Safe Range
Input File: SafeRangeIn.txt

Breanne just purchased a new car. She is planning a trip and would like to know the car's "safe" range. That is, beginning with a full tank of gas how far can she travel *and* still have one gallon of gas left in the car's tank. Write a program to perform this calculation for her using the EPA's estimate of Breanne's car's average miles per gallon, and the actual capacity of the car's fuel tank.

Inputs
There will be one line of input that contains two numbers. The first will be the EPA's estimate of the car's average miles per gallon, and the second will be the capacity of the car's fuel tank in gallons.

Outputs
There will be one line of output containing the two input numbers followed by the car's range. The output line should be annotated ***exactly*** as shown below. If a numeric output is an integer, you may output it with or without a trailing zero (e.g., 570 or 570.0).

Sample inputs
```
28.5 21.0
```

Sample output
```
MPG = 28.5; Fuel capacity = 21.0; gallons; Range = 570.0
```

Big Bucks
Input File: BigBucksIn.txt

Heather Thump, who lives in Manhattan, inherited fifty trillion dollars from her father, which she keeps in a Swiss bank account. Her friend Logan, who lives in Switzerland, is going to make a very large withdrawal for her. The amount of the withdrawal, which will always be an even dollar amount, will be sent to Logan in an email. He is concerned that if someone intercepted the email and discovered the amount of the withdrawal, he would be in jeopardy. So, Heather and Logan have agreed that the amount of the withdrawal will be the concatenation of the integers contained in the text of Heather's email.

For example, if the text of the email was:

> Today I ran 23 miles with my 3 dogs. As I ran I saw 20 blue birds in 5 trees.

The amount of the withdrawal would be $233,205.00 . Your task is to determine the amount of the withdrawal given the text of her email.

Inputs

The first line of input will contain the number of withdrawals to be made. This will be followed by one line of text per withdrawal. The email will never end with an integer.

Outputs

There should be one line of output per withdrawal that represents the amount of the withdrawal formatted as US currency with a leading dollar sign, a comma every three digits and two zeros trailing the decimal point: $x,xxx.00 .

Sample Inputs

```
3
Today I ran 23 miles with my 3 dogs. As I ran I saw 20 blue birds in 5 trees.
1 2 3 yes that's really me. 3 4 5 I'm glad to be alive.
My new address is 428 west 182 street, Brooklyn NY 10321 USA.
```

Sample Outputs

```
$233,205.00
$123,345.00
$42,818,210,321.00
```

SJC Penney
Input File: SJCPenneyIn.txt

SJC Penney, a chain of high end college apparel stores, is running a promotion to increase sales. For every purchase of $100 or more (before discount) a $10 discount will be applied to the purchase. The discount will be proportionally deducted from each item's cost based on the cost of the item in relation to the total non-discounted cost.

Inputs

There will be multiple lines of input. The first line will contain the number of items purchased, the remaining lines will contain the name and original cost of each item.

Outputs

The output will be in the format of a cash register receipt showing the name of each item and its discounted cost. The last line of the output will show the cost of the discounted items, which must always be exactly $10 less than the non-discounted total cost of the items purchased. And so, when the discounting process causes the discount to be less than $10, the discounted cost of the last item purchased will be adjusted to produce exactly a $10 discount. All numeric output will be displayed with two digits of precision, rounded up.

Sample inputs

```
5
SJC Sweatshirt      25.99
NYU Bag             30.00
RPI tee-shirt       10.99
MIT lounge pants    22.00
USC scarf           15.00
```

Sample Output

```
SJC Sweatshirt      23.49
NYU Bag             27.11
RPI tee-shirt        9.93
MIT lounge pants    19.88
USC scarf           13.57
TOTAL               93.98
```

Polly
Input File: PollyIn.txt

Maggie’s parrot Polly is learning to say sequences of letters. Maggie feels Polly has learned a sequence if she says it once, immediately repeats it, and then continues to repeat it until she is interrupted. For example, Maggie would say Polly learned the sequence “`ABC`” if Polly said “`ABCABCABCABC`” and was interrupted at the end of the third repetition, or said “`ABCABCABCA`″ and she was interrupted before the third repetition was complete. Sometimes it takes Polly a while to remember the sequence, and on those days she may say “`ABB`**`ABC`**`ABCABCA`” assuming she was again interrupted during the third repetition.

When Polly is having a bad day, she cannot learn a sequence. For example, on a bad day she may say: “`ABCABCABA`” or “`ABABCABA`” before she is interrupted. Your task is to examine what Polly says and determine if she has learned a letter sequence, or if she had a bad day.

Inputs
The first line of input will contain a single integer, which will be the number of days to consider. This will be followed with one line of input per day containing the character string, S, that Polly said that day before she was interrupted.

