1. Beautiful Naturally

Program Name: Beautiful.java Input File: beautiful.dat

A group of Texas students have formed an organization called Redefining Beautiful. They believe that people should be recognized as beautiful for who they are naturally and not who they are with makeup. The girls at your school have decided to participate in the Redefining Beautiful campaign and to publicize the campaign by donating the money that they will save by not buying makeup this month to a local charity.

You have created a list of the girls from your school who are participating in the Redefining Beautiful campaign and how much money each of them normally spent per month on makeup. You are to write a program to print the total amount of money the girls will donate to charity.

Input

The first line of input will contain a single integer `n` that indicates the number of girls at your school participating in the Redefining Beautiful campaign. Each the following `n` lines will contain a student's first name and a space followed by the amount of money that person saved this month to the nearest penny.

Output

You will print the total amount of money donated to charity by the girls participating in the Redefining Beautiful campaign in dollars and cents, complete with the dollar sign as shown below.

Example Input File

```
4
Mary 13.99
Ruth 22.04
Anne 12.39
Taylor 18.34
```

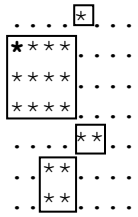
Example Output to Screen

```
$66.76
```

2. Blobs

Program Name: Blobs.java Input File: blobs.dat

Johnny is studying different shapes in a plane. For this particular study, he refers to the shapes as blobs even though they are solid rectangles. He represents his blobs in a rectangular grid as a collection of one or more contiguous asterisks (*). Contiguous means that the asterisks must be adjacent either horizontally or vertically. Non-blob characters are represented by periods (.). In the diagram below, there are 4 blobs.



You are to write a program that will determine the location of the uppermost, leftmost character of a blob given the coordinates of a given character in a grid. The uppermost, leftmost character of the largest blob (bolded *****) in the example above is row 2, column 1 or `2 1`. Rows and columns are numbered beginning with one.

Input

The first line of input will contain a single integer `n` that indicates the number of data sets to follow. For each data set:

- the first line will contain three integers in the form `r c s`
  - `r` ≥ 3 is the number of rows in the grid
  - `c` ≥ 3 is the number of columns in the grid
  - `s` is the number of test cases for that grid
- the next `r` lines will contain the grid
- the next `s` lines will each contain an ordered pair `x y`, 1 ≤ `x` ≤ `r` and 1 ≤ `y` ≤ `c` which is the location of a character in the grid

Output

For each ordered pair `x y`, you will print the coordinates in the form `j k` of the uppermost, leftmost character of the blob where 1 ≤ `j` ≤ `r` and 1 ≤ `k` ≤ `c`. If the test case falls on a square that is not part of a blob, print `NO BLOB`.

Example Input File

```
2
7 8 2
....*...
****....
****....
****....
....**..
..**....
..**....
2 1
5 3
5 6 3
...***
***...
...***
**.***
**.***
5 6
1 6
5 1
```

2. Blobs (cont.)

Example Output to Screen

```
2 1
NO BLOB
3 4
1 4
4 1
```

3. Diamonds are a Girl's Best Friend

Program Name: Diamonds.java Input File: diamonds.dat

Everyone knows that diamonds are a girl's best friend. You are to write a program that prints diamonds of different dimensions.

Input

The first line of input will contain a single integer `n` that indicates the number of diamonds that you are to print. Each the following `n` lines will contain an upper case letter of the alphabet, followed by a space and positive, odd integer `d`, $3 \leq d \leq 19$.

Output

For each line input, you will print a diamond of the letter input. The width and height of the diamond will be equal to `d`. Print a blank line after each diamond printed. The distance from the left edge of the screen is not important.

Example Input File

```
3
U 11
I 3
L 7
```

Example Output to Screen

```
      U
     U U
    U   U
   U     U
  U       U
 U         U
U           U
 U         U
  U       U
   U     U
    U   U
     U U
      U

 I
I I
 I

    L
   L L
  L   L
 L     L
L       L
 L     L
  L   L
   L L
    L
```

4. Elevator

Program Name: Elevator.java Input File: elevator.dat

Every elevator at the school has a sign that states that the maximum weight allowed on the elevator is 2000 pounds. You are to write a program that will print the number of people that will be allowed to ride on an available elevator at one time.

