On their long trips the avengers like to kill time by doing the NY Times crossword puzzle. Giv
2 words, you will determine if they can be perpendicular in a crossword puzzle. If they can, yo
will show them horizontally and vertically intersecting. The first word will be the "across" and
the second word will be the "down" word. Here are some examples:

```
cross and green (the r's meet)
playing and dog (the g's meet)

 g          d
cross       o
 e      playing
 e
 n
```

X = across.indexO
for(int i=0

The intersecting letter will be the first letter they have in common (the first common letter in th
first word). If there is no common letter, print out "NONE." The maximum word length will b
15.

Input
The first line (N) will contain the number of data sets. Each data set consists of two words separated by "and." The longest word will be 15 characters long.

Output
For each data set, print out either "none" or the intersecting words. Separate data sets with a blank line.

String
Set String
String words
short
for(int i=0; i<tem
for( )

Constraints:
1<=N<=10

Example Input file

```
4
cross and green
playing and dog
meet and non
school and love
```

(Continued on next page…)

11. Fight

Program Name: Fight.java **Input File: Fight.dat**

A few of the Avengers have started playing video games. Particularly Thor likes to play Fortnite, however he hates how random the pump shotgun's damage is. He and noobslayer69 are fighting it out each with a pump shotgun and you are to determine the winner.

Input

Each data set will contain a single line being the random seed for the data set. The seed will generate random number between 1 and 200 being how much damage that shot did. Thor and noobslayer69 will exchange shots, meaning Thor shoots first and then noobslayer and so on. Each player starts with 200 health.

Output

Output the winner of the fight for each data set.

Example Input File

```
1641715317
346734862
```

Example Output to Screen

```
Noobslayer69 wins!
Thor wins!
```

(0, 198) {0, 200)

197 +1

+2

199

seeds.append( )

1, 199

Thor health = 200;

noob health = 200;

Random rnm = new Random();

for (seed : seed)

rnm.setSeed(seed);

while (Thor health > 0 && noob

noob health -= nextInt(19

if noob health <= 0

System.out.println(Thor wi

Thor health -=

(1, 200)

next Int(198)

[1, 198] +1

[2, 199]

198 / +2

(0, 197)

[2, 199]

10. Words

Program Name: Words.java **Input File: words.dat**

Antman has just come out of the quantum realm but something is wrong with him. He's speaking gibberish, and the Avengers need your help deciphering it. They have figured out that whenever he tries to speak he's saying his words backwards and with random numbers in them. Given his words please decipher the text for the Avengers.

Input
The data set will contain space separated words all on one line that need to be deciphered.

Output
Output the deciphered words in the order they appear in the dataset, all punctuation should remain the same.

Example Input File
```
ti2aW o8h1w si0 10so2nah21T???
```

Example Output to Screen
```
Wait who is Thanos???
```

String.replace([0-9], "")

for

String.split(" ")

String.replace([0-9]

~~Split[~~

StringBuffer fhat String = "";

9. Travel

Program Name: Travel.java **Input File: travel.dat**

Quill is doing a bit of space travel and needs to know the fastest route to his destination. Since he isn't that smart (don't tell him I said that) he needs your help to find that route.

