Problem A: Connection Duration

Source: A.(c|cpp|py|java)
Input: A.in
Output: A.out

Problem Statement: A Coffee Net requested you to write a program to calculate connection time of each computer where serves an internet service to users on site. Each computer in this Coffee Net has a unique IP address usually in the range of 192.168.0.3 to 192.168.0.254. The other IP address mean192.168.0.1, 192.168.0.2 and 192.168.0.255 are dedicated to master computers in Coffee Net so they cannot be used by users. You should read the connection time and also end time for each IP from input file and calculate the cost based on announced rate by coffee net.

Input: There are several lines in input file. Each line equipped by an IP address, start and end connection time in format of hh:mm:ss. 00.00.00 is start point of each day and 23:59:59 is end time of each time. We assume that the program is restarted in each day. Like metro electronic tickets in Tehran if we have no track of end time we will assign 23:59:59 to that in order to calculate the connection time. At the end of file you will see a number which indicates the rate of charge in second. Instead of errors for input file we talk about them here there is no other error in input file.

Output: for each IP you have to write IP address and fee. For non allowed IP address in input file you have to write this error message. NOT ALLOWED

Sample Input file:	**Output for sample input:**
192.168.0.1 192.168.0.5 11:20:00 11:40:05 192.168.0.45 14:14:14 18:20:33 192.168.0.5 11:41:00 12:01:12 192.168.0.12 21:30:00 2	192.168.0.1 NOT ALLOWED 192.168.0.5 4834 192.168.0.45 29558 192.168.0.12 17998

Problem B: Reverse Sentences

Source: B.(c|cpp|py|java)
Input: B.in
Output: B.out

Problem Statement: We have installed a word processor but it doesn't support left-to-right languages. It shows all lines from right to left so we need to change the order of words in the document. Write a program that reorders words so that in each line words are in the reverse order.

Input: The input file starts with the integer N on a line by itself this is the number of lines to follow. The following n lines each contain one string of at most 100 letters.

Output: For each line in the input, first output the number of the line, as shown in the sample output. Then print the string that is read from the input. Print a blank line after each string.

Sample Input:

2
.test a is This
one this Reorder

Sample Output:

Line #1
This is a test.

Line #2
Reorder this one

Problem C: Additional Parentheses Removal

Source: C.(c|cpp|py|java)
Input: C.in
Output: C.out

Problem Statement: Some students are scrupulosity in working with operations in programming. They usually put several additional parentheses. You are a TA of Mr. Hooshmand's programming course at HUT. He asked you to write a program to remove these additional parentheses from students' home work.

Input: In the first line we have a positive number which indicates the number of statements. We assume that there is no compiler error for each statement.

Output: for each statement you must rewrite code without additional parentheses. Also additional spaces should remove from the statements.

Sample Input file:	**Output for sample input:**
2 ((A – B) * ((- A) + (B))) -A- B + ((((C))))	(A-B)*((-A)+B)) -A-B+C

Problem D: Sum of Digits

Source: D.(c|cpp|py|java)
Input: D.in
Output: D.out

Problem Statement:
Sum of digits of a number is sometime important to clarify a class which the number belongs to. Your task is very easy. You should read numbers from input file and calculate sum of digits in that. For the task just positive numbers plus zero are acceptable.

Input
The first line in input file is number of cases. Next numbers are cases. In input file you may visit negative number.

Output
You should calculate sum of digits for each number and put it in a separate line in output file. If the number is negative you must write the “WRONG” phrase in the output file.

Sample Input file:	**Output for sample input:**
3 0 100089 48987 -59	0 18 36 WRONG

Problem E: And Then There Was One

Source: E.(c|cpp|py|java)
Input: E.in
Output: E.out

Problem Statement:
Let's play a stone removing game.
Initially, n stones are arranged on a circle and numbered 1, . . . , n clockwise (Figure 1). You are also given two numbers k and m. From this state, remove stones one by one following the rules explained below, until only one remains. In step 1, remove stone m. In step 2, locate the k-th next stone clockwise from m and remove it. In subsequent steps, start from the slot of the stone removed in the last step, make k hops clockwise on the remaining stones and remove the one you reach. In other words, skip (k − 1) remaining stones clockwise and remove the next one.
Repeat this until only one stone is left and answer its number.
For example, the answer for the case n = 8, k = 5, m = 3 is 1, as shown in Figure 1.

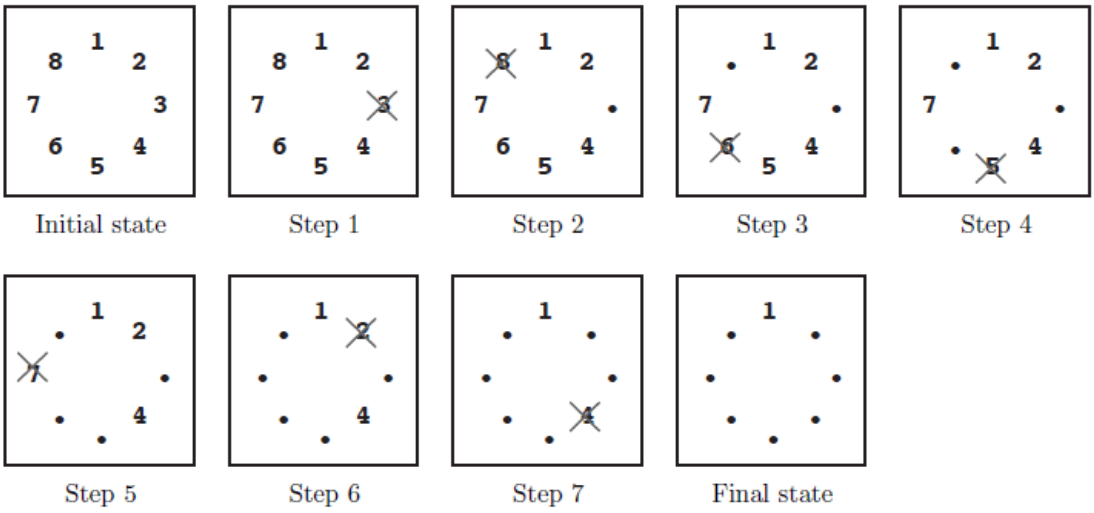

Figure 1: An example game

Initial state: Eight stones are arranged on a circle.
Step 1: Stone 3 is removed since m = 3.
Step 2: You start from the slot that was occupied by stone 3. You skip four stones 4, 5, 6 and 7 (since k = 5), and remove the next one, which is 8.
Step 3: You skip stones 1, 2, 4 and 5, and thus remove 6. Note that you only count stones that are still on the circle and ignore those already removed. Stone 3 is ignored in this case.
Steps 4–7: You continue until only one stone is left. Notice that in later steps when only a few stones remain, the same stone may be skipped multiple times. For example, stones 1 and 4 are skipped twice in step 7.
Final State: Finally, only one stone, 1, is on the circle. This is the final state, so the answer is 1.

Input
The input consists of multiple datasets each of which is formatted as follows.

n k m

The last dataset is followed by a line containing three zeros. Numbers in a line are separated by a single space. A dataset satisfies the following conditions.

$$2 \leq n \leq 10000, \quad 1 \leq k \leq 10000, \quad 1 \leq m \leq n$$

The number of datasets is less than 100.

Output
For each dataset, output a line containing the stone number left in the final state. No extra characters such as spaces should appear in the output.

Sample Input
8 5 3
100 9999 98
10000 10000 10000
0 0 0

Output for the Sample Input
1
93
2019