



2013 ACM ICPC Southeast USA Regional Programming Contest

2 November, 2013

Division 2

A: Cut the Cake	1
B: The n Days of Christmas	3
C: Ping!	4
D: Electric Car Rally.....	5
E: Count your Cousins	7
F: Decimal Representation	9
G: Politics	10
H: Perfect Shuffle	12
I: Speed Can Cost You.....	13
J: Text Roll.....	14

Hosted by:

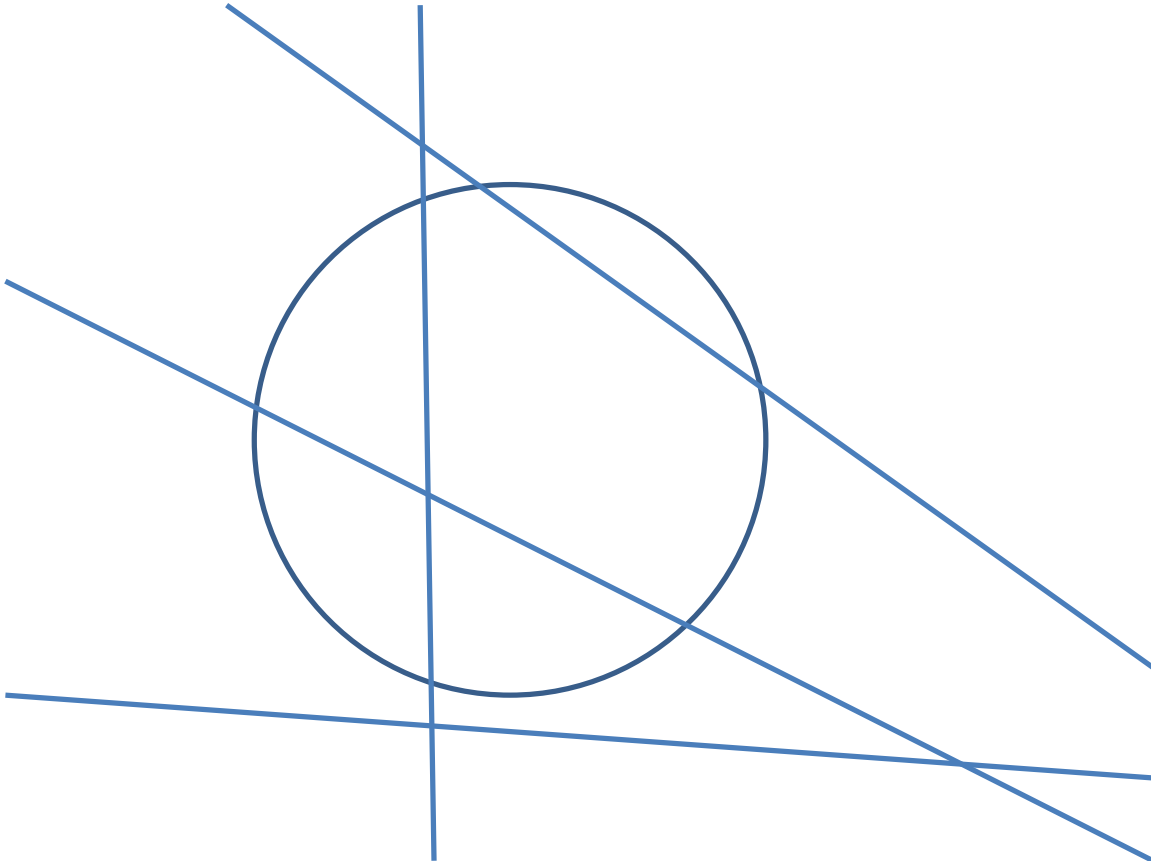
Florida Institute of Technology
Georgia Institute of Technology
University of West Florida





A: Cut the Cake

Given a circle, and a list of lines, how many parts has the circle been cut into? In the following example, four lines cut the circle into five parts.