Input

The first line of input will contain a single integer `n` that indicates the number of available elevators. The next `n` lines will each contain a list of weights of the students waiting for an available elevator in the order that they will enter the elevator. The weights will be separated by a single space. Once an elevator is full, the remaining students in that line must use the stairs.

Output

For each elevator, you will print the number of students who can ride that elevator.

Example Input File

```
3
150 270 120 178 302 245 230 124 145 267 232 125 245 98 110 237
323 232 134 232 342 231 145 176 156 238 214 125
231 324 324 265 215 324 321 148 153 131
```

Example Output to Screen

```
9
9
6
```

5. Hickory Dickory Dock

Program Name: Hickory.java Input File: hickory.dat

Write a program that will read a data file and print every other line beginning with the first line.

Input
The input file will contain an unknown number of lines of text.

Output
You will output every other line of the data file beginning with the first line.

Example Input File
```
HICKORY DICKORY DOCK,
THE MOUSE RAN UP THE CLOCK.
THE CLOCK STRUCK ONE,
THE MOUSE RAN DOWN,
HICKORY DICKORY DOCK.
```

Example Output to Screen
```
HICKORY DICKORY DOCK,
THE CLOCK STRUCK ONE,
HICKORY DICKORY DOCK.
```

6. Life vs. School

Program Name: Life.java Input File: life.dat

James is having problems meeting his obligations with his girl friend. Sometimes, he makes a date with her only to find out later that he has a class that meets at the same time. He knows that, to keep his girl friend happy, he cannot miss his date with her. He also knows that, in order to graduate, he cannot afford to miss his classes. You are to write a program that will determine whether his class will conflict with his date or not. There will be a conflict if any part of his date will overlap any part of one of his classes. For example, a date that ends at 08:30 would conflict with a class that starts at 08:30.

Input

The first line of input will contain a single integer `n` that indicates the number of possible days that James may have conflicts in his schedule. For each day, there will be two lines of data. The first line of the two will contain two times that indicate the start and stop time of the date with his girlfriend. The second line of the two will contain one or more pairs of start and stop times of his classes. James has 1, 2 or 3 classes in a day. All times will be in the form of `hh:mm` of a 24 hour clock and will be separated by a space. Class times will be in order of when they meet.

Output

For each class on each day, you will determine if James has a conflict. For each class that he has a conflict, you will print `DAY #`*x* `CONFLICT WITH CLASS #`*y* where *x* is the number of the day in question and *y* is the number of the class that he would miss. If he does not have a conflict for a given class, you will print nothing.

Example Input File

```
3
08:45 11:15
06:45 08:00 12:00 03:00
10:30 13:30
08:30 10:30 13:45 14:45 16:45 18:46
14:00 17:00
10:45 11:45 15:00 16:00 16:45 18:15
```

Example Output to Screen

```
DAY #2 CONFLICT WITH CLASS #1
DAY #3 CONFLICT WITH CLASS #2
DAY #3 CONFLICT WITH CLASS #3
```

7. Miles Per Gallon Calculator

Program Name: MPGCalc.java Input File: mpgcalc.dat

Walter Johnson, an old, stingy man, has hired you to write some software to help him get the highest miles per gallon possible in his El Camino. He wants to attribute his driving style to his miles per gallon so that he can decide the best way to drive. Every time he fills up his tank, he will document his current odometer reading, the amount of gas he put in, and an arbitrary word for his own sorting purposes. Since he knows handling errors and making the software robust takes more time and thus more of his money, he has told you that you can assume his data is in the right format and the odometer readings are always increasing on each line.

Input

The file will contain a single line containing a positive integer that represents the odometer reading at the beginning of his study followed by an unknown number of lines. Each of the following lines will contain, in order, a positive integer responding to the odometer reading for the fill up, a space, a number rounded to tenths representing the number of gallons put into the car, a space, and a single word for categorization.

Output

For each category, print the list of category names in alphabetical order followed by that category's miles per gallon with tenths precision, left justified and aligned vertically as shown below.

Note: The space between columns is not significant as long as both columns are left justified and there is at least one space between columns.

