Brainrot

You may know about the “brainrot” terms of this generation. Words like “skibidi” and “sigma” leave older generations confused! Your task is to write a program that outputs the **second part of the brainrot phrase when given the first word.**

Your program should do the following: - If given the word `skibidi`, print `toilet`. - If given the word `fanum`, print `tax`. - If given the word `sigma`, print `male`.

The input to your program will only be one of these three words (`skibidi`, `fanum`, or `sigma`)

Input

The first line of the input is the number of words. The next lines each contain a word.

Output

For each word, output the corresponding second word on a new line

Sample Input

```
3
skibidi
fanum
sigma
```

Sample Output

```
toilet
tax
male
```

Music Scales

In music, the C major scale has 8 notes: `C`, `D`, `E`, `F`, `G`, `A`, `B`, and `C` again. Each note has a position in the scale:

```
1 → C
2 → D
3 → E
4 → F
5 → G
6 → A
7 → B
```

Given a number from 1 to 8, determine the note in the C Major scale at that position.

Input

A single number (1 to 8).

Output

The note (capitalized) corresponding to that number in the C Major scale.

Sample Input

```
4
```

Sample Output

```
F
```

Cooking

Gordon Ramsey is preparing a new recipe, but the ingredient measurements are in cups, teaspoons, and tablespoons, and he wants them in ounces! Write a program to help Gordon Ramsey convert from cups, teaspoons, and tablespoons into ounces so he can follow the recipe.

Conversions

1 cup = 8 ounces

1 tablespoon = 0.5 ounces

1 teaspoon = 0.1667 ounces

Input

The first line is the number of ingredients to convert.

Each of the next lines contain a number representing the quantity, followed by a space, followed by a string representing the unit ("cup", "tablespoon", or "teaspoon")

Output

For each ingredient, output the equivalent quantity in ounces, rounded to 2 decimal places. The provided sample code handles rounding the number for you.

Sample Input

```
3
1 cup
3 tablespoon
12 teaspoon
```

Sample Output

```
8.00
1.50
2.00
```

Photo Cleanup

You love photography and take lots of pictures every day. Unfortunately, your computer is out of storage space! To free up space, you need to delete some photos.

You are given the size of each of your computer's photos (in megabytes) and your computer's storage limit. Write a program to calculate the minimum number of photos you need to delete to make the total size of the remaining photos less than or equal to the storage limit.

Input

The first line contains two numbers. The first number is the number of photos (up to 10) and the second number is the storage space you have (up to 1,000).

The second line contains the number of megabytes for each photo, separated by spaces.

Output

The output contains a single line that represents the minimum number of photos deleted such that the the total size of the remaining photos is less than or equal to the storage limit.

Sample Input

```
5 100
50 40 30 20 10
```

Sample Output

```
1
```

Ocean Depth Range

You are traveling in a submarine, recording the submarine's depth every minute. As you go deeper into the ocean, you travel through different "ocean zones". Here are the five "ocean zones" and the depths at which they start and end:

Sunlight Zone: 0 to 200 meters

Twilight Zone: 201 to 1000 meters

Midnight Zone: 1001 to 4000 meters

Abyssal Zone: 4001 to 6000 meters

Trenches: 6001 and beyond (infinite depth)

Write a program takes in a list/array of the recorded submarine depths and outputs the number of times your submarine was recorded as being in each ocean zone.

Input

The first line of the input is the number of recorded depths you will be given. Each following line is a number that represents a recorded depth.

Output

For each zone, output the number of times a recorded depth was in that zone. **If you were never in that zone, put zero.**

Make sure that your output is correctly ordered (starting with the "Sunlight Zone" and ending with "Trenches"). The provided sample code handles this for you.

Sample Input

```
8
50
250
1200
5000
7000
180
3490
900
```

Sample Output

```
Sunlight Zone: 2
Twilight Zone: 2
Midnight Zone: 2
Abyssal Zone: 1
Trenches: 1
```

Soccer Passes

You are a soccer player on a team practicing precise passes. The coach has drawn a soccer field as a coordinate plane, where:

- The center of the field is (0, 0).
- Each player stands at a specific coordinate point.

Your job is to calculate the distance between two players after a pass.

