Problem 1: Capsule Capacity

A capsule is a gelatin container for enclosing medicine for delivery by ingestion.

The Center for Capsule Tolerance (CCT) recently standardized the capsule wall thickness at 0.5mm. It is now considering standards for several new overall capsule sizes. The researchers need to know the volume of capsule wall gelatin as well as the volume of the medicinal payload for variously sized capsules. Due to a new production process, all capsules being considered by CCT will be formed from perfect cylinders and perfect hemispheres (unlike the capsules shown at the right, which have various imperfections).

Remember: the volume of a sphere is $4 \pi r^3 / 3$.

Create a program to compute the capsule wall volumne and the medicial payload volume of a capsule, given its diameter and overall length.

The first line of input will be the number of capsules to be processed.

Each capsule is described by a line containing the diameter of the capsule in millimeters and its overall length in millimeters, separated by one space.

Output should be the capsule wall volume followed by the medicinal payload of the capsule, measured in cm^3, rounded to 7 decimal places as shown below.

Sample Input

```
3
10.5 20
19 22.5
13 19
```

Sample Output

```
0.3064362 1.1223013
0.6394450 3.9442696
0.3633776 1.5833627
```

Problem 2: Fenders

Once upon a time there was a rope weaver who made realistic boat fenders used to keep boats from scraping against each other, docks or piles. As a professional member of the Conscientious Committee on Tying (CCT), the rope weaver's fenders always require exactly the same length of rope for a particular fender design. Ropes are never spliced as it subtly changes the shape of the fender and CCT members are facidious about accuracy. All CCT fenders are made from authentic hemp rope available only on 500' spools.

Given the length of rope required for a fender, and the number of fenders ordered, determine how many spools will be required to satisfy the order.

The first line of input will contain the number of data sets to process.

Each data set consists of the number of fenders ordered, followed by a single space, followed by the length in feet required to weave one fender.

Each data set should produce one line of output indicating the number of spools required, as shown below.

Sample Input

```
4
15 100
3 200
3 166.5
5 167
```

Sample Output

```
3
2
1
3
```

Problem 3: Maze, Size

Garden mazes have been commissioned by the wealthy for hundreds of years. Guests find their way from one side of the maze to the other. It is known that mazes containing many "dead ends" (path locations which are bounded on three sides by a hedge) can feel quite long. The rule of thumb is to count a dead end as equivalent to five regular path segments.

The Committee on Circuitous Turnings (CCT) recommends placing benches to reduce the perceived size of the maze. When figuring the perceived size, CCT does not count any path space containing a bench. Additionally, CCT assumes that benches occurring at a dead end are so soothing that they reduce by the perceived size by five, rather than increasing it.

Given a description of a hedge maze, report the CCT perceived size.

The first line of input will contain the number of data sets to process.

Each data set consists of the number of rows and columns on a single line, followed by one line for each row of the maze, as shown below. Hedges are indicated by "**X**", path segments by "-", and path segments with benches by "+". The number of rows and columns for each maze will be integers greater than two and not more than 100. You can assume all maze passages will be exactly one space wide and, except for the entrance and exit, all edges are hedge filled.

Each data set should produce one line of output with the CCT perceived size.

Sample Input

```
3
5 5
XXXXX
X----
XX-XX
----X
XXXXX
5 5
XXX-X
X+--X
XXX-X
X--+X
X-XXX
8 7
XXXXXXX
X-----X
--X+X-X
XXXXX-X
X+X----
X-X-X-X
X+----X
XXXXXXX
```

Sample Output

```
17
2
10
```

Problem 4: Maze, Steps

Garden mazes have been commissioned by the wealthy for hundreds of years. Guests find their way from one side of the maze to the other. It is known that mazes containing many "dead ends" (path locations which are bounded on three sides by a hedge) can feel quite long, so the Committee on Circuitous Turnings (CCT) recommends that garden mazes contain only one path. CCT measures the size of a maze by the number of steps it takes to complete it.

Given a description of a hedge maze, report the CCT determined size of the maze.

The first line of input will contain the number of data sets to process.

Each data set consists of the number of rows and columns on a single line, followed by one line for each row of the maze, as shown below. Hedges are indicated by "**x**", path segments by "-", and path segments with benches by "+". The number of rows and columns for each maze will be integers greater than two and not more than 100. You can assume all maze passages will be exactly one space wide and, except for the entrance and exit, all edges are hedge filled.