Input

There will be several test cases in the input. Each test case will begin with four integers, r ($1 \leq r \leq 1,000$), x , y ($-1,000 \leq x, y \leq 1,000$), and n ($0 \leq n \leq 1,000$), where r is the radius of a circle, x and y are the coordinates of the center of the circle, and n is the number of lines. On each of the next n lines will be four integers, x_1 , y_1 , x_2 and y_2 ($-1,000 \leq x_1, y_1, x_2, y_2 \leq 1,000$). These describe a line passing through (x_1, y_1) and (x_2, y_2) . Note that we're interested in the whole, infinite line, not just the segment between (x_1, y_1) and (x_2, y_2) . In any test case, no more than two lines will intersect at any point inside the circle, no lines will be tangent to the circle, and no two lines will be coincident. The input will end with a line with four 0s.

Output

For each test case, output a single integer, indicating the number of parts that the circle is cut into. Output no spaces, and do not separate answers with blank lines.



Sample Input	Sample Output
16 1 7 4 -15 -9 14 -11 -4 30 -3 -20 -20 12 -10 7 17 10 31 0 0 0 0 0	5



B: The n Days of Christmas

On the first day of Christmas, my true love gave to me, a partridge in a pear tree.

Great. I wanted an iPad.

On the second day of Christmas, my true love gave to me, two turtle doves, and a partridge in a pear tree.

OK, now I have 4 birds: two turtle doves and a partridge today, and a partridge from yesterday. My cats are going crazy.

On the third day of Christmas, my true love gave to me, three french hens, two turtle doves, and a partridge in a pear tree.

What's with the birds? I now have ten of them, and no place to put them (except inside the cats). How long can my true love keep this up?

Given a number of days of Christmas, figure out the total number of gifts I'm going to receive.

Input

There will be several test cases in the input. Each test case will consist of a single integer n ($1 \leq n \leq 1,000,000$), on its own line. The input will end with a line with a single **0**.

Output

For each test case, output a single integer, indicating the number of gifts I'm going to receive, in total. Output no spaces, and do not separate answers with blank lines.

Sample Input	Sample Output
1	1
2	4
3	10
0	



C: Ping!

Suppose you are tracking some satellites. Each satellite broadcasts a ‘Ping’ at a regular interval, and the intervals are unique (that is, no two satellites ping at the same interval). You need to know which satellites you can hear from your current position. The problem is that the pings cancel each other out. If an even number of satellites ping at a given time, you won’t hear anything, and if an odd number ping at a given time, it sounds like a single ping. All of the satellites ping at time 0, and then each pings regularly at its unique interval.

Given a sequence of pings and non-pings, starting at time 0, which satellites can you determine that you can hear from where you are? The sequence you’re given may, or may not, be long enough to include all of the satellites’ ping intervals. There may be satellites that ping at time 0, but the sequence isn’t long enough for you to hear their next ping. You don’t have enough information to report about these satellites. Just report about the ones with an interval short enough to be in the sequence of pings.

Input

There will be several test cases in the input. Each test case will consist of a single string on its own line, with from 2 to 1,000 characters. The first character represents time 0, the next represents time 1, and so on. Each character will either be a 0 or a 1, indicating whether or not a ping can be heard at that time (0=No, 1=Yes). Each input is guaranteed to have at least one satellite that can be heard. The input will end with a line with a single 0.

Output

For each test case, output a list of integers on a single line, indicating the intervals of the satellites that you know you can hear. Output the intervals in order from smallest to largest, with a single space between them. Output no extra spaces, and do not separate answers with blank lines.

Sample Input	Sample Output
01000101101000	1 2 3 6 8 10 11 13
1001000101001000	3 6 7 12 14 15
0	



D: Electric Car Rally

In an attempt to demonstrate the practicality of electric cars, ElecCarCo is sponsoring a cross-country road rally. There are n charging stations for the rally where cars may check in and charge their batteries.

The rally may require multiple days of travel. Each car can travel four hours (240 minutes) between charges. A car must be plugged into a charger for two minutes for each minute of travel time. Cars start the rally at noon on the first day, fully charged. Cars are permitted remain at a station even after they are fully charged.