Input

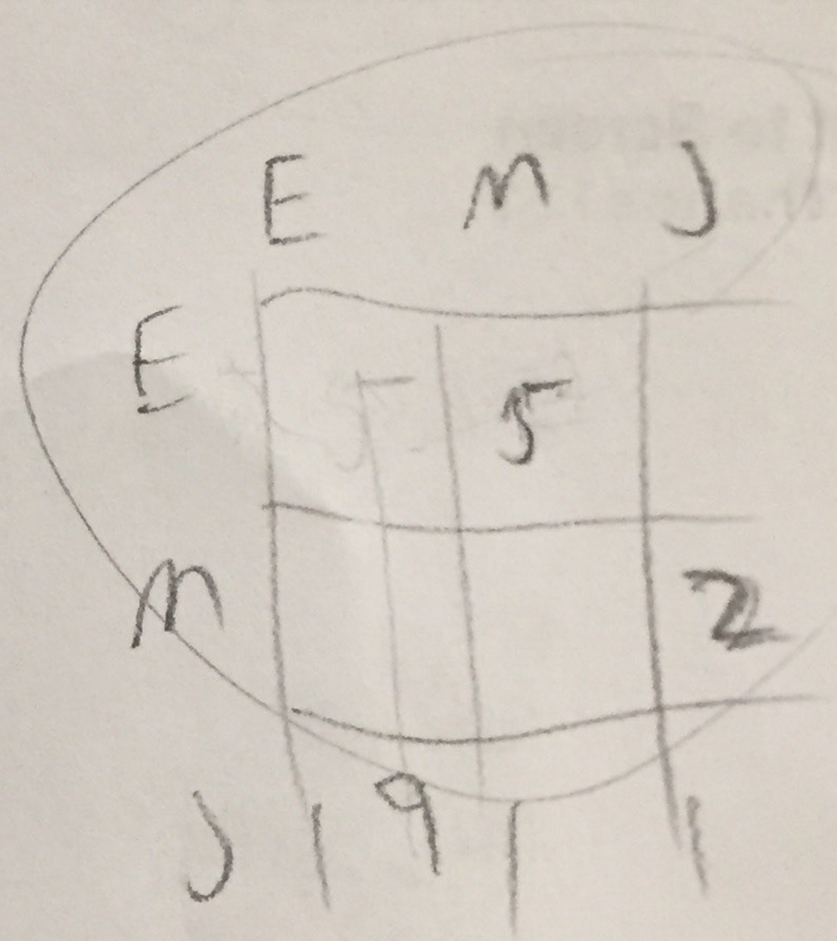
The first line will contain a single integer n that indicates the number of data sets that follow. Each data set will start with a single integer x denoting how many cosmic roads there are. The next x lines will contain two space separated strings s and d followed by an integer w denoting a path between s and d that takes w days to travel. The next line will contain two string indicating the start and finish planets.

Output

Output the list of cities from start to finish the format `fromplanet to toplanet` that takes the shortest path.

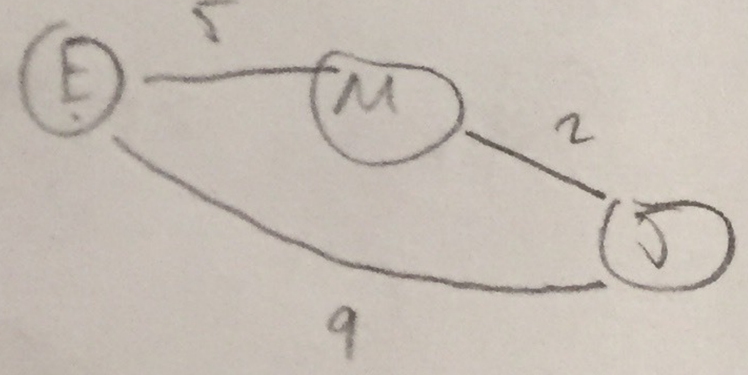
Example Input File

```
1
3
Earth Mars 5
Mars Jupiter 2
Jupiter Earth 9
Earth Jupiter
```



Example Output to Screen

```
Earth to Mars to Jupiter.
```



start @ start Node
Add all possible to priority queue

Path
Dist str
Dist int

Node:
~~Arraylist children~~
~~Arraylist dist~~

~~Array~~
~~Travel Data~~
~~Dist~~
~~Node~~
~~Prev~~

9. Travel

Program Name: Travel.java **Input File: travel.dat**

Quill is doing a bit of space travel and needs to know the fastest route to his destination. Since h isn't that smart (don't tell him I said that) he needs your help to find that route.

Input

The first line will contain a single integer n that indicates the number of data sets that follow. Each data set will start with a single integer x denoting how many cosmic roads there are. The next x lines will contain two space separated strings s and d followed by an integer w denoting a path between s and d that takes w days to travel. The next line will contain two string indicating the start and finish planets.