Each data set should produce one line of output indicating the CCT determined size.

Sample Input

```
3
5 5
XXXXX
XX---
XX-XX
---XX
XXXXX
5 5
XXX-X
X+X-X
XXX-X
X--+X
X-XXX
8 7
XXXXXXX
X-----X
--XXX-X
XXXXX-X
X+X-X--
X-X-XXX
X+----X
XXXXXXX
```

Sample Output

```
7
7
11
```

Problem 5: Code

Voice synthesizing is all the rage in theme park and pizza place animatronics. Park and pizza patrons pen prose at a prompt to partake in this playful pastime. An incident in integration incorrectly incremented input ending in an interesting enigma at Creative Characters Telematic (CCT). Each key on the latest creature input keyboard enters the key to its left. For example, trying to enter "**`Hello, Joe.`**" actually sends "**`Jr;;p. Kpr/`**" to the voice synthesizer. To avoid costly delays, CCT has decided to use a software decoder to fix the hardware problem.

Write a program to convert each incorrect input to the correct text. CCT input keyboards only contain keys which have letters, numbers, period, comma, and semi-colon characters on them.

The first line of input will contain the number of lines to process.

Each line should produce one line of output containing the correct text.

Sample Input

```
3
Jr;;p. Kpr/
s nrr * drs
VVY Ti;rd@
```

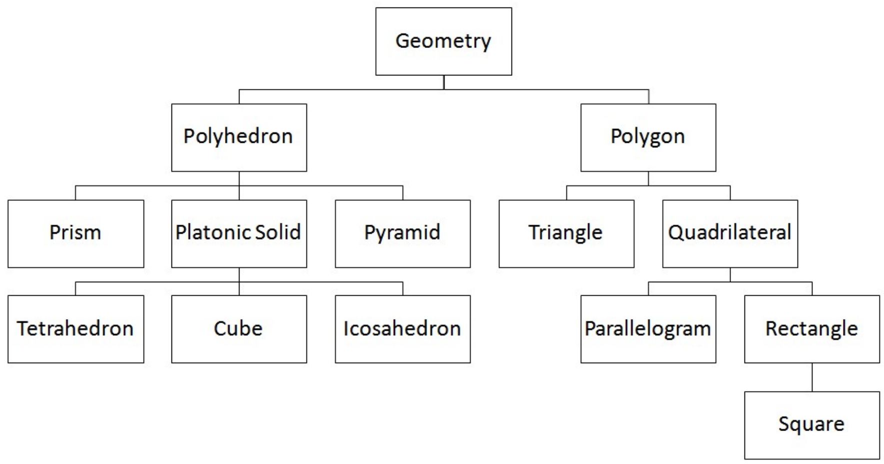
Sample Output

```
Hello, Joe.
a bee & sea
CCT Rules!
```

Problem 6: Parents

The chief architect at generiC aComplished daTabases (CCT) believes he has discovered a new way to store hierarchical data. He can't decide whether to call his discovery *Modified Pre-Order Traversal* or the *Nested Sets Method*. While explaining his idea he wrote the following table and chart on the whiteboard to show that descendants of any item can be found by looking for items with L values greater than, and R values less than, the item's own L and R values. For example, Quadrilateral (18,25) has descendants Parallelogram (19,20), Rectange (21,24), and Square (22,23); and no other items have values between 18 and 25.

L	R	Item
0	27	Geometry
1	14	Polyhedron
2	3	Prism
4	11	Platonic Solid
5	6	Tetrahedron
7	8	Cube
9	10	Icosahedron
12	13	Pyramid
15	26	Polygon
16	17	Triangle
18	25	Quadrilateral
19	20	Parallelogram
21	24	Rectangle
22	23	Square



Unfortunately, the CCT architect still needs to find a way to retrieve all the ancestors of an item. Write a program to print the ancestors of an item.

Problem 6: Parents (continued)

The first line of input will contain a single positive integer representing the number of items in the table. Each row of the table appears on a line by itself containing two integers, L and R respectively, and a name for the item; each separated by a single space. Names will not contain spaces.

Following the table will be a single positive integer representing the number of items to report on, each on a line by itself.

