

New Zealand Programming Contest 2019

PROBLEM SET

PREAMBLE

Please note the following very important details relating to input and output:

- Read all input from the keyboard, i.e. use `stdin`, `System.in`, `cin` or equivalent. Input will be redirected from a file to form the input to your submission.
- Do NOT prompt for input as this will appear in your output and cause a submission to be judged as wrong.
- Write all output to the screen, i.e. use `stdout`, `System.out`, `cout` or equivalent. Do not write to `stderr`. Do NOT use, or even include, any module that allows direct manipulation of the screen, such as `conio`, `Crt` or anything similar.
- Output from your program is redirected to a file for later checking. Use of direct I/O means that such output is not redirected and hence cannot be checked. This could mean that a correct program is rejected! You have been warned.
- Unless otherwise stated, all *integers* will fit into a standard 32-bit computer word. If more than one integer appears on a line, they will be separated by spaces.
- An *uppercase letter* is a character in the sequence 'A' to 'Z'. A *lower case letter* is a character in the sequence 'a' to 'z'. A *letter* is either a lower case letter or an upper case letter.
- Unless otherwise stated, a *word* or a *name* is a continuous sequence of letters.
- Unless otherwise stated, a *string* is a continuous sequence of visible characters.
- Unless otherwise stated, words and names will contain no more than 60 characters, and strings will contain no more than 250 characters.
- If it is stated that 'a line contains no more than n characters', this does not include the character(s) specifying the end of line.
- Input files are often terminated by a 'sentinel' line, followed by an end of file marker. This line should not be processed.

PROBLEM A**CAR PARKING****3 POINTS**

The manager of a car park has asked for your help with an occupancy survey. He wants to know how many spaces in the car park are occupied on consecutive days.

Input

The first line in the input will be a single positive integer, N ($0 < N \leq 250$), which is the number of parking spaces available.

The second line contains data for the first day. It consists of N characters, O for an occupied space, with $-$ for an empty space.

The third line contains similar data for the second day.

Output

Output consists of a single integer, the number of parking spaces that were occupied on both days.

Sample Input 1

```
5
00-00
-00--
```

Output for Sample Input 1

```
1
```

Explanation

Only parking space 2 (numbering from the left) was occupied on both days.

Sample Input 2

```
7
0000000
00-0--0
```

Output for Sample Input 2

```
4
```

Explanation

Parking spaces 1, 2, 4 and 7 were occupied on both days.

PROBLEM B

ARITHMETIC

3 POINTS

How is your arithmetic? In this problem, you will be presented with 2 numbers and have to calculate and display their sum, difference, product, quotient and remainder.

Input

Input will consist of 2 numbers, N1 and N2, each on a line of its own. Both numbers will be integers between -100 and 100 inclusive. N2 will not be 0.

Output

Output 4 lines showing the required values. The required output format is:

N1 + N2 = <sum>

N1 - N2 = <difference>

N1 x N2 = <product>

N1 divided by N2 is <quotient> remainder <remainder>

where <sum> is $N1 + N2$, <difference> is $N1 - N2$, <product> is $N1 \times N2$, <quotient> is N1 divided by N2 (integer part only) and <remainder> is the remainder when N1 is divided by N2.

Turn over for sample input and output

Sample Input 1

27
14

Output for Sample Input 1

$27 + 14 = 41$
 $27 - 14 = 13$
 $27 \times 14 = 378$
27 divided by 14 is 1 remainder 13

Sample Input 2

-8
11

Output for Sample Input 2

$-8 + 11 = 3$
 $-8 - 11 = -19$
 $-8 \times 11 = -88$
-8 divided by 11 is 0 remainder -8

Sample Input 3

15
-7

Output for Sample Input 3

$15 + -7 = 8$
 $15 - -7 = 22$
 $15 \times -7 = -105$
15 divided by -7 is -2 remainder 1

PROBLEM C**COOKING****3 POINTS**

Most people use a recipe when cooking something for the first time. They also check that they have all the ingredients they need before they start! This problem will require you to help with that process.

Input

The first line of input will contain the name of the item being made. It will be no more than 20 characters which may include spaces.

The next line will contain a single positive integer, N , the number of ingredients required ($N \leq 20$).

