and Mr. Spock, and well Mr. Spock didn't see the logic in the game.

Now StarFleet still kept to the tradition of the mascot, Ensign Redshirt, of course, doing 1 pushup for every point the team scored. So after the first touchdown Ensign Redshirt does 7 pushups. For the second touchdown, Ensign Redshirt does 14 pushups.

Your job, write a project that for a given number of points scored in a game print out how many pushups Ensign Redshirt does.

Input

The input contains several test cases. The first line of each test case contains an integer N (1<=N<=100) indicating the number of scores earned in the game. The following N lines contain the scoring record of the Captain Kirk's team in the order they were scored on each line. Touchdowns are worth 7 points; field goals, 3 points; and safeties, 2 points.

Output

After each score the team score is updated and Ensign Redshirt will do a pushup for each point on the scoreboard. Output how many pushups Ensign Redshirt has completed for the whole game.

Sample Input

```
1
touchdown
3
touchdown
touchdown
touchdown
5
touchdown
field goal
touchdown
safety
touchdown
0
```

Sample Output

```
Ensign Redshirt does 7 pushups!
Ensign Redshirt does 42 pushups!
Ensign Redshirt does 79 pushups!
```

H Touching Orbits?

Data File: H.txt

Even in the age of very advanced computers, cadets at StarFleet Academy must learn how to calculate orbits by hand. After all it seems as though the duotronic computers on starships, especially the Enterprise, have a disturbing tendency to behave strangely.

So a typical challenge for newly enrolled ensigns, given the coordinates of the center of a ship's orbit and the radius of that orbit and the coordinates and radius of another orbiting object such as an asteroid, determine whether the orbits overlap, don't overlap, or are tangential. The orbits are in perfect circles.

Input

The first line of the input contains an integer *N* (1<=*N*<=100) which is the number of test cases. Each test case consists of 2 lines representing the ship and the other orbiting body, containing 3 integers each. The first two integers are the *x* and *y* coordinate (We will assume each is orbiting on the same plane) followed by *r*, the radius of the orbit. The range of x, y, and r is from -1,000,000,000 to 1,000,000,000.

Output

For each test case, determine whether the orbits overlap, don't overlap, or are tangential. The orbits are tangential if they are within 1 unit of each other.

Sample Input

```
3
10 10 3
10 6 1
8 8 3
8 4 2
25 25 20
0 0 5
```

Sample Output

```
The orbits are tangential.
The orbits overlap.
The orbits don't overlap.
```

I What Ship Shape?

Data File: I.txt

Well, Starfleet is looking at 4 new ship shapes Borg Cube, Tholian Ball, Whale Cylinder, and a Remulak Cone. But the Admiral in charge doesn't quite know what shape makes the best ship. What he does know is the volume the new ship shape needs to hold and is willing to look at all of them. Admiral Harambe will decide on the ship shape based on resulting measurements.

So, given a volume, calculate the dimensions for each ship shape to two decimals places so that the Admiral can decide on the new ship shape. The height and diameter of the Whale Cylinder are equal. The height and diameter of the Remulak Cone are equal.

$Cone\ volume = \frac{1}{3}\pi r^2 h \qquad Sphere\ volume = \frac{4\pi r^3}{3}$

Input

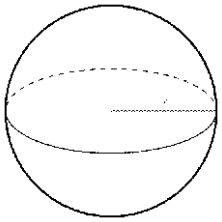
The input contains several lines of numbers which represent the desired volume of the new ship. The input is terminated by a zero, which should not be processed.

Output

The output for each test case requires the case number to be printed, followed by the measurements for the Borg Cube, Whale Cylinder, Tholian Sphere, and Romulak Cone.

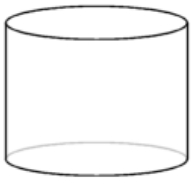
Sample Input

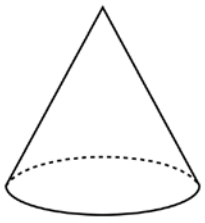
```
27.0
100239.345
0
```



Sample Output

```
Case #1
Borg Cube: 3.00m length, 3.00m width, 3.00m height
Whale Cylinder: 3.25m height, 1.63m radius
Tholian Sphere: 1.86m radius
Remulak Cone: 4.69m height, 2.34m radius
Case #2
Borg Cube: 46.45m length, 46.45m width, 46.45m height
Whale Cylinder: 50.35m height, 25.17m radius
Tholian Sphere: 28.82m radius
Remulak Cone: 72.61m height, 36.31m radius
```





J Who Won the Bat'leth Tournament?

Data File: J.txt

On Stardate 70254.9 Worf competed in the 2400th Annual Bat'leth Tournament on Qo'noS. However the scorekeeper just wrote down the winners and the order they placed in random scribbles in his Hoqra'. Your job, write the code that sorts and prints out the winners in the correct order and with the place they finished in listed first.