Each requested item should produce one line of output containing the item and its ancestors, each separated by a single space. Ancestors should appear in order beginning with the immediate predecessor.

Sample Input

```
14
0 27 Geometry
1 14 Polyhedron
2 3 Prism
4 11 PlatonicSolid
5 6 Tetrahedron
7 8 Cube
9 10 Icosahedron
12 13 Pyramid
15 26 Polygon
16 17 Triangle
18 25 Quadalateral
19 20 Parallelogram
21 24 Rectangle
22 23 Square
2
Square
Cube
```

Sample Output

```
Square Rectangle Quadalateral Polygon Geometry
Cube PlatonicSolid Polyhedron Geometry
```

Problem 7: Containers

In a classic puzzle, you have a three- and a five-gallon container, but wish to measure four gallons. The solution is to fill the five gallon container. Pour as much as will fit into the three gallon container (leaving two gallons remaining). Empty the three gallon container and pour the two gallons remaining in the five gallon container into the three gallon container. Refill the five gallon container and pour one gallon from the five gallon container by filling the three gallon container. The five gallon container now contains four gallons.

Different sequences of pouring and filling allow other amounts of liquid to be measured. The Consortium on Container Tabulation (CCT) advocates marking containers with a partial volume mark to allow additional measurements. For example, if the three gallon container had a one-gallon mark, four gallons could be easily measured by filling the five gallon container and pouring out one gallon into the three gallon container.

The CCT has adopted a standard notation to describe how to perform various measurements. Each container is represented by two digits, denoting respectively the number of gallons when filled to the mark and when filled to the brim. A three gallon container with a one-gallon mark is denoted as 13. Containers described as 05 and 55 are both containers without a mark.

A pouring is shown as S-D, where S is the source container code and D is the destination container code. Either can be blank. When pouring from one container to another, pouring stops when the destination unimark is reached, when the brim is reached, or the source container is empty, whichever comes first. A measurement is denoted as a sequence of pouring followed by a single container code indicating the container with the required amount. Each sequence assumes all containers begin empty.

Allow CCT to more easily test measuring schemes by creating a program to find the final amounts for each measuring.

The first line of input will contain the number of data sets to process. Each data set consists of a single line containing an CCT measuring and should produce one line of output containing the number of gallons contained in the indicated container at the end of the sequence.

Sample Input

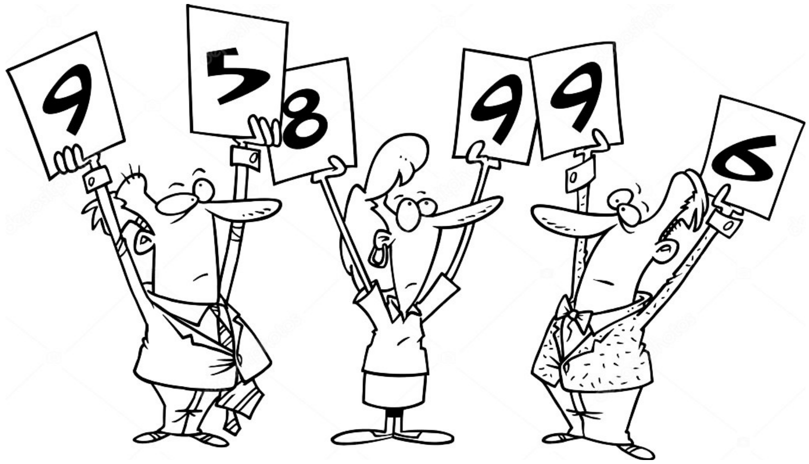
```
4
-05 05-33 33- 05-33 -05 05-33 05
-05 05-13 05
-05 05-13 05-13 05
-05 03
```

Sample Output

```
4
4
2
0
```

Problem 8: Scores

Various imfamous scandals involving ice ballet events have led the Committee on Civil Tallying (CCT) to develop a new system for scoring. Each of seven judges awards the performance a integer score between 0 and 10. The final score for the performance is the median.



Write a program to determine the final score for each ice ballet performance.

The first line of input contains the number of data sets to consider.

Each data set consists of seven numbers on a single line. Numbers will be separated by a single space.

Print the final score for each performance on a line by itself.

Sample Input

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

Sample Output

```
4
8
6
```