It is only possible to drive directly between select pairs of stations. Variations in traffic conditions, road conditions, availability of HOV lanes, etc., result in different travel times along each route depending upon the time of day at which travel along that route begins. All roads are two-way, and the prevailing conditions affect travel in both directions.

The winner is the first car to reach checkpoint $n-1$, starting from checkpoint 0. Other than the starting and ending conditions, cars may pass through the stations in any order, and need not visit all stations to complete the course.

Write a program to determine the earliest time, expressed as the total number of minutes elapsed since the start of the rally, at which a car could reach the final checkpoint.

The Input

There will be several test cases in the input. Each test case starts with a line containing n ($1 \leq n \leq 500$), the number of stations, and m ($1 \leq m \leq 1,000$), the number of connecting road segments.

This is followed by m blocks, each block describing one road segment. A road segment block has the following structure:

Each block begins with a single line containing two integers, a and b ($0 \leq a, b \leq n-1$, $a \neq b$). These numbers are the two checkpoints connected by that segment. The connections are undirected: a segment permitting travel from station a to station b will also allow travel from station b to station a .

This is followed by from one to twenty 'travel lines' describing travel times. Each of the travel lines contains 3 numbers: **Start**, **Stop**, ($0 \leq \text{Start} < \text{Stop} \leq 1,439$), and **Time** ($0 < \text{Time} < 1,000$). **Start** and **Stop** are the time of day (expressed in minutes since midnight) described by this line, and **Time** is the travel time, in minutes, required to traverse this road segment if travel begins at any time in the range [**Start**..**Stop**], inclusive. The first travel line in a block will have a start time of 0 (midnight, or 00:00). The final travel line in a block will have a stop time of 1439 (i.e., 23:59, or 1 less than 24 hours times 60 minutes). Adjacent travel lines in the input will be arranged in order, and



the start time of any line after the first is one higher than the stop time of the preceding line. The travel lines will cover all times from 00:00 to 23:59.

Input will end with a line with two 0s. All test cases will describe a course that can be completed by the cars.

The Output

For each test case, output a single integer representing the smallest number of minutes needed to complete the rally. Output no spaces, and do not separate answers with blank lines.

Sample Input	Sample Output
4 4	180
0 1	2360
0 1439 100	255
0 2	
0 1439 75	
1 3	
0 720 150	
721 824 100	
825 1000 75	
1001 1439 150	
2 3	
0 1439 150	
3 2	
0 1	
0 10 200	
11 1439 300	
1 2	
0 10 200	
11 1439 300	
4 3	
0 1	
0 719 500	
720 1439 240	
1 2	
0 964 500	
965 1439 2	
2 3	
0 971 500	
972 1439 3	
0 0	



E: Count your Cousins

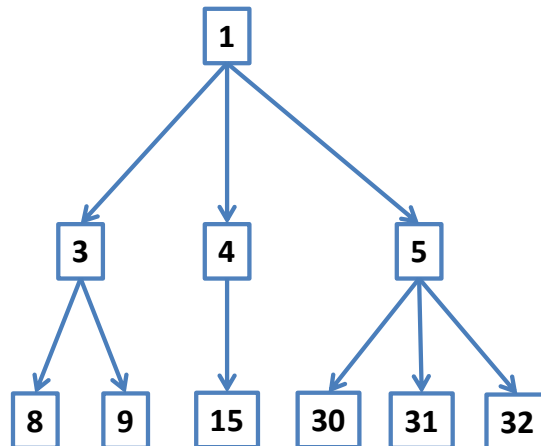
A tree is formed from a strictly increasing sequence of integers as follows:

- The first integer in the sequence is the root of the tree
- The next set of consecutive integers in the sequence describes the children of the root. The first of these will be greater than $root+1$.
- From there, each set of consecutive integers describes the children of the lowest numbered node which does not yet have children.
- Non-consecutive integers mark a break between one set of children and the next.

For example, the sequence:

1 3 4 5 8 9 15 30 31 32

Would produce the following tree:



Two nodes are considered to be *Cousins* if they have different parents, but their parents are siblings. Given a tree and a particular node of that tree, count the number of *Cousins* of the node.

