COMPUTER SCIENCES

George Fox University
H.S. Programming Contest
Division - II
2018

General Notes

1. Do the problems in any order you like. They do not have to be done in order
(hint: the easiest problem may not be the first problem)

2. Scoring: The team who solves the most problems in the least amount of time with the least submissions wins. Each wrong submission will receive a 20 min time penalty that will only be added to the time score once the problem has been successfully solved. Time is calculated for each problem as the total time from the start of the contest to the time it was solved.

3. There is no extraneous input. All input is exactly as specified in the problem. Integer inputs will not have leading zeros.

4. Your program should not print extraneous output. Do not welcome the user. Do not prompt for input. Follow the form exactly as given in the problem.
(hint: spaces? No spaces? What does spec say!)

5. All solutions must be a single source code file.

This page intentionally left blank

O. Quadratics PINK

After hearing that the quadratic formula is used constantly in calculus, you decide that it would be more efficient to just write a program for it! Write a program that solves the equation shown below:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

The $\pm$ sign represents two different equations, one with $+$ and one with $-$. Your answer will be both answers, separated by a comma.

Input

The first line will contain a single integer `n` that indicates the number of data sets that follow. Each data set will contain three decimals on a single line, representing `a`, `b`, and `c` respectively.

Output

Output the two answers to the equation, rounded to the thousandths digit and separated by a comma. The + sign will be the first answer, then the − sign will be the second. The solutions will always be real numbers.

Example Input

```
2
1 6 3
4 10 6.1
```

Example Output to Screen

```
-0.551, -5.449
-1.056, -1.444
```

This page intentionally left blank

P. 52 Cards YELLOW

Your friend Nathan is a bit of a wild card. You love to play competitive card games with him, but he always gets mad when he loses. When Nathan gets mad, he throws all the cards into the air. It's almost impossible to find all the cards on the first try. To see how many cards are missing, you must count the cards you found and subtract them from the total number of cards that are in a deck, 52. You're not going to stop playing cards with Nathan, so you decide to make a program to help you count how many cards are missing. Your program will count all the objects that you found and subtract the total number of cards from 52. For example, if you found 40 objects in the room (30 which were cards), there are 22 cards left to be found.

Input

The first line will contain a single integer `n` that indicates the number of data sets that follow. Each data set will be on its own line. Each data set will contain a single string, `s`. `s` will contain a mix of numbers (in the range of 0-9) and alphabetical letters (a-z and A-Z) representing all the objects in the room. Each number in `s` represents an object that is **NOT** a card. Each alphabetical letter in `s` represents an object that **IS** a card.

Output

On a unique line for each data set, output an integer that represents the total number of cards that still need to be found. The number of cards left to be found will never be negative.

Example Input

```
3
A7sdfh87h324h342hrsfsdahfsdf32
7sahdfhh723h4sdhrfsadf87sdafhd
a
```

Example Output to Screen

```
33
29
51
```

This page intentionally left blank

Q. Star Student GREEN

Your teacher needs help finding the student with the highest grade in her class. Being the only computer science student in the class, she asks you for help to make a program. Given all her students' grades, write a program to output the name of the student with the highest grade.

Input

The first line will contain a single integer `n` that indicates the number of data sets that follow. After the first line, there will be `n` data sets. Each data set will start with a single integer `x` denoting how many students are in each data set. The next `x` lines will contain a string and an integer that represent the student's name and grade.

Output

For each `n` data sets, print out the name of the student with the highest grade on a unique line. There will only be **ONE** student with the highest grade. Grades will be in the range of (0-100).

Example Input

```
2
3
Timmy 94
Jake 82
Luke 13
4
Alex 99
Duke 84
Nathan 100
William 78
```

Example Output to Screen

```
Timmy
Nathan
```

This page intentionally left blank

R. Coins ORANGE

You’re a wealthy human, and wealthy humans don’t count their own pocket change. Wealthy humans have their robot butlers count their pocket change. However, you are slightly frugal and bought a robot butler that still needs to be programmed. Luckily, you are a computer scientist. You job is to write a program for your robot butler. The program will determine if you have a certain amount of change given the coins in you pocket.

Input

The first line will contain a single integer `n` that indicates the number of data sets that follow. Each data set will consist of 2 lines of data. The first line of the data set has two integers. The first integer represents the desired value to be made in cents, and the second integer, `c`, represents the number of coin denominations there are. On the following line, there will be `c` integers representing the value, in cents, of the coins available. All coins can be used an infinite number of times.