Example Input File

```
11365
11415 2.7 HIGHWAY
11545 10.9 CITY
11650 9.3 CITY
11800 12.2 HIGHWAY_TRAFFIC
11843 2.9 HIGHWAY
```

Example Output to Screen

```
CITY               11.6
HIGHWAY            16.6
HIGHWAY_TRAFFIC    12.3
```

8. Palindromic Essays

Program Name: Palindrome.java Input File: palindrome.dat

Hannah Radar is a high school English teacher. However, she has a major idiosyncrasy. She loves palindromes, words that are spelled the same both forward and backward. When students write an essay, they are instructed to use as many palindromes in the essay as possible. When determining the student's grade on an essay, Madam Radar considers the number of palindromes that are two or more letters long that a student uses in that essay.

Since the essays are submitted electronically, Madam Radar wants you to write a program for her to find the palindromes in a student's essay. You are to write a program for Ms. Radar that will find all of the palindromes in an essay that are two or more than two letters long. She wants you to print an alphabetical list of the palindromes in the essay and the number of times each palindrome appears in the essay.

Input

The first line of input will contain a single integer `n` that indicates the number of essays to follow. On the following lines, `n` essays will appear with a single line containing only a dash (`-`) at the end of each essay. There will be no blank lines. All essays will be written in only upper case letters with correct punctuation. Periods and commas are the only punctuation marks included in the essay and will always be at the end of a word and followed by a single space.

Output

For each essay, you will print `ESSAY` *x*`:`, where *x* is the number of the essay, followed by an alphabetical list of palindromes found in the essay, one palindrome per line. If the same palindrome appears more than once, you will print a space followed by the number of times in parentheses that the palindrome appears. Print a blank line after the results of each essay.

Example Input File

```
2
BOB WAS FROM WASSAMASSAW, SOUTH CAROLINA. HE LOVED TO DRIVE HIS RACECAR. MARY
WAS FROM KINIKINIK, COLORADO, AND SHE LOVED TO KAYAK ON THE RIVERS.
THEY BOTH LOVED TO PLAY TIDDLYWINKS.
-
OTTO GOT A SUMMER JOB TO REPAPER THE DINING ROOM OF A HOUSE. THIS WAS HIS
FIRST SUMMER JOB EVER. IT WAS HARD FOR OTTO TO START THE WALLPAPER LEVEL TO
THE FLOOR BUT WITH PRACTICE, HE SUCCEEDED.
HE GAVE THE MONEY HE EARNED TO HIS MOM.
-
```

Example Output to Screen

```
ESSAY #1:
BOB
KAYAK
KINIKINIK
RACECAR
WASSAMASSAW

ESSAY #2:
LEVEL
MOM
OTTO (2)
REPAPER
```

9. Profit

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

Before computers, the price tag on items at local stores frequently had a set of letters as well as the price the owner wished to charge his customer for each item. The set of letters was a net cost code that represented the owner's net cost of the item.

You will be given the string of unique alphabetic characters that the owner used on the price tag to code his net cost in pennies, The code will contain 10 consecutive uppercase letters of the alphabet that will represent the code for the digits 0 through 9 in that order. For example, the code ABCDEFGHIJ would mean A=0, B=1, C=2, ... , J=9 and on the price tag, a code of DECJ would stand for $34.29. Additionally, you will be given the price of the item in dollars and cents, including the dollar ($) sign.

You are to write a program that will find the amount of profit the owner will make on a variety of purchases for different customers.

Input

The first line of input will contain a single integer `n` that indicates the number of customers to follow. For each customer, the first line will contain the ten-character code to be used for the net profit, a space, the customer's first name, a space, and a single integer `m` that indicates the number of items the customer bought. The next `m` lines will each contain the item's net cost code followed by a space and the price of the item.

Output

For each customer, you will print the customer's first name, a space, and the amount of profit the owner made on that customer's purchases. The profit should contain a dollar sign, no leading zeroes, and be correct to the nearest penny.