Outputs
There will be one line of output per day. If Polly learned a sequence that day, the line will contain a string followed by an integer. The string and the integer will be separated by a space. The string will be the character sequence Polly learned that day, and the integer will be the index (beginning from zero) in the input string S where the learning began. If she did not learn a sequence that day, the output will be the string: “Polly had a bad day”.

Sample inputs

```
7
ABCABCABCABC
ABCABCABCA
ABBABCABCABCA
ABCABCABA
ABABCABA
FF
ABCABCABCABCFF
```

Sample outputs

```
ABC 0
ABC 0
ABC 3
Polly had a bad day
Polly had a bad day
F 0
F 12
```

Jimmy the Geek
Input File: JimmyIn.txt

Jimmy the geek has discovered a fool proof algorithm for predicting the winner of a playoff basketball game. It uses the average number of regular season points per game each team's offence ***s***cores, S, and each team's defense has ***a***llows, A. For example, if St. Joe's offence scores an average of 80 points per game, and Duke's defense allows an average of 41 points per game, then when they play each other St. Joe will score 60 points = (80 + 41) / 2. To determine the winner of the game, Jimmy would perform a similar calculation to determine the number of points Duke will score, and the winner of the game would be the team with the highest score. There will never be a tie.

Write a program that uses Jimmy's algorithm to determine the winner of the college basketball playoffs, which is a single elimination tournament (one loss and you're out). In the first round, team 1 plays team 2, team 3 plays team 4, etc. In the second round, the winner of the team 1 vs. team 2 game plays the winner of the team 3 vs 4 game, etc.

Inputs

The first line of input will contain the number of college playoff seasons to consider. This will be followed by a group of input lines for each season. The first line in a group will contain the number of teams in the playoffs, N, which will always be a power of 2. This will be followed by one line of input per team that contains two integers, separated by a space. The first integer will be the team's S point value the second integer will be the team's A point value. The first of these lines will be team 1's statistics, the second team 2's statistics, etc.

Outputs

There should be two lines of output for each season. The first line will contain the team number of the team that won the playoffs. The second line will contain the margin of victory (winner's minus looser's points scored) for each game the champion team won, each separated by a space. These will be in round number order, with the last integer being the margin in the championship game.

Sample Inputs

```
2
4
100 76
74 52
88 68
85 93
8
100 98
130 121
88 79
88 74
88 82
91 81
```

Sample Inputs (continued)

```
82 80
89 78
```

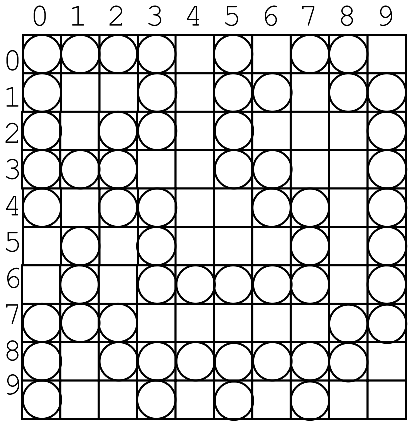
Sample outputs

```
1
1 2
4
2 2 2
```

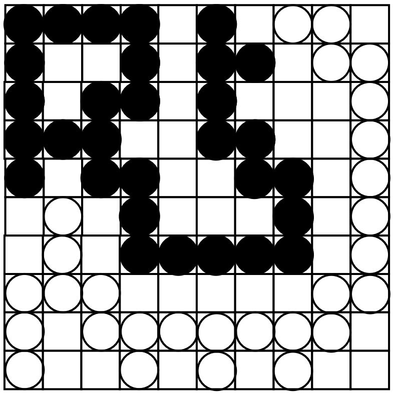
Froggy
Input Data File: FroggyIn.txt

Froggy, the frog, lives in a rectangular pond that has been divided into an orthogonal grid of 1 foot squares. Circular stepping stones have been placed at the center of many of the squares to allow Froggy to "stroll" around the pond. During his strolls, he can hop between adjacent stones: forward, backward, left or right, but *not* diagonally. He can revisit stones, and his objective during a stroll is to visit *every* stone he can reach without reentering the water.

For example, consider the stone arrangement in the pond shown below on the left. Assuming Froggy began his stool on the stone located in row 2 column 3, then the stones he hopped on during the strool are the black stones depicted in the pond shown below on the right.



The pond



The stroll beginning at row 2 column 3

You are to write a program to determine which stones are part of a stroll that begins at a given stone.

Inputs

The first line of the input contains the number of ponds to be considered. This is followed by one data set for each pond. The first line of a data set contains four positive integers, the number of rows, R, and columns, C, in the pond followed by the row and column number of the stone where the stroll begins. Each of the next R lines of a data set contains C marker characters, each of which corresponds to a square on the pond. The marker character **0** (zero) means that there is a stone on the square, and the marker character − (hyphen) means there is not. All inputs on a line are separated by a single space.

