PLU February 2017 Programming Contest

Novice Division

I. General Notes

1. Do the problems in any order you like. They do not have to be done in order from 1 to 12.

2. Problems will have either no input or will read input from standard input (stdin, cin, System.in -- the keyboard). All output should be to standard output (the monitor).

3. All input is as specified in the problem. Unless specified by the problem, integer inputs will not have leading zeros.

4. Your program should not print extraneous output. Follow the form exactly as given in the problem.

II. Names of Problems

Number	Name
Problem 1	Moo
Problem 2	Bunnies
Problem 3	Shipping
Problem 4	Simple Sum
Problem 5	Quack
Problem 6	Crowing
Problem 7	Acres
Problem 8	Tractor Path
Problem 9	Goats
Problem 10	Chicken Pen
Problem 11	Livestock Count
Problem 12	Cowspeak

1. Moo

You've decided to buy a farm and start a new life. To pass some time while you wait for the title of the land to go through, produce the following ascii cow.

Input
There is no input data for this problem.

Output
```
12345678901
2(___)
3(o o)_____/
4 @@       \
5  \ ____,/
6  //    //
7 ^^    ^^
```

Output to Screen
```
(___)
(o o)_____/
 @@       \
  \ ____,/
  //    //
 ^^    ^^
```

2. Bunnies

You're going to raise farm animals and you decided to start with bunnies, the easiest of animals. To your surprise they are breeding like rabbits, so much so that you're unable to count them accurately. However, you know that rabbits' breeding patterns always follow the Fibonacci sequence. The Fibonacci sequence is defined as follows:

F(0) = 1, F(1) = 1, F(N) = F(N-1)+F(N-2)

Given the number of months the rabbits have been breeding, use the Fibonacci sequence to determine the number of rabbits you should have.

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 the number of months that have passed since you bought your initial pair of rabbits. 0<=x<=45

Output

For each test case, output the expected number of rabbits after `x` months.

Sample Input

```
3
0
5
45
```

Sample Output

```
1
8
1836311903
```

3. Shipping

Your animals have begun producing products, and you're honestly a little strapped for cash. Since you have far more animal byproducts than you know what to do with, you've decided to begin shipping them for extra money. Given the information in your shipping ledger, determine how much money you can expect to make.

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 items follow. The next `x` lines consist of a string, and integer, and a floating point number to two decimal places, representing the name of what was sold, the quantity, and the unit price of each item.

Output

For each test case, output the amount of money you expect to make with all of the goods you sold, rounded to two decimal places.

Sample Input

```
2
3
Eggs 12 0.75
Milk 1 2.00
Feathers 50 .02
2
Cow 1 100.00
Cheese 3 0.54
```

Sample Output

```
$12.00
$101.62
```

4. Simple Sum

You have hired someone to help you with inventory on the farm. Unfortunately, the new cowhand has very limited math skills, and they are having trouble summing two numbers. Write a program to determine if the cowhand is adding these numbers correctly.

Input

The first and only line of input contains a string of the form:

`a + b = c`

It is guaranteed that **`a`**, **`b`**, and **`c`** are single-digit integers. The input line will have exactly 9 characters, formatted exactly as shown, with a single space separating each number and arithmetic operator.

Output

Print on a single line, **`YES`** if the sum is correct; otherwise, print **`NO`**.

Sample Input	**Sample output:**
`1 + 2 = 3`	`YES`

Sample Input	**Sample output:**
`2 + 2 = 5`	`NO`

5. Quack

Now that all the crops are in you have lots of extra time. To pass time while the corn grows, produce the following ascii duck.

Input
There is no input data for this problem.

Output
```
123456789
2<(o )___
3 ( \_> /
4  "~~~"
```

Output to Screen
```
<(o )___
 ( \_> /
  "~~~"
```

6. Crowing

Farming seems to be the easy life. You have lots of time to practice your asci art. Use your computer skills to produce the following ascii roster.

Input
There is no input data for this problem.