The next N lines will each contain details of 1 ingredient. The format for this will be

`<name>`, `<units>`, `<amount>`

`<name>` will be the name of the ingredient with up to 20 lower case letters or spaces

`<units>` will be the units in which the ingredient is measured. Solid ingredients will be **g** (grams) or **kg** (kilograms, 1000 grams), powders or liquids in **ml** (millilitres) or **l** (litres, 1000 millilitres) while **x** will denote a simple count.

`<amount>` will be the number of the units required. The number will be positive and may contain a decimal part.

The final N lines will contain the amount of each ingredient available. The format will be the same as that described above for the ingredients, but they may not be listed in the same order. The `<amount>` value may be zero. The units will not necessarily be the same.

Output

If an appropriate amount of each of the ingredients required is available, the output should be

You may make your `<name>`.

where `<name>` is the name of the item from the input.

If there is not enough of one or more ingredients, the output should be

You may NOT make your `<name>`.

You need `<ingredient list>`.

where `<name>` is the name of the item from the input and

`<ingredient list>` contains the names of each ingredient for which not enough is available. Ingredients are to be listed in alphabetical order with a comma and space between each.

Turn over for sample input and output

Sample Input 1

Basic butter cake
4
butter, g, 200
caster sugar, g, 200
eggs, x, 4
self raising flour, g, 200
eggs, x, 10
butter, g, 500
self raising flour, kg, 1.5
caster sugar, g, 450

Sample Input 2

Honey-glazed ham
6
ham, kg, 2
honey, ml, 150
orange juice, ml, 75
soy sauce, ml, 18
allspice, ml, 6
ground cloves, ml, 3
orange juice, ml, 0
allspice, ml, 17
ham, kg, 2
honey, ml, 250
ground cloves, ml, 8
soy sauce, ml, 25

Output for Sample Input 1

You may make your Basic butter cake.

Note that 1.5 kg of flour is 1500 g so more than enough to make the cake.

Output for Sample Input 2

You may NOT make your Honey-glazed ham.
You need orange juice.

PROBLEM D

CROQUET

3 POINTS

Association Croquet is a game played between 2 teams either as singles or doubles. Each team has 2 balls which they hit with their mallets. During a turn, the same ball must be hit each time.



The aim is to score points by manoeuvring both balls on your side through 6 hoops, once in each direction, then hitting the centre peg. A point is scored each time a ball is hit through the next hoop, and when the ball hits the centre peg. 26 points wins the game.

A number of strokes are possible; they are represented by the following codes.

- M Miss – a ball does not hit another ball, does not hit the centre peg, nor does it run a hoop. This ends the player’s turn. Play continues with a player from the opposing team.
- R A roquet. A ball hits another ball; this gives the player another hit which must be a croquet shot.
- C A croquet shot. The player’s ball is placed in contact with the ball it roqueted and played so that both balls move. The player then has a free hit.
- H A hoop shot. The ball is hit through the next hoop in the correct direction. The player scores a point and then has a free hit.
- P A ball is pegged out by hitting it against the centre peg. This scores a point and ends the player’s turn. The pegged out ball is removed from the game, and play continues with a player from the opposing team.

This problem requires you to follow a series of strokes in a game and track the score.

Input

The first 2 lines contain the names of the two teams who are playing the game recorded, one per line. Names may contain spaces or punctuation but will not be longer than 20 characters.

The next line contains the score at the start of the series of recorded strokes. It will consist of 2 integers, both in the range 0 to 25 inclusive, separated by a space. The first score will be for the first named team.

The fourth line will be a positive integer, N, being the number of characters in the final line ($0 < N \leq 255$).

The final line will consist of a series of N characters, each being one of the letters M, R, C, H and P. These are codes for strokes as defined above. The first stroke is played

by the first named team. The sequence of strokes will not necessarily complete a game, but will not take a team's score beyond 26 points.

Output

The score at the end of the sequence of strokes recorded in the format

team_1 x team_2 y

team_1 and team_2 are the names of the 2 teams in the order they were input.

x is the score of team_1 and y is the score of team_2

Sample Input

Sam and Tom

Trish & Kathy

3 12

27

RCRCHRCHRCHRCMMRCRCHRCHRCRCH

Output for Sample Input

Sam and Tom 8 Trish & Kathy 12

Explanation

RCRCHRCHRCHRCM This is the first turn where 3 hoops are run by Sam and Tom.

M This is the second turn where no hoops are run by Trish & Kathy

RCRCHRCHRCRCH This is the third turn where 2 hoops are run by Sam and Tom.
The turn is not finished but no more was recorded.