Outputs

For each pond, there should be R lines of output. Each output line will contain C marker characters, each separated by a single space. This grid of marker characters should be the same as the input pond description except that the **0** marker character of all of the stones on Froggy's stroll should be replaced with an **X** marker character. Separate marker characters with one space, and the output for each pond should be separated by one blank line.

Sample Input

Sample Output

```
2
10 10 2 3
0 0 0 0 - 0 - 0 0 -
0 - - 0 - 0 0 - 0 0
0 - 0 0 - 0 - - - 0
0 0 0 - - 0 0 - - 0
0 - 0 0 - - 0 0 - 0
- 0 - 0 - - - 0 - 0
- 0 - 0 0 0 0 0 - 0
0 0 0 - - - - - 0 0
0 - 0 0 0 0 0 0 0 -
0 - - 0 - 0 - 0 - -
9 10 6 0
0 - - 0 - 0 0 - 0 0
0 - 0 0 - 0 - - - 0
0 0 0 - - 0 0 - - 0
0 - 0 0 - - 0 0 - 0
- 0 - 0 - - 0 0 - 0
- 0 - 0 0 0 0 - - 0
0 0 0 - - - - - 0 0
0 - 0 0 0 0 0 0 0 -
0 - - 0 - 0 - 0 - -
```

```
X X X X - X - 0 0 -
X - - X - X X - 0 0
X - X X - X - - - 0
X X X - - X X - - 0
X - X X - - X X - 0
- 0 - X - - - X - 0
- 0 - X X X X X - 0
0 0 0 - - - - - 0 0
0 - 0 0 0 0 0 0 0 -
0 - - 0 - 0 - 0 - -

0 - - 0 - 0 0 - X X
0 - 0 0 - 0 - - - X
0 0 0 - - 0 0 - - X
0 - 0 0 - - 0 0 - X
- X - 0 - - 0 0 - X
- X - 0 0 0 0 - - X
X X X - - - - - X X
X - X X X X X X X -
X - - X - X - X - -
```

Lloyd of London
Input File: LloydIn.txt

Lloyd lives in London, and has a friend in each of the cities that British Airways serves. The airline has direct flights from London to some of the cities they serve. The other cities can be reached via a series of connecting British Airways flights originating from London. Every year, Lloyd will visit one of his friends. When making his flight plans, Lloyd always uses two criteria: 1- being British, he always flies British Airways and 2- being cheap, he always chooses the flights that result in the *minimum* air fare and he will never pay more than $10,000 to visit a friend. Your task is to determine the most inexpensive airfares between London and all the other cities British Airways flies to.

For example, if British Airways serviced four cities London, Paris, Rome and Madrid abbreviated as city number 1, 2, 3, and 4 respectively - then their flight and fare schedule could be expressed as:

			TO		
		1	**2**	**3**	**4**
F	**1**	0	100	0	200
r	**2**	150	0	50	0
o	**3**	100	0	0	30
m	**4**	0	75	0	0

This would indicate that there is a flight from London to both Paris and Madrid and the fares are $100 and $200 respectively, there is a flight from Paris to both London to and Rome and the fares are $150 and $50 respectively, there is a flight from Rome to both London to and Madrid and the fares are $100 and $30 respectively, and there is a flight from Madrid to Paris and the fare is $75. In this case the most *inexpensive* airfare from London to Madrid would be $180 (1 to 2, 2 to 3, 3 to 4).

Inputs

The first line of the input contains an integer, which is the number of years to consider. This will be followed by a group of lines for each year. The first line in a group will contain two integers separated by a space, which will be the number of cities Brtish Airways services, C, followed by the London airport number. This will be followed by one line of input per city, in city number order. Each line will contain C integers seperated by a space, which will be the fares from that city to all the cities, in city number order. If there is no service between two cities, the fare will be zero.

Outputs

There will be one line of output per year each containing C integers separated by a space. These integers will be the most inexpensive fares from London to all the other cities in city number order. If a city cannot be reached from London, output a fair of zero.

Sample Inputs and outputs

(see next page)

Sample Inputs

```
2
4 1
0 100 0 200
150 0 50 0
100 0 0 30
0 75 0 0
8 8
0 9 0 9 3 0 0 0
9 0 3 0 1 0 0 0
0 3 0 2 2 0 0 0
9 0 2 0 0 9 0 0
3 8 2 0 0 8 0 0
0 0 0 9 8 0 3 9
0 0 0 3 0 3 0 4
0 0 0 0 0 9 4 0
```

Sample Outputs

```
100 150 180
14 12 9 7 11 7 4
```