Output

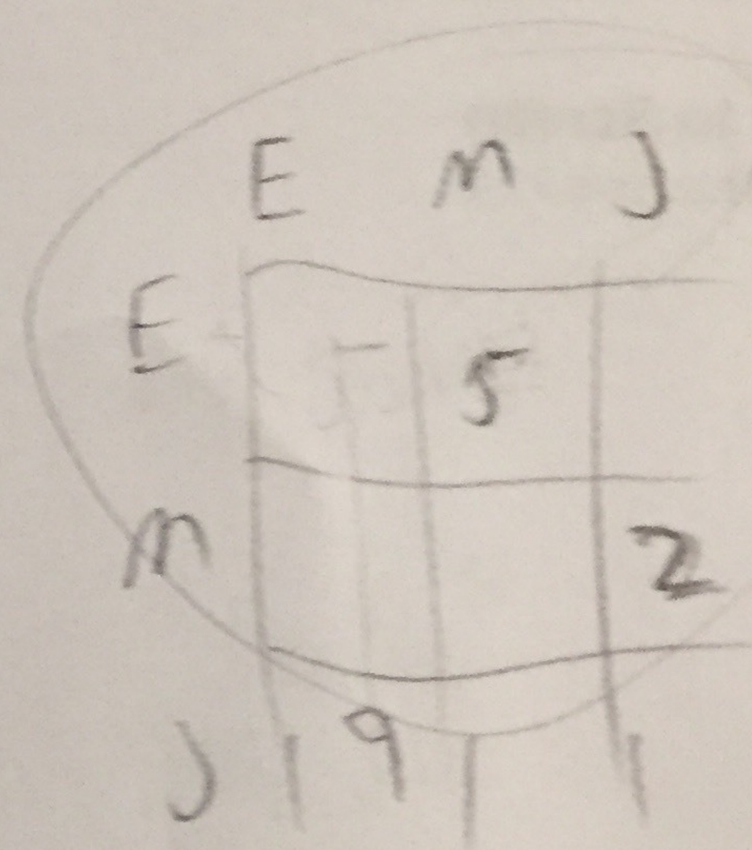
Output the list of cities from start to finish the format `fromplanet to toplanet` that takes the shortest path.

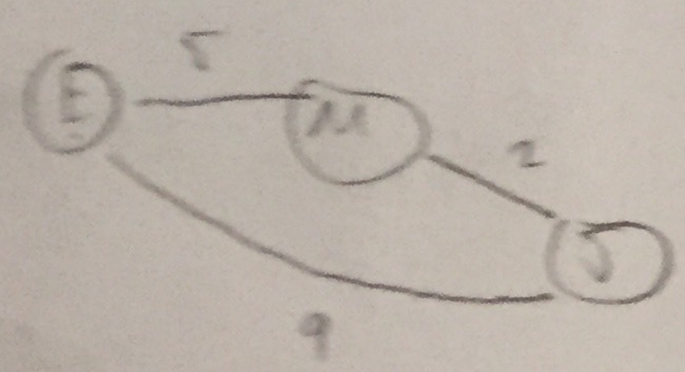
Example Input File

```
1
3
Earth Mars 5
Mars Jupiter 2
Jupiter Earth 9
Earth Jupiter
```

Example Output to Screen

```
Earth to Mars to Jupiter.
```





8. Dates

Program Name: Dates.java **Input File: dates.dat**

Not many people know this, but a lot of work goes into making the Avengers arrive on time for those awesome fights. They get a secretary or an intern to organize all this stuff, and in this case you! They have some reoccurring dates but don't feel like marking the calendar themselves.

Input

The first line will contain a single integer n that indicates the number of data sets that follow. Each data set will contain two lines. The first line will contain a date in `mm/dd/yyyy` format. The second will contain an integer n being the number of days before this date reoccurs.

Output

Output the new date exactly n days after the first in the format `mm/dd/yyyy`.

Example Input File

```
2
01/01/0001
1422
04/25/1959
24
```

Example Output to Screen

```
11/23/0004
05/19/1959
```

7. Escape

Tree

Program Name: Escape.java **Input File: escape.dat**

Some of the other Avengers are stuck in a maze, like the one that Tony Stark was stuck in. However, this is different because these guys can't blast their way out. They must see if they can make it out, and they ask you to know if they should call for help. Movement is up, down, left, and right only.

Input

The first line will contain a single integer `n` that indicates the number of data sets that follow. Each data set will contain two integers `r` and `c` denoting the row and column size of the maze. The next `r` lines contain `c` characters either `.` indicating a walkable space or `#` indicating a barrier. The next line contains four integers `x, y, z,` and `w` being the destination x and y and the beginning x and y values.

Note: $0 < r, c < 10{,}000$

Output

Output the word Escaped of they can make it out or Trapped if they need to call for help.