Input

There will be several test cases in the input. Each test case will begin with a line with two integers, n ($1 \leq n \leq 1,000$) and k ($1 \leq k \leq 1,000,000$), where n is the number of nodes in the tree, and k is the particular node of interest. On the following line will be n integers, all in the range from 1 to 1,000,000, and guaranteed to be strictly increasing. These describe the tree, in the manner described above. The integers will be separated with a single space. There will be no extra spaces. The value k is guaranteed to be one of the integers on the second line. Input will end with a line with two 0s.



Output

For each test case, output a single integer, indicating the number of cousins of node k . Output no spaces, and do not separate answers with blank lines.

Sample Input	Sample Output
10 15	5
1 3 4 5 8 9 15 30 31 32	1
12 9	0
3 5 6 8 9 10 13 15 16 22 23 25	
10 4	
1 3 4 5 8 9 15 30 31 32	
0 0	



F: Decimal Representation

Any rational fraction can be represented in decimal notation. Using parentheses to denote repeating decimal digits, consider the following examples:

$$\begin{aligned}4/2 &= 2 \\1/4 &= 0.25 \\10/3 &= 3.(3) \\1/7 &= 0.(142857) \\1/45 &= 0.0(2)\end{aligned}$$

Each of these requires a different number of characters. $4/2 = 2$, so it requires only one character. $1/7 = 0.(142857)$, so it needs 10. Given an integer n , what is the greatest number of characters needed to represent any fraction a/b , where $1 \leq a, b \leq n$?

Input

There will be several test cases in the input. Each test case will consist of a single integer n ($1 \leq n \leq 500$) on its own line. Input will end with a line with a single **0**.

Output

For each test case, output a single integer, indicating the maximum number of characters needed to represent any a/b , where $1 \leq a, b \leq n$. Output no spaces, and do not separate answers with blank lines.

Sample Input	Sample Output
12	10
19	22
156	152
0	



G: Politics

You are helping a political party canvass their members. The party has a list of candidates they wish to support, and they also have a list of members, each of which supports one candidate.

The party wants for you to sort the list of members, according to some rules. You will be given the list of candidates, in the desired order. Then, you'll have access to a list of supporters; each supporter will have one candidate that they support.

The party wants you to put the list of supporters in order, sorted by their preferred candidate. There may be some supporters who support the same candidate. If so, order them in the order that they appear on the original list. Some supporters may have a 'write-in' candidate, thereby mentioning a candidate who was not on the original preferred candidate list. Put these supporters at the end, grouped by candidate, in the order that their *candidates* appear in the list of supporters. There also may be candidates who have no supporters. They will probably be taken off the list later, but that's just the politics of the situation.

Input

There will be several test cases in the input. Each test case will begin with a line with two integers, n ($1 \leq n \leq 1,000$) and m ($1 \leq m \leq 100,000$), where n is the number of candidates, and m is the number of supporters. On the following n lines will be the names of the candidates, one per line. These names will each be a single word consisting of from 1 to 30 capital letters. All n listed candidate names in any test case will be unique. After the candidates, the next m lines will describe the supporters, one per line. Each of these lines will contain two words separated with a single space. Each word contains from 1 to 30 capital letters. The first word is the name of the supporter, and the second is the name of the candidate that they support. The input will end with a line with two 0s

Output

For each test case, print out the names of the supporters, one name per line, in the requested order. Do not output any spaces. Do not separate answers with a blank line.

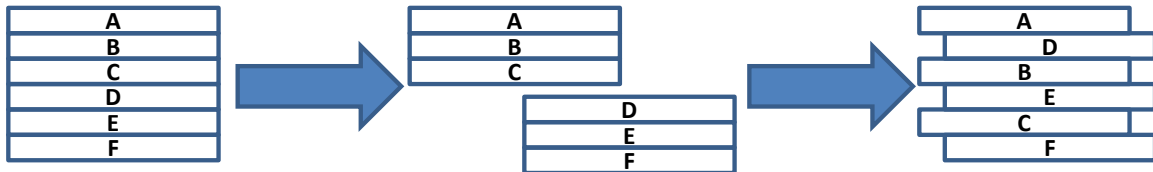


Sample Input	Sample Output
<pre>3 5 STEVENS MICHAELS JORDAN BOB JORDAN JACK STEVENS MACK MICHAELS BILL JORDAN CHRIS MATTHEWS 1 5 FRED SAM FRED ARTHUR GEORGE DANIEL HERBERT MARK GEORGE MIKE HERBERT 0 0</pre>	<pre>JACK MACK BOB BILL CHRIS SAM ARTHUR MARK DANIEL MIKE</pre>



