WEEK – 2

Problem - 1

Candy I

Problem code: CANDY

Jennifer is a teacher in the first year of a primary school. She has gone for a trip with her class today. She has taken a packet of candies for each child. Unfortunatelly, the sizes of the packets are not the same.

Jennifer is afraid that each child will want to have the biggest packet of candies and this will lead to quarrels or even fights among children. She wants to avoid this. Therefore, she has decided to open all the packets, count the candies in each packet and move some candies from bigger packets to smaller ones so that each packet will contain the same number of candies. The question is how many candies she has to move.

Input specification

The input file consists of several blocks of data. Each block starts with the number of candy packets **N**(1<= **N**<=10000) followed by **N** integers (each less than 1000) in separate lines, giving the number of candies in each packet. After the last block of data there is the number -1.

Output specification

The output file should contain one line with the smallest number of moves for each block of data. One move consists of taking one candy from a packet and putting it into another one. If it is not possible to have the same number of candies in each packet, output the number -1.

Example

Input file:

```
5
1
1
1
1
6
2
3
4
-1
```

Output file:

```
4
-1
```

Problem - 2

Julka

Problem code:JULKA

Julka surprised her teacher at preschool by solving the following riddle:

Klaudia and Natalia have 10 apples together, but Klaudia has two apples more than Natalia. How many apples does each of he girls have?

Julka said without thinking: Klaudia has 6 apples and Natalia 4 apples. The teacher tried to check if Julka's answer wasn't accidental and repeated the riddle every time increasing the numbers. Every time Julka answered correctly. The surprised teacher wanted to continue questioning Julka, but with big numbers she could't solve the riddle fast enough herself. Help the teacher and write a program which will give her the right answers.

Task

Write a program which

- reads from standard input the number of apples the girls have together and how many more apples Klaudia has,
- counts the number of apples belonging to Klaudia and the number of apples belonging to Natalia,
- writes the outcome to standard output

Input

Ten test cases (given one under another, you have to process all!). Every test case consists of two lines. The first line says how many apples both girls have together. The second line says how many more apples Klaudia has. Both numbers are positive integers. It is known that both girls have no more than 10^{100} (1 and 100 zeros) apples together. As you can see apples can be very small.

Output

For every test case your program should output two lines. The first line should contain the number of apples belonging to Klaudia. The second line should contain the number of apples belonging to Natalia.

Example

Input:

```
10
2
[and 9 test cases more]
```

Output:

```
6
4
[and 9 test cases more]
```

Problem - 3

The Last Digit

Problem code: LASTDIG

Nestor was doing the work of his math class about three days but he is tired of make operations a lot and he should deliver his task tomorrow. His math's teacher gives two numbers a and b. The problem consist in find the last digit of the potency of base a and index b. Help Nestor with his problem. You are given two integer numbers: the base a (0 <= a <= 20) and the index b (0 <= b <= 2,147,483,000), a and b both are not 0. You have to find the last digit of a^b.

Input

The first line of input contains an integer t, the number of test cases (t <= 30). t test cases follow. For each test case will appear a and b separated by space.

Output

For each test case output an integer per line representing the result.

Example

Input:
```
2
3 10
6 2
```

Output:
```
9
6
```

Problem - 4

Rectangles

Problem code: AE00

Byteman has a collection of N squares with side 1. How many different rectangles can he form using these squares?

Two rectangles are considered different if none of them can be rotated and moved to obtain the second one. During rectangle construction, Byteman can neither deform the squares nor put any squares upon any other ones.

Input

The first and only line of the standard input contains one integer N ($1 \leq N \leq 10000$).

Output

The first and only line of the standard output should contain a single integer equal to the number of different rectangles that Byteman can form using his squares.

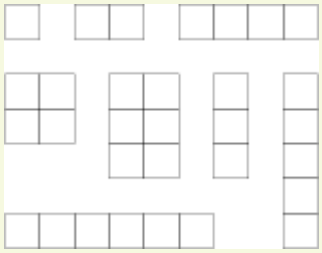
Example

For the input data:

```
6
```

the correct result is:

```
8
```



Problem - 5

Candy III

Problem code: CANDY3

A class went to a school trip. And, as usually, all **N** kids have got their backpacks stuffed with candy. But soon quarrels started all over the place, as some of the kids had more candies than others. Soon, the teacher realized that he has to step in: "Everybody, listen! Put all the candies you have on this table here!"

Soon, there was quite a large heap of candies on the teacher's table. "Now, I will divide the candies into **N**equal heaps and everyone will get one of them." announced the teacher.

"Wait, is this really possible?" wondered some of the smarter kids.

Problem specification

You are given the number of candies each child brought. Find out whether the teacher can divide the candies into **N** exactly equal heaps. (For the purpose of this task, all candies are of the same type.)

Input specification

The first line of the input file contains an integer **T** specifying the number of test cases. Each test case is preceded by a blank line.

Each test case looks as follows: The first line contains **N** : the number of children. Each of the next **N** lines contains the number of candies one child brought.

Output specification

For each of the test cases output a single line with a single word "YES" if the candies can be distributed equally, or "NO" otherwise.

Example

Input:

```
2

5
5
2
7
3
8

6
7
11
2
7
3
4
```

Output:

```
YES
NO
```

Problem - 6

Bytelandian Gold coins

Problem code: COINS

In Byteland they have a very strange monetary system.

Each Bytelandian gold coin has an integer number written on it. A coin n can be exchanged in a bank into three coins: n/2, n/3 and n/4. But these numbers are all rounded down (the banks have to make a profit).

