Problem A: Doctors Office

Doctor Karimi had created an office, working on Image Processing stuff. In it's first day of work 3 applicants send their resume and their expected salary. Dr. Karimi had decided to accept just one of the applicants and as their previous works are same in value he just was the same, he has decided to reject the person who wants the most salary and the one who wants the least.

You will be given the salaries of these 3 applicants for DrKROIP (Doctor Karimi's reseach office on Image Processing). You have to find out the salary of the person who will be accepted.

Input

The first line of input is an integer T ($T < 20$) that indicates the number of test cases. Each case consists of a line with 3 distinct positive integers. These 3 integers represent the salaries of the three employees. All these integers will be in the range $[1000, 10000]$.

Output

For each case, output the case number followed by the salary of the person who survives.

Sample Input

```
3
1000 2000 3000
3000 2500 1500
1500 1200 1800
```

Sample Output

```
Case 1: 2000
Case 2: 2500
Case 3: 1500
```

Problem B: ZamZam Man

K1 has opend a supermarket and in order to increase his income, he had done something cool. Every b empty bottles of zamzam you return you get one full bottle of zamzam.

Saba is so happy as she love zamzam a lot. she currenly can by a bottles with her money now with this inovation K1 did she wants to calculate who many bottle she can have at last.

So the problem is:

Having initially a ($1 \leq a \leq 1000$) ZamZam bottles and knowing that you can get a free ZamZam bottle for returning b ($2 \leq b \leq 1000$) empty bottles, what is the maximum number of ZamZam bottles you can drink ?

Input

The input contains two numbers in every line. These two numbers in each line denotes the number of bottles Saba can buy at first (a). Then b is the number of empty bottles K1 accepts to change with a full bottle.

Output

A single line containing the number of bottles Saba can have at last.

Sample Input

```
4 2
6 3
```

Sample Output

```
7
8
```

Problem C: Easy GCD

Greatest common divisor $GCD(a, b)$ of two positive integers a and b is equal to the biggest integer d such that both integers a and b are divisible by d. There are many efficient algorithms to find greatest common divisor $GCD(a, b)$, for example, Euclid algorithm. Formally, find the biggest integer d, such that all integers a,$a+1$,$a+2$,...,b are divisible by d. To make the problem even more complicated we allow a and b to be up to googol, 10^{100} such number do not fit even in 64-bit integer type!

Input

each line of input contains two integers a and b ($1 \leq a \leq b \leq 10^{100}$). input is terminated with EOF.

Output

for each test case output one integer greatest common divisor of all integers from a to b inclusive.

Sample Input

```
1 2
6180339887498948482045868343656381177203091798057 6
6180339887498948482045868343656381177203091798057 6
```

Sample Output

```
1
6180339887498948482045868343656381177203091798057 6
```

Problem D: Alireza's Continues Hours of Sleeping

Alireza is one of cool members of ACM chapter but he has a very strange behaviour. ACM chapter send him to a doctor and after weeks of studing this case doctors returned a report.

"First day (Today) Alireza Slept h hours and the next day (tommorow) he will sleep one minute more than today. Each day an strange part of his brain will remember the yesterday's (a_{-1}) and the day before yesterday's (a_{-2}) oversleeping and for the the night he will sleep h hours plus the two remembered numbers in minutes. A day will come in which he will sleep 24 hours and that day winter sleep begins for a month and after that he will be cured and he will sleep 8 hours a day". Knowing that fact alireza wants to calculate the exact remaining days left till he reach his winter sleep days. help him to find that.

Input

Each line of input consists of a single integer h denoting the hours alireza slept today. ($0 \leq h \leq 24$)

Output

for each test case output one integer denoting the days he has from today to the day of winter sleeping. (modolo $10^9 - 1$)

Sample Input

9
24

Sample Output

15
0

Problem E: Sorry! I didn't get that!

Sometimes one has to spell email addresses over the phone. Then one usually pronounces a dot as "dot" , an at sign as "at" . As a result, we get something like acmatiutdotacdotir . Your task is to transform it into a proper email address (acm@iut.ac.ir).