Input

The first line of the input contains the number of tournaments. Each tournament will start with an integer *P* (2<=*P*<=100) which represents the number of players in that tournament. Following are *P* lines, each containing the player's name and place. Player's name may have spaces and symbols, except dashes. The name and place are separated by space, dash, space: " **–** "

Output

The output should list the player's and their place as shown in the order they finished the tournament. Separate the tournaments with a blank line.

Sample Input

```
1
4
Worf - 2
Qe Qvuquq - 3
DaDey - 1
BuchnguQ – 4
```

Sample Output

```
1 - DaDey
2 - Worf
3 - Qe Qvuquq
4 - BuchnguQ
```

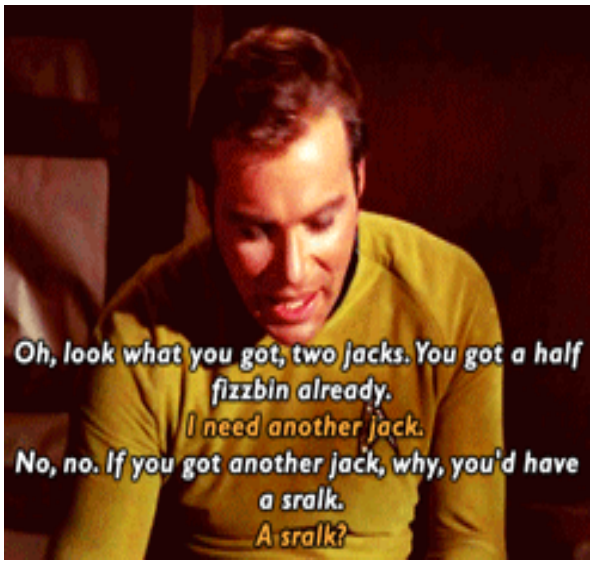
K Fizzbin

Data File: K.txt

On Sigma Iota II, Captain Kirk, Mr. Spock, and Dr. McCoy are being held by the henchman of Bela Okmyx. To buy time for Scotty to locate them on the planet and beam them out of trouble (as usual), Captain Kirk invented the game Fizzbin.

Each player is dealt a hand of eight cards. The deck is a standard deck of 52 cards, where each value is represented by an integer 1 to 52 according to the table below. There are no duplicate cards in the deck.

1 – 13	Ace, 1-10, Jack, Queen, King of Spades
14 – 26	Ace, 1-10, Jack, Queen, King of Diamonds
27 – 39	Ace, 1-10, Jack, Queen, King of Clubs
40 – 42	Ace, 1-10, Jack, Queen, King of Hearts



You are going to capture this famous game by programming the rules below. These are ranked in order from lowest to highest.

- A “Half-Fizzbin” is a pair of Jacks, Queens, or Kings of any suit.
- BUT if you have 3 Jacks, Queens, or Kings, then the player doesn’t have a “Half-Fizzbin”, but has a “Shralk” and is disqualified.
- A “Royal Fizzbin” is a King, 2, Jack, 6, two Queens and two Aces in any order. A Royal Fizzbin outranks a Half-Fizzbin of course.
- Finally the last card dealt in a hand could be a “Kronk” if it is an King, Queen, Jack, Ace, 2, 4, 6, or 8. And this card has a special effect. When a Kronk is the last card then Scotty successfully beams the party up and you should indicate that “KRONK! Scotty beams us up!” instead of evaluating the hand!

Input

The first line contains an integer *N* (1<=*N*<=50), the number of hands of cards. On each of the following *N* lines, a hand of cards will consist of 8 integers representing the cards in the order they were dealt, from left to right.

Output

For each hand of cards, print out the highest value ranking for each hand on a separate line.

Sample Input

```
4
13 26 6 2 9 8 5 5
11 26 6 2 9 8 5 5
11 26 6 2 9 8 5 8
26 13 39 2 9 8 5 7
```

Sample Output

```
Half-Fizzbin
Nothing
KRONK! Scotty beams us up!
Disqualified
```

L Circle of Friends

Data File: L.txt

Starfleet is a “small world” despite having a huge and diverse population spanning countless galaxies. Of course, many meet each other at Starfleet Academy, on board star ships, or on planets.

Your job is to determine how many nonintersecting circles of friends there are from a list of pairs of friends. Any friend of a friend is in the same circle of friends. For instance, Spock is friends with Sulu, and Sulu is friends with Uhura, then Spock, Sulu, and Uhura are in the same circle of friends.

Input

The input will consist of pairs of names. Each name will contain only alphabetic characters with no spaces. The pairs of names will be separated by a single space.

Output

Output a single integer, the number of nonintersecting circles of friends.

Sample Input

```
Spock Sulu
Sulu Uhura
Spock Kirk
Crusher Worf
Picard Worf
Chekov Uhura
Janeway Chakotay
```

Sample Output

```
3
```

M Planet Search

Data File: M.txt

The youngest of the junior Starfleet students learn everything in a very logical way. In their astronomy class they use word searches to learn the planets of the known universe. Instead of circling the answers (how archaic!) they must indicate the row and column of the first letter of each planet and the direction to read the planet's name using compass directions (N, NE, E, SE, S, SW, W, NW).