Output

On a unique line for each dataset, output “`Possible`” or “`Not Possible`” based off whether the desired value can be made with the available coins.

Example Input

```
2
101 2
5 50
10 3
2 3 4
```

Example Output to Screen

```
Not Possible
Possible
```

This page intentionally left blank

S. Lost LIGHT BLUE

You and your friends are currently lost in a garden. Sadly, the garden is planted in the shape that forms a complex maze. The bushes are so high that the only way you and your friends can escape is to traverse the maze. However, you decide to send a robot through the maze to see how far you and your friends are from the exit. Write the program for the robot to use in order to calculate the distance to the exit.

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 two integers on a line, `r` and `c` respectively, indicating the size of the maze's `r` rows and `c` columns. On the following `r` lines, there will be `c` characters per line that represent the maze.

- A wall that cannot be passed through.
. – An empty space that can be passed through.
S – The location that the robot starts at.
E – The location of the exit.

Output

For each test case, output the distance to the exit from the starting location on a single line. If the maze is unsolvable, print "`Not Possible`" instead.

Example Input

```
2
5 5
#####
#S..#
#...#
#..E#
#####
7 7
#######
#S....#
#####.#
#.....#
#.#####
#....E#
#######
```

Example Output to Screen

```
4
16
```

This page intentionally left blank

T. DataMiner DARK BLUE

You work for the government as a data miner. Your job is to find the password hidden in a string of characters within a file. Being in computer science, you decide to write a program to do it for you automatically. The password is in a file that contains a mix of numbers (0-9) and uppercase and lowercase alphabetical letters (a-z). In this file, the password is formatted as such: “p:`password`:p” (where `password` is a random string of letters that represents the password). For example, an example file is “123abcp:henlo:pte31. In this case, “henlo” is the password. The password will always be formatted in the exact same way. It begins with a lowercase p followed by a colon, and it ends with a colon followed by a lowercase p.

Input

The first line will contain a single integer `n` that indicates the number of data sets that follow. Each data set will be on a new line and contain a single string. Each string will contain a series of uppercase and lowercase alphabetical letters (a-z and A-Z) and numbers (0-9). Inside the mix of letters and numbers there will be a password in the format of “`p:password:p`”. The password will be a random string of alphabetical letters that begins with “`p:`” and ends with “`:p`”. There will only be one password per data set. There will only be one “p:” and “:p” per data set.

Output

For each data set, output the password **without** the beginning “p:” and ending “:p” on a unique new line.

Example Input

```
2
dsfasdfsdfp:henlo:p12fdsabfsadf
p:HDSAsa:pasd123
```

Example Output to Screen

```
henlo
HDSAsa
```

This page intentionally left blank

U. Robbery RED

The newest video game just came out. However, you are short on money. You have decided to turn criminal and rob a bank with a squirt gun. Ignoring the reliability of your plan, you need a program that calculates the most value you can take from the bank given the amount of weight you can hold. Because you are only so strong (and can only carry so much weight), you need to find the optimal combination of items to take with you to get the most payoff.

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 two integers on a line, `W` and `O` respectively. `W` represents the maximum amount of weight that you can hold. `O` represents the number of items in the bank. On the next line, there will `O` integers that represent the value of each item. On the following line, there will be `O` integers that represent the weight of the object directly above it on the previous line. Objects are finite and can only be used once.

Output

For each data set, output the maximum value you can steal from the bank on a single line.

Example Input File

```
2
10 7
1 2 3 4 5 5 5
1 1 1 1 1 1 1
5 5
5 10 15 20 5
2 3 2 3 5
```

Example Output to Screen

```
25
35
```

This page intentionally left blank

V. Area LIGHT PURPLE

Math is not easy. Addition and subtraction are quite easy for you, but you struggle with multiplication. Because of this, you have decided to let the computer solve your math problems. You are to write a program that calculates the area of a rectangle given its length and width.

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 double values on a single line, `L` and `W`. `L` represents the length of the rectangle, and `W` represents the width of the rectangle.

Output

For each data set, output the area of the rectangle rounded to two decimal places.