It is known that a proper email address contains only such symbols as . @ and lowercase Latin letters, doesn't start with and doesn't end with a dot. Also, a proper email address doesn't start with and doesn't end with an at sign. Moreover, an email address contains exactly one such symbol as @ , yet may contain any number (possible, zero) of dots. You have to carry out a series of replacements so that the length of the result was as short as possible and it was a proper email address. If the lengths are equal, you should print the lexicographically minimal result.

Overall, two variants of replacement are possible: dot can be replaced by a dot, at can be replaced by an at. In the ASCII table the symbols go in this order: . @ ab ... z

Input

A single line of string

Output

A single line of string

Sample Input

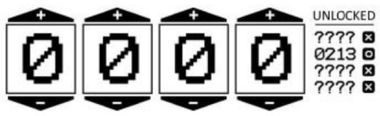
```
acmatiutdotacdotir
```

Sample Output

```
acm@iut.ac.ir
```

Problem F: ACM CraftLock Manufacture

Lately, there is one serious problem with San@i's Safe Box: several safes have been robbed! The safes are using old 4-digits rolling lock combination (you only have to roll the digit, either up or down, until all four of them match the key). Each digit is designed to roll from 0 to 9. Rolling-up at 9 will make the digit become 0, and rolling-down at 0 will make the digit become 9. Since there are only 10000 possible keys, 0000 through 9999, anyone can try all the combinations until the safe is unlocked.



Whats done is done. But in order to slow down future robbers attack, ACM CraftLock Manufacture (ACM) has devised a new safer lock with multiple keys. Instead of using only one key combination as the key, the lock now can have up to N keys which has to be all unlocked before the safe can be opened. These locks works as the following:

- Initially the digits are at 0000.
- Keys can be unlocked in any order, by setting the digits in the lock to match the desired key and then pressing the UNLOCK button.
- A magic JUMP button can turn the digits into any of the unlocked keys without doing any rolling.
- The safe will be unlocked if and only if all the keys are unlocked in a minimum total amount of rolling, excluding JUMP (yes, this feature is the coolest one).
- If the number of rolling is exceeded, then the digits will be reset to 0000 and all the keys will be locked again. In other word, the state of the lock will be reset the cracking is failed.

ACM is quite confident that this new system will slow down the cracking, giving them enough time to identify and catch the robbers. In order to determine the minimum number of rolling needed, PSA wants you to write a program. Given all the keys, calculate the minimum number of rolls needed to unlock the safe.

Input

The first line of input contains an integer T, the number of test cases follow. Each case begins with an integer N ($1 \leq N \leq 500$), the number of keys. The next N lines, each contains exactly four digits number (leading zero allowed) representing the keys to be unlocked.

Output

For each case, print in a single line the minimum number of rolls needed to unlock all the keys. Explanation for the 2nd case:

- Turn 0000 into 1111, rolls: 4
- Turn 1111 into 1155, rolls: 8
- Jump 1155 into 1111, we can do this because 1111 has been unlocked before. Turn 1111 into 5511 rolls: 8

Total rolls = 4 + 8 + 8 = 20

Sample Input

```
4
2 1155 2211
3 1111 1155 5511
3 1234 5678 9090
4 2145 0213 9113 8113
```

Sample Output

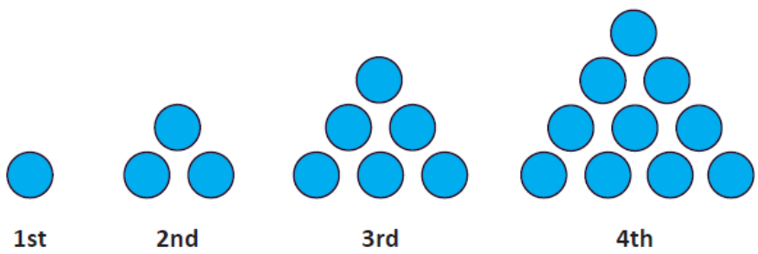
```
16
20
26
17
```

Problem G: San@i's Azadi Tower - Student are Legos

Once a group of nerds in their dorm was bored from lots of facilities and fantastic things univeristy had brought them and they decided to do something new. The result was this:

But the other day they decided to create a bigger strudent (structure made of students). Making a bigger strudent is done by adding a row below the smaller strudent using following structure:

Given the height of the strudent you are to calculate how many students are needed!



Input

The first line denotes the number of test cases T. Then each of the following T lines has a single integer denoting the height of strudent.

