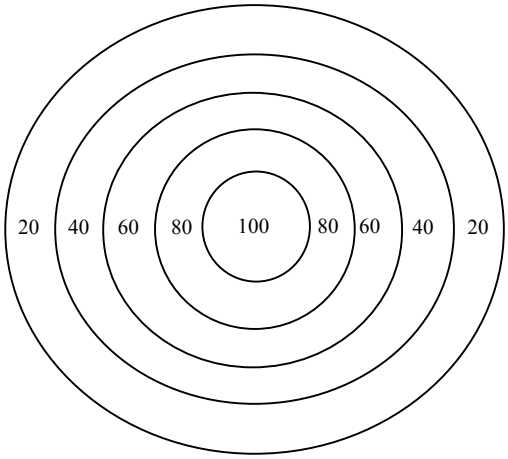
Problem
Darts

Input File: DartsIn.txt
Output File: DartsOut.txt
Project File: Darts

Because of the arguments over the scoring of the dart matches, your dart club has decided to computerize the scoring process. The dart board consists of concentric rings of radii 3", 6", 9", 12", and 15" with each ring given a point value as shown below.

A match is between two players, each player throwing three darts. Each dart thrown is awarded the point value of the ring in which it lands. If a dart lands exactly on a circle, it is award the higher (inner ring) point value. A dart landing outside of the outer circle is awarded zero points. A player's score is the sum of the points awarded for the three darts thrown. The player with the highest score wins the match.



Write a program that computes the scores of the two players in a match and then determines who, if anyone, wins the match.

Inputs

The input consists of one or more data sets, one data set per line. Each data set consists of six pairs of (x,y) coordinates that locate the six darts on a Cartesian plane whose origin is at the center of the inner circle. The range of x and y is -20.0<= x,y <= 20.0 . The first six number on a line locates player one's darts, and the last six numbers locates player two's darts. The input data sets are terminated when the first value in a data set is greater than 20.0.

Outputs

The output will be one line per match with the line containing three numbers: the score of player one, followed by the score of player two, followed by the number of the winning player (1 or 2). In the event of a tie, the word "tie" will be the third output on the line. The terminator line is not a valid match and should not be part of the output.

Sample input

```
0.0   0.0   -3.0   0.0   0.0   6.0   0.0   15.0   -12.0  0.0   0.0   9.0
0.0   0.0   -3.0   0.0   0.0   6.0   0.0   0.0    -3.0   0.0   0.0   6.0
4.0   -1.0  19.0   0.0   -4.1  -8.1  -4.0  -1.0   -4.0   -8.1  -4.1  -8.1
50.0  0.0   -3.0   0.0   0.0   6.0   0.0   15.0   -12.0  0.0   0.0   9.0
```

Sample output

```
280 120 1
280 280 tie
120 160 2
```

Problem
Grocery

Input File: GroceryIn.txt
Output File: Grocery.txt
Project File: Grocery

Cousin Tom owns a small convenience grocery store stocked with no more than one-thousand different items. Once a month his accountant gives him a new price list for all the items in the store. The list consists of an item's name and its new price, ordered in *ascending* order by price. Tom would like the list ordered in *descending* order by price, and would like it to include the amount of sales tax on each item.

Knowing your aptitude for computers and your affinity to ice cream, Tom as asked you to rearrange the list in exchange for all the ice cream you can eat. Naturally you have agreed. Write a program to rearrange the list and to calculate the sales tax on each item.

Inputs

The first line of input will contain the number of items in the price list, which will always be less than 1000. This will be followed by the price list with the items ordered in *ascending* order by price, one line of input per store item. Each of these lines will consist of an item's one word name followed by its price.

Outputs

The output will be the input price list re-ordered in *descending* order by price. There will be three outputs on each line: the items name, followed by the item's price, followed by the sales tax on the item assuming a sales tax rate of 8.5%. The sales tax should be rounded to the nearest penny. The outputs will be presented in three columns.

Sample input

```
6
gum         0.25
peas        0.76
carrots     1.84
potatoes    2.53
steak       4.59
coffee      5.93
```

Sample output

```
coffee      5.93        0.50
steak       4.59        0.39
potatoes    2.53        0.22
carrots     1.84        0.16
peas        0.76        0.06
gum         0.25        0.02
```

Problem
Parity

Input File: ParityIn.txt
Output File: ParityOut.txt
Project File: Parity

An even parity scheme is a technique for detecting errors in the transmission of digital information. Under this scheme the sender counts the number of on bits in the sequence of bits to be transmitted. Then a bit, called the parity bit, is added to the transmission. The value of the parity bit is chosen to make the total number of on (1) bits in the transition (including the parity bit) even.

For example, if the data to be transmitted is: 1011011 (decimal 91) and even parity was being used, the value of the parity bit add to the transmission would be 1 to bring the total number of on bits to an even value, 6. Assuming the parity bit was added on the left side of the data, the transmission would be 11011011.

Write a program to receive positive integers represented as seven bit unsigned binary numbers. An even parity scheme will be used to detect errors, with the left most bit being the parity bit. If the transmission is error free, output the decimal value of the transmitted unsigned integer, otherwise the output should be: "Re-transmit please".

