## Contents

| i  | Contest Rules and Scoring                           | 1 |
| ii | Example Problem Layout                              | 2 |
| 1  | Number Chain                                         | 3 |
| 2  | Choice Numbers                                       | 4 |
| 3  | Well, Golygons                                       | 5 |
| 4  | Who hit whom?                                        | 7 |
| 5  | Bored. Bored, bored, bored.                         | 9 |
| 6  | Repeating Expansions                                | 10|
| 7  | Whoa, big!                                           | 11|
| 8  | Tree Surgeon                                         | 12|
| 9  | Words, words, words                                  | 13|
i. Contest Rules and Scoring

The contest will last three hours. Each team may have a maximum of four students. Calculators, notes, outside software, and books other than language references are prohibited. Books are subject to approval by the ACM staff. Team members may not discuss the contest with other teams or with their school sponsor during the competition.

Problem solutions will be graded as the test progresses. Each incorrect submission of a problem will result in a 10 minute time penalty added to your score. Solutions may be submitted as many times as you like. When a team is ready to submit a solution, a representative must raise his/her hand and wait for a grader. Once a hand is raised, a problem attempt will be logged, even if the team decides it is not ready. The grader can only provide the problem’s test input and state whether the solution is correct or incorrect. The graders cannot answer any questions! If a team has any questions about the problems, the grader will summon a judge. The judge’s decision is final.

The following three criteria will be used to determine final placement and prize winners:

- The team with the highest point total will be declared the winner, the second highest second place, and so on.

- At the end of the competition, should two or more teams have the same number of points, the higher rank will be awarded to the team with the lowest overall time, including penalty minutes. If a tie still exists, the team with the fewest attempts will be ranked higher.

- In the unlikely event that two or more teams are still tied after applying the tie-breaking criteria, a tie will be declared.

A note about input and output: Most of the contest problems do not depend upon exact formatting, so embedded spaces in programs’ input and output will be tolerated. If a team has any questions about how to format a program’s input or output, the team should ask a judge before submitting the problem to be graded.
ii. Example Problem Layout

This block contains a little story. Some might be humourous, and some might be really awful. We make no claims as to the quality of our fiction.

Problem Statement

This is the basic statement of the problem, along with any necessary definitions.

Notes

Typically, the input and output specifications are in the Notes section. Other included tidbits might be constraints on the arguments, hints, etc.

Examples

Example 1:
Obviously, this is the example section.
Computer output is shown like this, and input like this.

Example 2:
Exact input and output formats rarely matter, but it would greatly simplify our grading if your programs appear similar to our examples.
1. Number Chain

Yeah, you were caught skipping class. Ignoring that it was to go to a lecture by a Nobel laureate (What? Learn something?), your wonderful administration has decided to sentence you to hard labor in a chain gang. This isn’t just any chain gang, though; this one is run by the math department.

You show up to find that you are to repeatedly calculate number chains, which are the repeating loops formed by subtracting the number with its digits sorted two different ways. Each worker has been given a set of numbers; any mistakes result in being given twice as many numbers on which to work. Naturally, you think, this kind of mindless work is perfect for a computer. No one notices when you slip out your laptop...

Problem Statement

Write a program to find the chain associated with any given number.

Start with the given positive integer. This is the first number in the chain. Sort the digits (of the base 10 number) into ascending order, and also in descending order. Subtract the ascending number from the descending number. This is the next number in the chain. Continue until you find a cycle.

Notes

Input will consist of a number of at most 6 digits, and the chain will repeat within 20 steps.

The output should show each step in finding the chain along with both the length until the chain is identified and the length of the chain.

Examples

Example 1:
Enter an integer: 56472
76542 - 24567 = 51975
97551 - 15579 = 81972
98721 - 12789 = 85932
98532 - 23589 = 74943
97443 - 34479 = 62964
96642 - 24669 = 71973
97731 - 13779 = 83952
98532 - 23589 = 74943
Length until repeat: 8
Length of chain: 4

Example 2:
Enter an integer: 893
983 - 389 = 594
954 - 459 = 495
954 - 459 = 495
Length until repeat: 3
Length of chain: 1
2. Choice Numbers

“There are choice cuts of beef, choice art, cars that are (at least according to Ferris Bueller) choice, so why are there no choice numbers?” rants Larry Coro, a rather reversed, somewhat deranged mathematician.

This is typical for the Plaza, where people holding very strange views tend to gather around lunchtime. Perhaps they gather for the food, or perhaps the food is what shoves their minds off that delicate balance called sanity. You don’t know; you don’t care. What you care about is getting this crazy guy out of your face.

“So, I’ve come up with a neat way to put numbers on equal footing with beef,” he says as you try your patented dodge-the-crazy-guy move. It doesn’t work.

“All I need is a little computer help to find out just which ones are choice...”

Problem Statement

Write a program to determine which numbers of a given set are considered choice and which are not.

A number is considered choice if one of its factors is also the number of factors of the number. For instance, the number 24 has 8 factors: 1, 2, 3, 4, 6, 8, 12, 24. Eight is itself one of those factors, so 24 is considered choice. Eleven is not choice, as 2, the number of factors, is not a factor of 11.

A number’s factors are all the positive integers which evenly divide the number, giving no remainder.

Notes

The input will consist of a sequence of positive integers less than 32,767 and a terminal 0. Obviously, the last 0 should not be processed. There will be at most 10 numbers given, not including the zero.