Input

The first line of input has 3 integers W, R, and C, where W is the number of planet names on the following W lines, and R and C are the number of rows and columns of the word search. Following the planet names are R lines, of C characters each separated by a single space, representing the word search. The word search consists entirely of uppercase alphabetic characters. All the planet names are alphabetic characters.

Output

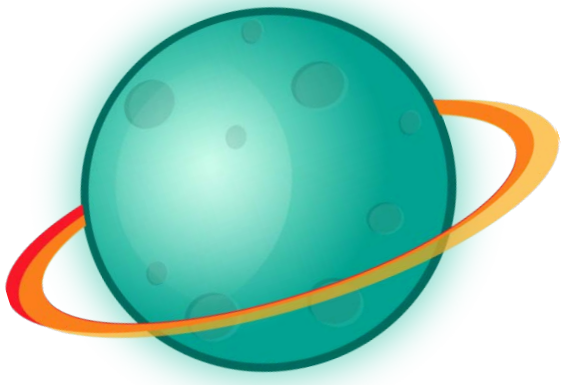
The output will contain one line for each planet. Each line will have the planet's name, followed by 2 integers for the row and column of the first letter, followed by the compass direction.

Sample Input

```
5 8 8
Akritiri
Braslota
Iconia
Nibiru
Quarra
I D N I K S B A
Q R B I B Q T T
U V I N B O B L
A E K T L I U F
R F K S I T R W
R Y A Q F R B U
A R M R K Z K W
B I C O N I A A
```

Sample Output

```
Akritiri 8 8 NW
Braslota 8 1 NE
Iconia 8 2 E
Nibiru 1 3 SE
Quarra 1 2 S
```



N Weight WHAT??

Data File: N.txt

The Enterprise is always beaming its crew down to different planets. What no one knows though the transporter has a fail-safe built in that doesn't allow beaming to a planet where the planetary force exerted on the crew members exceeds a certain factor.

So you are going to re-create the fail-safe mechanism by applying Newton's Universal Law of Gravity which states:

- Any two objects in the universe attract each other gravitationally...
- With a force that's proportional to the product of their masses, and...
- Inversely proportional to the square of the distance between them. (distance is measured from the center of the object - so if you're standing on Earth, you are about 6353 km away from it.
- Because this is only proportionality (not equality), you will need a constant multiplier - this is called G, the gravitational constant.

This gives us the remarkably simple formula:

$$force = G \times \frac{(\text{mass of first object} \times \text{mass of second object})}{(\text{distance between objects})^2}$$

This force is applied on both objects equally and in opposite directions, toward each other. The value of G is currently known to be about 6.67×10^{-11} which is why gravity is so weak - you can overcome the force of the entire planet just by jumping!

We're going to assume all planets are perfect spheres. This means you can find the volume of a planet, given its radius, with the formula $V = 4/3 \times \pi \times \text{radius}^3$ like a normal sphere. We'll also assume they are made of a material which has the exact same density everywhere - so a handful of material from one bit of the planet weighs the same as any other. This means, given a density (in kilograms per cubic meter), and using the volume you worked out, you can compute the mass of the planet with the formula mass = volume × density. Assume the units you are using are kilograms and meters.

Sorry, Klingons no chebs or cheb'a' – you have to do your own!

Now, in case you are new to physics, you may need to know a little bit about forces. Forces are measured in Newtons (N) and measure, essentially, how hard an object is pushing another object. The object could be pushing physically - eg. pushing a lawn mower - or via an elementary force, such as Earth's gravity pushing you toward it. They can all be measured in Newtons. The force of a planet on something due to gravity is called weight - which is not to be confused with mass, which is measured in kilograms and is a measure of how much matter something contains. As we saw before, the more mass two objects have, the greater the force they exert on each other. As gravitational force is dependent on

the product of the masses of both objects, an object will weigh more if either the object itself, or the planet, is heavier - which is why you weigh less on the Moon!

The challenge for you today is, given the dimensions of several planets and an object's mass, calculate how much force is applied on the object at the surface of the planet. Pretend the object is quite small for simplicity of your calculations.

Input

The first line will contain an integer M which is the mass of an object in kilograms. Followed by an integer N, which is the number of planets to evaluate. The following N lines contain data about each planet, which will be given as its name, its radius (in meters), and its average density (in kilograms per cubic meter) separates with a comma and space.

Output

The output will include the name of the planet followed by a colon and the weight in Newtons of the object if it were at the surface of each planet, rounded to the nearest thousandth.

Sample Input

```
100
4
Tantalus, 3104500, 5009
Reach, 7636500, 4966
Circumstance, 4127000, 4132
Tribute, 2818000, 4358
```

Sample Output

```
Tantalus: 434.467
Reach: 1059.536
Circumstance: 476.441
Tribute: 343.117
```