Inputs
The first input will be the number of unsigned integers to be transmitted. Each subsequent line will contain eight bits: a seven bit unsigned binary number with an even parity bit appended as the left most bit.

Outputs
The output will be one line per unsigned integer transmitted. The line will contain the decimal value of the integer transmitted or, if an error has occurred, the line will contain the string: Re-transmit please.

Sample input
```
3
10100101
10101101
01011100
```

Sample output
```
37
Re-transmit please
92
```

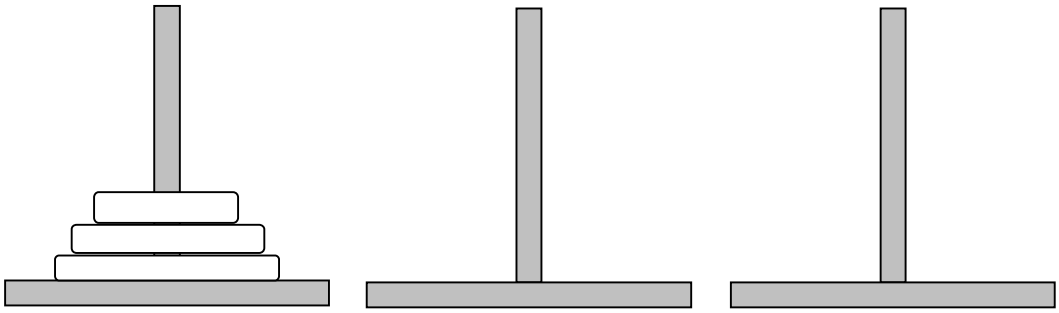
Problem
Rings

Input File: RingsIn.txt
Output File: RingsOut.txt
Project File: Rings

You baby brother has been given a game that consists of a collection of n donut like rings, all of different radii, and three towers to stack them on. The game starts out with all the rings stacked on tower 1 in radius order, with the biggest ring on the bottom. The objective is to move the rings from tower 1 to tower 2 without violating the game's two rules:

a) Rings must be moved from one tower to another, one at a time.
b) A large ring cannot be placed on top of a small ring.

The rings can be placed on any of the three towers as long as these two rules are not violated. The starting condition of the game for three rings, n=3, is shown below.



Although you baby brother seems to have no trouble playing the game with one or two rings, he's having trouble playing the game with three or more rings. You have decided to write a computer program to output the sequence of moves for an arbitrary number of rings.

Inputs
The first line of input consists of the number of games to be played. This line is followed by one input line per game. Each game's input line contain an integer: the number of rings for that game.

Outputs
The moves to complete each game, one line per move. Each games output will be separated by a blank line.

Sample input
```
3
1
2
3
```

Sample output
```
Move 1 ring from tower 1 to tower 2

Move 1 ring from tower 1 to tower 3
Move 1 ring from tower 1 to tower 2
Move 1 ring from tower 3 to tower 2
```

Move 1 ring from tower 1 to tower 2
Move 1 ring from tower 1 to tower 3
Move 1 ring from tower 2 to tower 3
Move 1 ring from tower 1 to tower 2
Move 1 ring from tower 3 to tower 1
Move 1 ring from tower 3 to tower 2
Move 1 ring from tower 1 to tower 2

Problem
Sequence

Input File: SequenceIn.txt
Output File: SequenceOut.txt
Project File: Sequence

A sequence of integers, called the 3n+1 sequence, is generated as follows: The first number in the sequence, n, is always a positive integer. The next number in the sequence, and all subsequent numbers in the sequence, is obtained as follows: if the previous number is even, divide by 2. If the previous number is odd, multiply be 3 and add 1. The sequence terminates when a term in the sequence calculates to 1.

It is conjectured, but not yet proven, that the sequence will terminate for every beginning integer, n. Still, the conjecture has been shown to be true for all values of n up to 1,000,000.

Inputs
The first term in the sequence, an integer between 1 and 1,000,000

Outputs
The sequence of integers, one integer per line.

Sample input
22

Sample output
22
11
34
17
52
26
13
40
20
10
5
16
8
4
2
1

Problem
Willie

Input File: WillieIn.txt
Output File: WillieOut.txt
Project File: Willie

Inspired by the recent Olympics in Turin, your great-uncle Willie has decided to get in shape with his eye on a gold medal in the 2008. Every morning he takes a brisk walk around his living room counting his strides as he walks. Last evening he called you quite upset that none of the Olympic events were measured in strides.

When you suggested he simply "convert", great-uncle Willie became quite agitated and responded that he "was not about to change his faith at this point in his life". To calm great-uncle Willie, you have promised to write a program that converts his strides to meters, so that he can target an event and realize his life long dream of becoming an Olympic champion (without changing his faith).

Inputs

The first line of the file will contain two real numbers: the length of great-uncle Willie's stride measured in feet, and the number of feet in a meter. This line will be followed by the length of great-uncle Willie's five most recent walks measured in strides, one walk per line.

Outputs

There will be six lines of output. The first five lines of output will be the length of each walk in meters, one walk distance per line. These will be in the same order as the input distances. The final output will be the average distance of great-uncle Willie's five most recent walks.

Sample input

```
2.1 2.28
103
200
120
500
31
```

Sample output

```
94.8684
184.211
110.526
460.526
28.5526
175.737
```