Example Input File

```
1
10 10
..######..
#....#....
##..##....
.#..##....
..#.......
........##
......##..
.......#..
.###.#.#..
.#..#.....
9 9 0 0
```

Example Output to Screen

```
Escaped
```

```
queue <coord>
visited [][]
map [][]

queue.add(sx, sy)

while (! queue.empty)
    coord = queue.poll()
    if (valid (cx-1, cy))
        queue.add (cx-1, cy)
    !
    !
```

```
bool valid(x, y)
return ! visited [y][x] and
    map[y][x] != '#'
x >= 0 and x < Visited[0].length

if (coord == end)
    escape = true
    break
visited [cy, cx] = true
```

6. Profit

Program Name: Profit.java **Input File: profit.dat**

As you may know Rocket loves to steal things. Now that he isn't saving the universe and has his ship back, he can go back to stealing! He also is very greedy with his money and doesn't want to hire an accountant, so he hires you to make a program to calculate his money for him. Though he may steal he doesn't steal from the galactic government, and he always pays his taxes. Write a program to help Rocket out.

Input

The first line will contain two integers `x` and `t`, denoting the number of items to come and the percent tax rate. The next `x` lines will contain a number indicating the value of an item Rocket stole and sold.

Output

Output the calculations in the format exactly as shown below.

Example Input File

```
2 10
1000
13249
```

Example Output to Screen

```
      Price *       Tax *    Profit *
************************************
    $1,000.00 *   $100.00 *   $900.00 *
   $13,249.00 * $1,324.90 * $11,924.10 *
************************************
   $14,249.00 * $1,424.90 * $12,824.10 *
```

print("%15.2f", 1000)

print(("%15.2f *",)

Trace

5. Blast

Program Name: Blast.java **Input File: blast.dat**

Iron man is trapped inside a building with a broken suit. His suit only has enough energy left to blast a few walls down. The building Tony is trapped in is a secret facility where they dropped him into a maze. In this maze he can only move in the four cardinal directions. Help him find a way out or print impossible if he cannot escape.

Input

Each data set will start with a 10 x 10 map of the facility Tony is trapped in, on the map there will be 5 characters. Each map will contain an `S` for where Iron man starts, an `E` for where he must exit, a period (.) which denotes a walkable space, a `*` which denotes a blastable wall, and an `X` which is an impenetrable wall. Walking one space, that is on a walkable space, takes 1 unit of energy, while blasting through a wall takes 5 units of energy. Given the amount of energy `n` in Mr. Starks suit determine if he can escape.

Output

Output the word "Escaped!" if Iron man escaped the facility or "Impossible" if he cannot escape.

Example Input File

```
XXXXXXXXXX
XX..*...XX
X..XXXX.XX
XXS.XXX.*X
XXX.XXXX.X
XX..*.XX.X
XX.XX.XX.X
XX*.XX..EX
XX.....XXX
XXXXXXXXXX
20
```

Coord
x, y
energy

Example Output to Screen

```
Escaped!
```

4. Sort

Program Name: Sort.java **Input File: sort.dat**

Thanos is trying to sort all the people left on the earth. As you all know, he has a thing for only doing half of something. He decides that it would be a good idea to sort the people on earth like this. He sorts half of the list of people by last name, first name, height in that order. Then he sorts half of the remaining unsorted list and so on.

Input

The data file will contain a list of people remaining on Earth. Each line will contain three elements `l`, `f`, and `h`. Each being last name, first name, and height.

Output

Output the sorted list of people according to the process described above. Output the names in the following order : first name, last name, height order.

Example Input File

```
Guinn Keith 5.1
Myles Dayjuan 5.8
Flavors Mr 8.9
Willis Preston 6.1
Lee Mercedes 5.3
Knight Bill 6.3
Phillips Willie 5.11
```

if (last.compareTo(other.last) != 0
return
else if

Example Output to Screen

```
Mr Flavors 8.9
Keith Guinn 5.1
Dayjuan Myles 5.8
Mercedes Lee 5.3
Preston Willis 6.1
Bill Knight 6.3
Willie Phillips 5.11
```

3. Lift

Program Name: Lift.java **Input File: lift.dat**

After the war with Thanos all the Avengers need to find new jobs. Luckily for Banner since he is now professor Hulk he can carry heavy objects. Banner is paid in items carried per trip, but even he can only carry a certain amount. However, he thinks he has found a loophole and uses his strength to break the items into pieces and only takes certain pieces. Since each item has a value, help Banner make the most money in a single trip as possible.

Input

The first line will contain a single integer n that indicates the number of data sets that follow. Each data set will start with a single integer x denoting how many items are in each set. The next two lines have x items each. The first line is the weights of each item, and the next line is the values of each item. The next line has a single integer denoting the maximum amount of weight Banner can carry.