Input

The input consists of two lines:

1. The coordinates of Player A, (x_1, y_1).
2. The coordinates of Player B, (x_2, y_2).

Output

Print the distance between Player A and Player B, rounded to two decimal places.

Sample Input

```
3 4
7 1
```

Sample Output

```
5.00
```

Explanation

The distance between (3,4) and (7,1) is:

$\sqrt{(7-3)^2 + (1-4)^2} = \sqrt{16+9} = \sqrt{25} = 5.00$

Guess Who

You are hosting a party, and there's some mystery guests you're trying to identify! Each guest has a name and exactly three traits. Your job is to write a program that outputs all of the guests that match both of the given clues.

If no one matches the two clues, your program should print "No match found". If multiple people match the given clues, output each person's name, separated by a space. If only one person matches the clues, simply output their name.

Input

The first line contains a number that represents the number of test cases (up to 10).

For each test case there will be:

- A line that contains the number of guests at the party.
- The next lines describe each guest. Each line contains the guest's name and their three traits, separated by spaces (e.g., Alice blonde tall glasses).
- The last line contains exactly 2 clues, separated by a space (e.g., blonde glasses).

Output

The output contains a single line with either the name or names of the guests who fit the description or the phrase "No match found".

Sample Input

```
2
3
Alice blonde tall glasses
Bob brown tall hat
Charlie blonde short glasses
blonde glasses
2
Diana red tall glasses
Eve blue short hat
red hat
```

Sample Output

```
Alice Charlie
No match found.
```

Robot Navigator

You have a robot on a grid that starts at position $(0, 0)$. The robot follows a sequence of instructions to move around the grid. The grid has an infinite size in all directions. Your task is to determine the final position of the robot after it has executed all the instructions.

The possible instructions are:

- `UP x` - Move x units up
- `DOWN x` - Move x units down
- `LEFT x` - Move x units left
- `RIGHT x` - Move x units right

Input

The first line contains the number of instructions. The next lines contain all of the instructions (one instruction per line). The instructions have the direction and the number of units that the robot will move in the specified direction.

Output

The output contains one line with the final coordinates of the robot in the format: `x y`

Sample Input

```
5
UP 3
RIGHT 4
DOWN 2
LEFT 1
UP 1
```

Output

```
3 2
```

Fashion

Emily is planning her outfits for the week. She has a list of all her shirts, pants, and shoes, and their associated colors. She wants to know all the possible color combinations if she chooses one shirt, one pair of pants, and one pair of shoes. Write a program that generates all of the unique color combinations for her outfit.

Input

1. The first line is the number of shirts Emily has.
2. The second line is the color of each of the shirts, separated by a space.
3. The third line is the number of pants Emily has.
4. The fourth line is the color of each of the pants, separated by a space.
5. The fifth line is the number of shoes Emily has.
6. The sixth line is the color of each of the shoes, separated by a space.

Output

The output will consist of every possible combination with one on each line. Each line of your output should be of the form: `{shirt color} - {pants color} - {shoes color}`

Your program should follow the ordering shown in the below sample output.

Sample Input

```
2
red blue
3
green black yellow
2
orange white
```

Sample Output

```
red - green - orange
red - green - white
red - black - orange
red - black - white
red - yellow - orange
red - yellow - white
blue - green - orange
blue - green - white
blue - black - orange
blue - black - white
blue - yellow - orange
blue - yellow - white
```

Car Chase

A rogue criminal is escaping in a car through a city grid, and it's your mission to catch them before they vanish. Luckily, the criminal's car broke down while they were still within city limits.

The city is represented as a 2D grid. The rogue criminal's car breaks down at a at a specific coordinate. You also start at a specific coordinate and can move one block in any direction per minute. You can also assume that the criminal will start with an x and y coordinate greater than or equal to your starting x and y coordinates.

Input

The first line consists of an integer, T, representing the number of test cases.

For each test case:

The first line contains two integers, X__criminal and Y__criminal, the starting coordinates of the criminal.

The second line contains two integers, X__you and Y__you, the starting coordinates of your car.

Sample Input

```
1
5,5
1,1
```

Output

The output consists of 1 line for each test case that contains the amount of time it takes (in minutes) to catch up to the criminal.

Sample Output

```
8
```