New Zealand Programming Contest 2019

PROBLEM E

NETBALL COACHING

10 POINTS

Netball has had a high profile recently in New Zealand. The men's team beat the women's team (the Silver Ferns) in a recent series. The Silver Ferns then went on to win the World Championship. This has led to an upsurge in the number of people who want to take part in netball coaching classes.

Alison Cooke runs a highly rated course at various centres across New Zealand. To pass the course, a student must participate in 5 exercises for which they will be graded.

Performance in each exercise is given a pass/fail grade, and must be passed to score a mark above 0. A mark of 10 is awarded if an exercise is passed first time, 7 if passed second time and 4 if passed after 3 or more attempts.

Exercises are weighted according to how difficult Ms Cooke judges them to be. This will vary from course to course, but each exercise will be weighted between 1 and 3. The student's mark is multiplied by the weighting of the exercise to give their weighted mark for each exercise. Exercise marks are added to give each student a total weighted mark for the assessment.

To pass a student's total weighted mark must be at least 50% of the total weighted marks available for the 5 netball exercises.

Input

The first line of input will contain 5 numbers, separated by spaces, which are the weightings for each of the 5 exercises. Each weighting will be an integer between 1 and 3 inclusive.

The next line will contain a single integer, S ($0 < S \leq 25$), the number of students to assess. There then follow S lines, each defining one student. A line will contain no more than 32 characters. The line contains a 4 digit ID, the student's first name followed by the student's last name, all separated by spaces.

There then follows a series of lines representing the results for students in the exercises. Each line has the following format:

```
<student id> <exercise> <result>
```

where <student id> is the 4 digit ID of the student whose result is shown,

<exercise> is one of the letters A to E to identify the exercise,

<result> is one of the letters P or F to indicate Pass or Fail

A student who fails an exercise may take it again until it is passed.

The final line will be

```
0 # #
```

This line marks the end of input and should not be processed.

Output

Output consists of one line for each student from the input, in the same order as they were input. Each line will contain the name of the student followed by a space, then the word Pass or the word Fail.

Sample Input	Output for Sample Input	Explanation
1 1 2 3 3 4 1206 Angela Williamson 1244 Barry Read 3564 Josephine Henry 2272 Thomas Hansen 3564 A P 1206 C F 1244 B F 3564 C F 1244 B F 1244 D F 2272 A F 1206 B F 2272 B F 1206 E P 2272 A P 1244 A P 1206 D P 3564 D P 1244 C P 2272 C F 1244 D P 3564 C F 2272 C P 3564 B P 3564 C P 1244 B F 2272 E F 2272 B P 1206 A F 1206 B P 2272 E P 0 # #	Angela Williamson Pass Barry Read Pass Josephine Henry Pass Thomas Hansen Fail	Total possible mark is $10+10+20+30+30$ ie 100 Angela Exercise A Failed 0 Exercise B Second time $1 \times 7 = 7$ Exercise C Failed 0 Exercise D First time $3 \times 10 = 30$ Exercise E First time $3 \times 10 = 30$ Total 67 Pass Barry Exercise A First time $1 \times 10 = 10$ Exercise B Failed 0 Exercise C First time $2 \times 10 = 20$ Exercise D Second time $3 \times 7 = 21$ Exercise E Not taken 0 Total 51 Pass Josephine Exercise A First time $1 \times 10 = 10$ Exercise B First time $1 \times 10 = 10$ Exercise C Third time $2 \times 4 = 8$ Exercise D First time $3 \times 10 = 30$ Exercise E Not taken 0 Total 58 Pass Thomas Exercise A Second time $1 \times 7 = 7$ Exercise B Second time $1 \times 7 = 7$ Exercise C Second time $2 \times 7 = 14$ Exercise D Not taken 0 Exercise E Second time $3 \times 7 = 21$ Total 49 Fail

New Zealand Programming Contest 2019

PROBLEM F




SET GAME

10 POINTS

Set is a card game played with a pack of 81 unique cards. Each card in the pack has 4 properties, having one of the three possible values for each property.

Colour – Red (R), Green (G), Blue (B)	Number 1, 2, 3,
Shape – Diamond (D), Oval (O), Squiggle (S)	Fill – Filled (F), Shaded (S), Empty (E)

This sample contains 3 cards and illustrates all the variations. If it is not in colour, note that the left card is red, the middle card is green and the right card is blue.