Output

```
123456789
2,~
3( 9> //)
4 )(_///
5( \_>/
6 \__/
7  ^^
```

Output to Screen

```
,~
( 9> //)
 )(_///
( \_>/
 \__/
  ^^
```

7. Acres

You have a rectangular field you want to plant with corn. You know the dimensions of the field in yards, and you know that one bag of corn seed will cover 5 acres. Having passed all your elementary math courses you also know that 4840 square yards is equal to 1 acre. So, given the dimensions of the rectangular field in yards you must compute the number of bags of corn seed you need to plant the entire field.

Input

The first line contains two positive integers that are the dimensions of the rectangular field in yards.

Output

A positive integer that is the number of bags of corn seed needed to plant the entire field.

Sample Input	**Sample output**
`100 1000`	`5`

Sample Input	**Sample output**
`200 10500`	`87`

8. Tractor Path

You just finished disking your field and it is time to head back to the farmhouse. Some of the livestock has been acting up (you think it is the Clydesdale horses but you're not sure) and putting up obstacles to prevent you from getting home.

Input

The first line contains a single positive integer, **n**, denoting the side length of the square field. The next **n** lines will consist of a map of your field, with `'.'` denoting an open space and `'x'` denoting an obstacle. You are in the top left corner of the field and the farmhouse is in the bottom right corner of the field. You don't have a lot of gas in the tractor, so each move you make must be either down or right (no backtracking).

Output

Print **"Yes"** if there is a simple path (down and right moves only) to your farm house. If there is no simple path print **"No"**.

Sample Input	**Sample output:**
6 ..x... .x.... ....x. xx.x.. ..x...	No

Sample Input	**Sample output:**
5 .x.x. ..... xxx.. ....x .x...	Yes

9. Goats

The goats have an interesting dice game they like to play. Each goat brings their own six-sided die with specified values on the six faces. Each die is fair; that is, when it is thrown, each of its six faces is equally likely to come up on top.

The first player throws the first die and the second throws the second die. If the values shown on the top of the dice differ, the player with the higher value wins. If the values are the same, both players throw the dice again.

Given two dice with specific values, what is the probability that the first player wins?

Input

The first line of input contains six space-separated integers, representing the values written on the first player's die. The second line of input contains the values on the second player's die in the same format.

It is guaranteed that all the values are between 1 and 6, inclusive.

Output

Print, on a single line, a floating-point value representing the probability that the first player wins, rounded and displayed to exactly five decimal places. The value should be printed with one digit before the decimal point and five digits after the decimal point. The sixth digit after the decimal point of the exact answer will never be 4 or 5 (eliminating complex rounding considerations).

Sample Input	**Sample output:**
1 2 3 4 5 6 1 2 3 4 5 6	0.50000

Sample Input	**Sample output:**
4 4 4 4 1 1 3 3 3 3 3 3	0.66667

10. Chicken Pen

The free-range chickens are out of control, and it is time to build a pen for them. You need to draw the smallest square pen that is large enough to hold all the chickens.

Input
The first line will have a positive integer that denotes the number of chickens.

Output
A drawing of the smallest square pen that is large enough to hold all the chickens. The symbol `'x'` will denote the perimeter of the fence, and a `'.'` will denote a space for chickens inside the pen.

Sample Input	**Sample Output**
1	xxx x.x xxx

Sample Input	**Sample Output**
5	xxxxx x...x x...x x...x xxxxx

11. Livestock Count

Print a table that describes the current count of all your livestock.

Input
There is no input data for this problem.

Output
Print the table below as shown. The character "-", is a dash not an underscore.

Sample Input
none

Output to Screen

```
Animal      Count
----------------
Chickens      100
Clydesdales     5
Cows           40
Goats          22
Steers          2
```

12. Cowspeak

The animals are conspiring again. The cows are mooing very strangely and you're certain of it. Your theory is that the length of each "MM" sound and the length of the "OO" sound can be translated to hexadecimal values and then converted to ascii. Create a program to translate cowspeak to English so you can find out what they're up to.

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 one line containing many space separated strings representing the cows' moos.

Output

For each test case, output the decoded cowspeak

Sample Input

```
2
MMMMOOOOOOOOOOOOO MMMMOOOOOOOOOOOOOOO MMMMOOOOOOOOOOOOOOO
MMMMMMOOOOOOOO MMMMMMOOOOOOOOO
```

Sample Output

```
MOO
hi
```