Example Input File

```
2
ABCDEFGHIJ Jerry 3
DECJ $39.00
ABC $2.01
CFIF $32.79
GUSOCPRTWD Tom 2
PTCSO $679.35
WUG $10.29
```

Example Output to Screen

```
Jerry $13.54
Tom $107.31
```

10. Shake

Program Name: Shake.java Input File: shake.dat

Kim and Pat like to text each other but are afraid that their parents might read their text messages. They have decided to write in code so their parents will not know what they are saying. They have developed a method for encoding their messages. The process for their code is:

- Put the message, including spaces, in the smallest square matrix that will hold the message. Fill cells in each row from left to right before moving to the next row. Rows are numbered beginning with row 1 as shown below.
- If there are unused cells at the end of the matrix, the first cell will be filled with an asterisk (*) and the remaining unused cells will be filled with consecutive letters of the alphabet beginning with the letter `A` and continuing until all empty cells have been filled.
- "Shake" the matrix to encode the message as follows:
  - odd numbered rows rotate each letter to the right one cell with the last cell becoming the first.
  - even numbered rows rotate each letter to the left one cell with the first cell becoming the last.
- Rewrite the coded message in the new order by rows.
- Send the coded message.

For example, the message:

`I love Computer Science` would fit into the 5x5 matrix at the left below and after the "Shake" the matrix would look like the matrix on the right below.

1	I		L	O	V
2	E		C	O	M
3	P	U	T	E	R
4		S	C	I	E
5	N	C	E	*	A

1	V	I		L	O
2		C	O	M	E
3	R	P	U	T	E
4	S	C	I	E	
5	A	N	C	E	*

The encoded message sent would be: `VI LO COMERPUTESCIE ANCE*`

You are to write a program that will decode the encoded message.

Input
The first line of input will contain a single integer `n` that indicates the number of encoded messages to be sent. Each the following `n` lines will contain a single encoded message less than 625 characters long.

Output
For each encoded message, you will print the decoded message. Do not include the characters used to fill the matrix.

Example Input File
```
2
VI LO COMERPUTESCIE ANCE*
 WINNINGIL DISTU RICT ISHE FIRSTTT STEP WARD STODATE*ABCLEFGHIJK
```

Example Output to Screen
```
I LOVE COMPUTER SCIENCE
WINNING UIL DISTRICT IS THE FIRST STEP TOWARD STATE
```

11. Shopping to the Max

Program Name: Shopping.java Input File: shopping.dat

Richard's mother has given him $200 to spend for clothes for college. The problem is that Richard must spend it all at one store and cannot spend more than $200. Richard has gone to several stores and found the prices, which include any taxes, of many items that he wants to buy.

You will write a program that will find which items he should buy at each store given that he wants to spend as much as possible without going over the $200 maximum.

Input

The first line of input will contain a single integer `n` that indicates the number of stores that Richard has researched. Each the following `n` lines will contain a list of the prices of the items that Richard selected from that store. All prices will be in the form `ddd.cc`, where `d` represents dollars with no leading zeros and `c` represents cents. The prices will be separated by a single space.

Output

For each store, you will print the smallest amount of money that he would have left from the $200 he was given if he bought the items that would total the most without going over his $200 limit. The output should be in the form `ddd.cc`, with no leading zeros to the left of the decimal. For example, 19 cents would be output as .19.

Example Input File

```
3
14.99 79.99 80.29 13.44 15.17 33.29
88.10 15.00 111.90 45.14 2.25 33.45
20.00 50.00 10.00 20.00 30.00 40.00 70.01 30.10 25.25
```

Example Output to Screen

```
6.43
.00
4.64
```

12. Students' Grades

Program Name: Students.java Input File: students.dat

Ms. Appleworth is a teacher and needs you to write a program that will create a bar graph of her students' grades.

Input
The first line of input will contain a single integer `n` that indicates the number of students in her class. The following `n` lines will each contain a student's name and a space followed by the student's grade in the range [0...100].

Output
The output will consist of 7 lines with the labels as shown below and a space followed by one asterisk (*) for each student's grade that falls in the given interval. The colons should be in a vertical line and the grade intervals before the colons right justified to the left of the colons.

Note: The number of spaces to the left of the label does not matter as long as the colons are aligned.

Example Input File
```
13
GEORGE 67
AMOS 54
ROBERT 87
RICHARD 98
MARY 45
ANN 98
LUCY 20
WENDY 64
SHARLA 62
JOHN 59
JOE DON 98
MARY LOU 90
LARRY 92
```

Example Output to Screen
```
   <40: *
 40-49: *
 50-59: **
 60-69: ***
 70-79:
 80-89: *
90-100: *****
```