H: Perfect Shuffle

A *Perfect Shuffle* of a deck of cards is executed by dividing the deck exactly in half, and then alternating cards from the two halves, starting with the top half.



Given a deck of cards, perform a *Perfect Shuffle*. If there is an odd number of cards, give the top half split one more card than the bottom half.

Input

There will be several test cases in the input. Each test case will begin with a line with a single integer n ($1 \leq n \leq 1,000$), indicating the number of cards. On each of the next n lines will be a string from 1 to 80 characters in length, which is the name of a card. It will contain only capital letters and dashes. Within a test case, all card names will be unique. Input will end with a line with a single 0.

Output

For each test case, output n lines, consisting of the deck after a perfect shuffle. Output no extra spaces. Do not print a blank line between answers.

Sample Input	Sample Output
4 ACE KING QUEEN JACK	ACE QUEEN KING JACK
5 SKIP DRAW-TWO REVERSE WILD WILD-DRAW-FOUR	SKIP WILD DRAW-TWO WILD-DRAW-FOUR REVERSE
0	



I: Speed Can Cost You

A speeding ticket can be expensive. Is it worth it? It's surprising how little time is gained by speeding.

Given a distance (in miles) and two speeds (in miles per hour, or MPH), determine the difference in time to travel the distance.

Input

There will be several test cases in the input. Each test case will consist of three integers on a single line, d ($1 \leq d \leq 10,000$), $s1$ and $s2$ ($1 \leq s1 < s2 \leq 1000$), where d is a distance in miles, and $s1$ and $s2$ are speeds in MPH. The integers will be separated by a single space, and there will be no leading or trailing blanks. The input will end with a line with three 0s.

Output

For each test case, output a time in the form:

H:MM:SS

Minutes (MM) and Seconds (SS) should be exactly two characters long, padded with a 0 if necessary. Seconds should be rounded. Hours should use the minimum number of digits necessary. Output no spaces. Do not print a blank line between answers.

Sample Input	Sample Output
21 70 80	0:02:15
26 65 80	0:04:30
0 0 0	



J: Text Roll

Take some text. Put a small ball at the top of the first letter of the first word of the first sentence. The ball is drawn via gravity downwards. The text is also at a slight angle, so the ball wants to also move towards the right. The ball can freely move between the lines, and can drop through spaces. Considering the first column to be column 1, what column will the ball end up in? In this example, the ball ends up in column 8.

Lorem ipsum dolor sit amet, consectetur adipiscing elit,
sed do eiusmod tempor incididunt ut labore et dolore magna
aliqua. Ut enim ad minim veniam, quis nostrud exercitation
ullamco laboris nisi ut aliquip ex ea commodo consequat.

Input

There will be several test cases in the input. Each test case will begin with an integer n ($1 \leq n \leq 1,000$) on its own line, indicating the number of lines of text. On each of the next n lines will be text, consisting of printable ASCII characters and spaces. There will be no tabs, nor any other unprintable characters. Each line will be between 1 and 100 characters long. The input will end with a line containing a single 0.

Output

For each test case, output a single integer on its own line, indicating the column from which the ball will drop. Output no spaces, and do not separate answers with blank lines.

Sample Input

```
4
Lorem ipsum dolor sit amet, consectetur adipiscing elit,
sed do eiusmod tempor incididunt ut labore et dolore magna
aliqua. Ut enim ad minim veniam, quis nostrud exercitation
ullamco laboris nisi ut aliquip ex ea commodo consequat.
```

```
3
Supercalifragilisticexpialidocious! Can you handle
short
lines?
```

```
0
```

Sample Output

```
8
```

```
36
```