You can also sell Bytelandian coins for American dollars. The exchange rate is 1:1. But you can not buy Bytelandian coins.

You have one gold coin. What is the maximum amount of American dollars you can get for it?

Input

The input will contain several test cases (not more than 10). Each testcase is a single line with a number n, 0 <= n <= 1 000 000 000. It is the number written on your coin.

Output

For each test case output a single line, containing the maximum amount of American dollars you can make.

Example

Input:
```
12
2
```

Output:
```
13
2
```

You can change 12 into 6, 4 and 3, and then change these into $6+$4+$3 = $13. If you try changing the coin 2 into 3 smaller coins, you will get 1, 0 and 0, and later you can get no more than $1 out of them. It is better just to change the 2 coin directly into $2.

Problem - 7

Triple Fat Ladies

Problem code: EIGHTS

Pattern Matchers have been designed for various sorts of patterns. Mr. HKP likes to observe patterns in numbers. After completing his extensive research on the squares of numbers, he has moved on to cubes. Now he wants to know all numbers whose cube ends in 888.

Given a number **k**, help Mr. HKP find the k^{th} number (indexed from 1) whose cube ends in 888.

Input

The first line of the input contains an integer **t**, the number of test cases. **t** test cases follow.

Each test case consists of a single line containing a single integer **k** (1 <= **k** <= 2000000000000).

Output

For each test case, output a single integer which denotes the k^{th} number whose cube ends in 888. The result will be less than 2^{63}.

Example

Input:
1
1

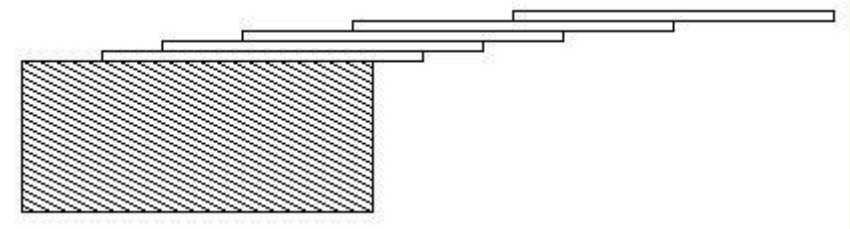
Output:
192

Problem - 8

Hangover

Problem code: HANGOVER

How far can you make a stack of cards overhang a table? If you have one card, you can create a maximum overhang of half a card length. (We're assuming that the cards must be perpendicular to the table.) With two cards you can make the top card overhang the bottom one by half a card length, and the bottom one overhang the table by a third of a card length, for a total maximum overhang of 1/2 + 1/3 = 5/6 card lengths. In general you can make *n* cards overhang by 1/2 + 1/3 + 1/4 + ... + 1/(*n* + 1) card lengths, where the top card overhangs the second by 1/2, the second overhangs tha third by 1/3, the third overhangs the fourth by 1/4, etc., and the bottom card overhangs the table by 1/(*n* + 1). This is illustrated in the figure below.



Input

The input consists of one or more test cases, followed by a line containing the number 0.00 that signals the end of the input. Each test case is a single line containing a positive floating-point number *c* whose value is at least 0.01 and at most 5.20; *c* will contain exactly three digits.

Output

For each test case, output the minimum number of cards necessary to achieve an overhang of at least *c*card lengths. Use the exact output format shown in the examples.

Input:
1.00
3.71
0.04
5.19
0.00
Output:
3 card(s)

```
61 card(s)
1 card(s)
273 card(s)
```

Problem - 9

The Next Palindrome

Problem code: PALIN

A positive integer is called a *palindrome* if its representation in the decimal system is the same when read from left to right and from right to left. For a given positive integer *K* of not more than 1000000 digits, write the value of the smallest palindrome larger than *K* to output. Numbers are always displayed without leading zeros.

Input

The first line contains integer *t*, the number of test cases. Integers *K* are given in the next *t* lines.

Output

For each *K*, output the smallest palindrome larger than *K*.

Example

Input:
2
808
2133

Output:
818
2222

Warning: large Input/Output data, be careful with certain languages

Problem - 10

Anti –Blot System

Problem code: ABSYS

Jimmy is a hard-working pupil in his second year at primary school. Recently he decided to convert all his notes into an electronic version. Sadly, he found that his math notes were full of ink blots.

He scanned the notes and sent them through his own OCR package (yes, he coded it all by himself at the age of 8). The OCR package replaced all ink blots by the string "machula".

Problem specification

You are given Jimmy's notes, processed by the OCR. They contain simple math exercises, which were used to practice addition on positive integers. Your task is to recover the damaged part of the notes.

Input specification

The first line of the input file contains an integer **T** specifying the number of test cases. Each test case is preceded by a blank line.

Each test case consists of exactly one line. The line represents an equation of the form "number + number = number", where each number is a positive integer. One part of the equation will be replaced by the string "machula". The string always covers a contiguous non-empty sequence of digits, possibly even an entire number. You may assume that for each equation in the input there will be exactly one way to fill in the missing digits.

Output specification

For each test case, the output shall contain one line of the form "number + number = number". The line must represent the equation from that test case with all missing digits filled in.

Example

Input:

```
3

23 + 47 = machula

3247 + 5machula2 = 3749

machula13 + 75425 = 77038
```

Output:

```
23 + 47 = 70
3247 + 502 = 3749
1613 + 75425 = 77038
```

Note: **int** in C++/C/Java or **longint** in Pascal is enough.