Output

Output a single value indicating the maximum amount that Banner can make in one trip rounded to two decimal places.

Example Input File

```
1
5
400 140 90 230 300
60 20 15 20 130
1000
```

weights
values
max

~~so for max~~ $\frac{\text{value}}{\text{weight}}$

$\frac{\text{no}}{\text{200}}$ Item

Example Output to Screen

```
$231.09
```

```
if (weight + item.w > max)
    val += (max - weight)/item.w * item.value
    break
else
    val += item.value
    weight += item.w
```

© 2019 A+ Computer Science

1. Help

Program Name: Help.java **Input File: none**

The Avengers decided to take a little trick they saw from Batman. They really liked the bat signal but aren't as creative so they just flash help in the sky. They would like to have a program do the same to the screen.

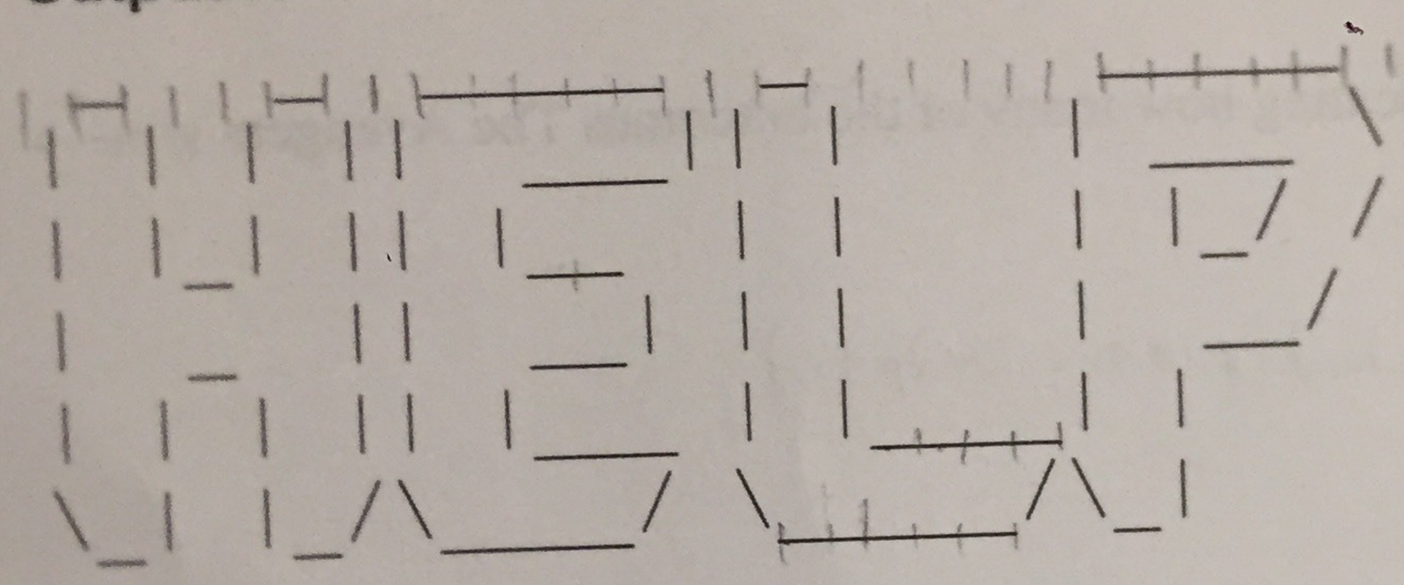
Input

None.

Output

The picture exactly as shown below.

Output to Screen



2. Chances

Program Name: Chances.java **Input File: chances.dat**

The Avengers know what percent chance they have of defeating Thanos, with the help of Dr. Strange. Now Strange is on another planet and they only remember their percent chance of winning and how many possible outcomes there are. Help them figure out how many of the outcomes they win.

Input

The first line will contain a single integer n that indicates the number of data sets that follow. Each data set will contain a decimal value p and an Integer value n representing the percent chance of winning and the number of possible outcomes. The number of outcomes can be very large.

Output

Output a rounded whole number indicating how many of the outcomes The Avengers win.

Example Input File

```
2
20 5
17.977 89
```

long outcomes = Math.round((p · 0.01 · (double) n))

Example Output to Screen

```
1
16
```