Example Input

```
5
10.20 11.30
5.60 7.40
1.00 81.00
13.23 97.20
120.10 0.60
```

Example Output to Screen

```
115.26
41.44
81.00
1285.96
72.06
```

This page intentionally left blank

W. Combinations DARK PURPLE

You are in line waiting to pay for your school lunch. In your pocket, you have a series of coins with different values. Because the line is moving so slow, you decide to calculate all the different combinations of coins you can use to pay for your food. You decide to add a twist. Instead of just using the coins once, you decide that each of the coins can be used an infinite number of times. Your job is to write a program that will calculate all the different combinations a desired value can be made with the different coins in your pocket. Keep in mind, all the coins have different values, but they can be used an infinite number of times.

Input

The first line will contain a single integer `n` that indicates the number of data sets that follow. On the first line of each data set, there will be 2 integers `g` and `c`. `g` represents the desired value, in cents, to be made, and `c` represents the number of coins that will be listed on the next line. On the next line, there will be `c` integers representing the values, in cents, of the coins you can use. Each coin can be used an infinite number of times.

Output

For each data set, output the total number of combinations that can make the desired value.

Example Input

```
2
5 2
1 5
5 4
1 2 3 5
```

Example Output to Screen

```
2
6
```

This page intentionally left blank

X. Pottery BLACK

3D printing is becoming more and more prominent in today's society. Unfortunately, you don't currently have access to a 3D printer. Instead, you can just print them out with your knowledge of coding! Print out a clay pot.

Input

There is no input for this problem.

Output

Output the pot exactly as shown in the example output. There are no blank lines before or after the output, and there are no trailing spaces on any of the lines.

Example Input

There is no input for this problem.

Example Output to Screen

```
 _________
 \_     _/
   \   /
    | |
   /   \
  /     \
 |       |
/---------\
| \-/ \-/ |
\---------/
 \_______/
```

This page intentionally left blank

Y. Sorting Students WHITE

Your teacher asks you to write a program that will sort the students in her class based off their grades. Of course, your teacher wants the list to be sorted in descending order.

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 students are in the class. On the following `x` lines, there will be a single string and integer (in the range 0-105), separated by a space, representing the name and grade of the student respectively. Because your teacher's class is so diversely intelligent, all grade values will be unique.

Output

For each data set, output the name and grade of the students, with one student per line, sorted in descending order. After each data set, **EXCEPT** for the last, print a blank line.

Example Input

```
2
3
Nathan 91
Steve 101
Darrin 97
2
Alex 100
Ben 99
```

Example Output to Screen

```
Steve 101
Darrin 97
Nathan 91

Alex 100
Ben 99
```

This page intentionally left blank

Z. Draw SILVER

You’re bored during class one day. Instead of drawing shapes all over your paper, you decide to write a program that will do it for you! Write a program that, when given the type of shape and the dimensions, will draw the specified shape. The program will also be able to either leave the shape empty or fill it in. The shape names and examples are shown below:

`rectangle`	`##`
	`##`
		`###`
		`# #`
		`###`
			`###`
			`###`
			`###`

`left triangle`	`#`
	`##`
	`###`
		`#`
		`##`
		`# #`
		`####`
			`#`
			`##`
			`###`
			`####`

`right triangle`	`  #`
	` ##`
	`###`
		`   #`
		`  ##`
		` # #`
		`####`
			`   #`
			`  ##`
			` ###`
			`####`

`diamond`	` #`
	`# #`
	` #`
		`  #`
		` # #`
		`#   #`
		` # #`
		`  #`
			`  #`
			` ###`
			`#####`
			` ###`
			`  #`

The rectangle can be any number of rows and columns. The left triangle, the right triangle, and the diamond will always have the same number of rows and columns. For the diamond, the number of rows and columns will always be odd.

Input

The first line will contain a single integer `n` that indicates the number of data sets that follow. Each data set will be one line that starts with the shape name. If the shape is a rectangle, the name will be followed by two integers, `r` and `c`, representing the number of rows and columns respectively. If not, then the shape name will be followed by one integer, denoting the number of both rows and columns. The line will end with either `y` or `n`, `y` meaning that the shape is filled in and `n` meaning that the shape is empty.

Output

You will print the specified shape of the specified size, either shaded or unshaded as denoted by the letter at the end of the line. There are examples of the shapes in the table above. There are no lines of whitespace between data sets.

(Continued on the next page)

Z. Draw - Continued

Example Input

```
3
rectangle 3 5 n
right triangle 4 n
diamond 7 y
```

Example Output to Screen

```
#####
#   #
#####
   #
  ##
 # #
####
   #
  ###
 #####
#######
 #####
  ###
   #
```

This page intentionally left blank