		
RD1F	GO2S	BS3E

Below each card is its representation, being the value for its colour, shape, number and fill in that order. In this problem, cards will always be represented in this manner.

The 3 cards illustrated form a set. A set is a collection of 3 cards such that for each property all 3 cards are either the same or different. The 3 cards above are different for all 4 properties.

Because each card is unique, at least one property must be different.

Here are some other examples.

RD1F	RD1S	RD1E	SET. Same colour, shape and number, different fill.
RD1F	RD1S	RD2E	NOT a set. Two are 1s but one is not.

It is also true that given any two cards, only 1 other card will complete a set with them. In this problem you will be given 2 cards and have to identify the 3rd card required to make a set.

Input

Input begins with a single integer, N, on a line by itself (0 < N <= 30). There then follow N lines, each line representing a group of 2 cards. Each card will be represented by 4 characters as described above. The cards will each be separated by a space.

Output

For each line of input, produce one line of output containing the 4 characters, in the specified order, that identify the other card required to form a set.

Sample Input

4
RD1F RD1S
RD1F RD2E
RS1F G02F
BS3E BD1E

Output for Sample Input

RD1E
RD3S
BD3F
BO2E

PROBLEM G

POSTFIX OPERATIONS

10 POINTS

To correctly evaluate an expression like $3 + 4 * 2$, you need to know the rules of arithmetic. You do the multiplication first, giving 8, then add the 3 to get 11. If you add first to get 7 then multiply that by 2 you get 14, an incorrect answer.

An alternative way to write this expression is $3 4 2 * +$. Here the operators come after the operands (the numbers), hence postfix. To evaluate this, when you encounter the $*$, multiply together the previous two operands (so $2 * 4$) and store the answer. You next encounter the $+$ so add the stored answer, 8, to the 3. This is the same as the answer in the first paragraph.

In this problem you will be required to evaluate expressions which use postfix notation.

Input

The first line of input will be a single integer, N ($0 < N \leq 20$), the number of expressions to follow. The remaining N lines will each contain a valid expression for you to evaluate.

Each expression will contain a list of non-negative integers less than 100 and arithmetic operators ($+ - * /$) with no more than 30 characters. Every operator will have two available operands. $/$ is integer divide (returns only the integer part of the quotient). Division by zero will not occur.

Output

For each expression in the input, output the evaluation of the expression, a single integer.

Sample Input	Output for Sample Input	Explanation
3		
3 4 2 * +	11	$2 * 4 + 3$
2 9 5 + * 3 -	25	$(9 + 5) * 2 - 3$
17 10 + 3 * 9 /	9	$(17 + 10) * 3 / 9$

PROBLEM H

ARRIVAL

10 POINTS

If you have ever travelled internationally by plane, you will be aware that working out when you are going to arrive at your destination depends on the flight time and the time zone difference between departure and destination airports.

In this problem you will be given the departure city, its time zone and the departure time, the destination city and its time zone, and the flight time. From this data you must calculate the arrival time.

Input

Input will be for a single journey in the following format:

Line 1: The departure city

Line 2: The time of departure as `mm dd hh:mm` (month, day, hours and minutes all as 2 digits and separated by spaces and a colon as shown).

Line 3: The time zone of the departure city as (UTC) + or -hh:mm

Line 4: The destination city

Line 5: The time zone of the destination city as (UTC) + or -hh:mm

Line 6: The flight time as `hh:mm` (hours and minutes both as 2 digits and separated by a colon)

A time zone which is exactly UTC will be shown as `00:00`. Flights have been selected so that:

- No flight crosses the International Date Line.
- Flights arrive on the day of departure or the following day.

Output

Two lines giving the city and time of departure and the city and time of arrival. The lines will start with the words `Departs` and `Arrives` respectively.

Departure time to be shown as `mm dd hh:mm`, as in the input.

Arrival time to be shown as `hh:mm` (hours and minutes, both as 2 digits and separated by a colon) followed by either `same day` or `following day` as appropriate.

Turn over for Sample Input and Output

Sample Input

Dubai
09 11 10:05
+4:00
Auckland
+12:00
15:45

Output for Sample Input

Departs Dubai 09 11 10:05
Arrives Auckland 09:50 following day

Explanation

The flight took 15 hours and 45 minutes and there was an 8 hour time difference. It arrived in Auckland the following day at 9:50 am.