Output

For each test case print a line like the following manner: `We need n student(s).`

Sample Input

```
3
1
2
3
```

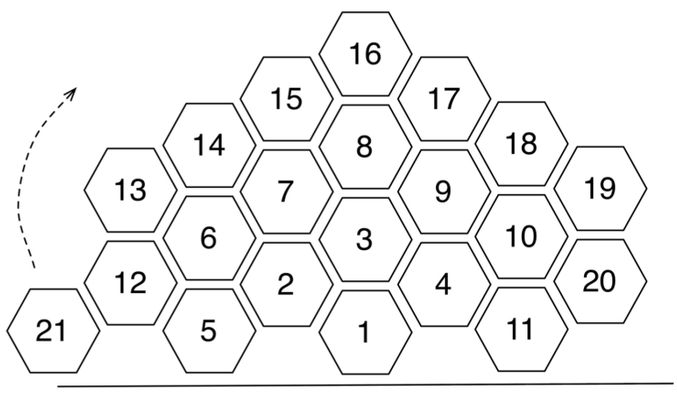
Sample Output

```
We need 1 student.
We need 3 students.
We need 6 students.
```

Problem H: Beehive

There is an infinite beehive like the one given in the figure. We consider two cells to be adjacent if and only if they share a side. A path of length k from cell c_0 to cell ck is a sequence of cells $c_0, c_1, ..., c_k$ such that ci and $ci + 1$ are adjacent for all $0 \leq i < k$. The distance between cells i and j is the length of the shortest path from cell i to cell j.

The cells of the beehive are indexed using positive integers as shown. The cells with larger distance from cell 1 are given larger indices. The indices of cells with the same distance from cell 1 increases from left to right. Each positive integer is the index of exactly one cell.



We want to know the distance of two cells whose indices are given.

Input

There are multiple test cases in the input. Each test case is a single line containing two space-separated integers i and j as the indices of two cells ($1 \leq i, j \leq 10^4$). The input terminates with a line containing 0 0 which should not be processed as a test case.

Output

For each test case, output a single line containing the distacne of the given cells.

Sample Input

8	4
11	12
365	365
0	0

Sample Output

2
5
0

Problem I: Number of Ways

You've got array $a[1], a[2], ..., a[n]$, consisting of n integers. Count the number of ways to split all the elements of the array into three contiguous parts so that the sum of elements in each part is the same. More formally, you need to find the number of such pairs of indices i, j $(2 \leq i \leq j \leq n-1)$, that $\sum_{k=1}^{i-1} a_k = \sum_{k=i}^{j} a_k = \sum_{k=j+1}^{n} a_k$.

Input

Input consist of several test cases each of them contains integer n $(1 \leq n \leq 5 * 10^5)$, showing how many numbers are in the array. The second line of each test case contains n integers $a[1], a[2], ..., a[n]$ $(|a[i]| \leq 10^8)$, the elements of array a. Input ends with EOF.

Output

For each test case Print a single integer, the number of ways to split the array into three parts with the same sum.

Sample Input

```
5
1  2  3  0  3
4
0  1  −1  0
2
4  1
```

Sample Output

```
2
1
0
```

Problem J: Find me if You Can

Given an array of n elements, for each segment of the fixed length k you must find the maximum element of those that occur on the given segment exactly once. ($1 \leq n \leq 10^5$, $1 \leq k \leq n$) ($-10^9 \leq a_i \leq 10^9$) Print $n-k+1$ numbers, one per line: on the i-th line print of the maximum number of those numbers from the subarray a_i a_{i+1} ... $ai+k-1$ that occur in this subarray exactly 1 time. If there are no such numbers in this subarray, print "Nothing".

Input

The first line contains two positive integers n and k ($1 \leq n \leq 10^5, 1 \leq k \leq n$). The number of array elements and the lenght of segments.

Then follows n lines: the i-th one contains a single number a_i ($-10^9 \leq a_i \leq 10^9$)

Output

print $n-k+1$ numbers, one per line: on the i-th line print of the maximum number of those numbers from the subarray $a_i, a_{i+1}...a_{i+k-1}$ that occur in this subarray exactly 1 time. If there are no such number s in the subarray print "`Nothing`".

Sample Input

```
5 3
1
2
2
3
3
```

Sample Output

```
1
3
2
```