The output must give all the choice numbers in one block and all the non-choice numbers in another. Order is not important.

Examples

Example 1:
Enter numbers to test: 20 21 24 11 2 0
Choice numbers: 24, 2
Non-choice numbers: 20, 21, 11

Example 2:
Enter numbers to test: 1946 35 200 0
No choice numbers were given.
Non-choice numbers: 1946, 35, 200
3. Well, Golygons

Gridsylvania is a small country with a crazy royal family. Each generation passes new decrees that make very little sense. Once, around 100 years ago, King Mesh the Magnificent decided that all cities must have their streets laid out in a regular grid. The work on his project has been plagued with scandal, such as the Great Grid Graft, but is now complete, giving Gridonia some of the most convoluted architecture in the world.

King Ned the Repairman has noticed the general state of disrepair in some of the older construction. He has taken a personal interest in micromanaging the repairs, moving repair teams from intersection to intersection fixing everything from pot holes to fallen bridges.

The oldest city, Latticia, is home to a yearly conference for geometricians from around the globe. This is obviously a big event for such a small country. Being the oldest city, Latticia is also the center of the most repair work.

The geometricians who come to Latticia play an odd game. They walk for a block, turn at a right angle, walk for two blocks, turn again, and continue until they either return to their starting point or know they cannot. There is great prestige in being the geometrician who is the most unerring walker of golygons, as these figures are called.

The current repairs make this job a little more difficult. No golygon walker will dare cross an intersection where the King may be working. The King can draft any passer-by into any repair project. This would be very bad for the geometricians, who would doubtless crumble under the physical strain of real work.

A little travel agency (well, it can hardly be a big one under such circumstances) has decided to capitalize on the golygon competitions. They want to make maps showing every possible golygon and sell these to the geometricians.

The travel agency desperately wants to publish maps showing all the possible golygons, but the King moves the repair crews before the agency brains can finish the maps. You just happen to be passing through, trying to find some money to get home after an overly expensive spring break...

Problem Statement

Write a program that constructs all possible golygons for a city. Each city will have at least one possible golygon.

A golygon is a possibly self-intersecting figure formed by increasing, integer-length edges perpendicular to each other. An edge must be one unit longer than and at a right angle to the “previous” edge. The first edge must be of unit length.

No golygon may pass through an intersection under repair.

Notes

The input parameters for a single city are:

- the length of the longest edge of the golygon,
- the number of blocked intersections,
- and the coordinates of the blocked intersections.
The traveller starts (and ends) at (0,0), the center of the city. The coordinates of the blocked intersections are relative to the center of the city.

Your output consists of a series of golygons, one per line. Each golygon consists of a list of directions (N,S,W, or E) separated by a comma which indicate the path of the traveler, starting from the beginning of the expedition. Following the list of golygons, output the number of golygons found for the city.

A diagram of the city used in the example, along with the possible golygon, is given in figure 1.

![Diagram of Example City](image)

Figure 1: Diagram of Example City (not required in output)

Only solvable cities will be tested.

## Examples

**Example 1:**

Enter length of longest edge: **8**
Enter number of blocked intersections: **2**
Enter blocked intersection: **-2 0**
Enter blocked intersection: **6 -2**

Sample Output:

W,S,E,N,E,N,W,S

Found 1 golygon(s).
4. Who hit whom?

Lit E. Gator is a prominent attorney who specializes in bar brawls. In order to maximize his profit, he wants to blame as many people as possible for hitting his client. Thus, he also has to find who was hit the most and by whom. He tries to convince courts that bar brawls are like runaway reactions; the blame should follow the entire chain of who hit whom.

Unfortunately (?), Mr. Gator has a very small brain. He loses track of who hit whom very quickly. Being a prominent attorney, he is rich. Being a computer science student, you are poor. There is an opportunity here...

Problem Statement

Write a program to find who ultimately hit whom. Person A is defined as hitting person B if either A hit B directly or someone A hit directly hit B.

Input will be the number of people involved along with pairs of integers indicating who hit whom. Data will be ended with a pair of zeros.

Notes

There are at most 20 people involved, and the numbering begins with 1.

Incorrect, out of range data may be given and should be acknowledged with a warning and then ignored. The examples are elaborate for explanation’s sake. Your program does not need to specify the error so precisely.

Examples

Example 1:
Number of people in the scuffle? 4

Who hit whom? 1 2
Who hit whom? 2 3
Who hit whom? 4 3
Who hit whom? 0 0

Person 1 was not hit.
Person 2 was hit by person 1.
Person 3 was hit by person 1, person 2, and person 4.
Person 4 was not hit.

Example 2:
Number of people in the scuffle? 3

Who hit whom? 1 3
Who hit whom? 4 2
There are only 3 people involved, person 4 is innocent.
Who hit whom? 2 4
There are only 3 people involved, person 4 is innocent.
Who hit whom? 5 6
There are only 3 people involved, person 5 and person 6 are innocent.
Who hit whom? 3 2
Who hit whom? 0 0

Person 1 was not hit.
Person 2 was hit by person 1 and person 3.
Person 3 was hit by person 1.

The teacher’s going over arrays again. And again. Ooohh, now it’s getting exciting. Someone’s confused about the difference between arrays starting at zero and those starting at one.
You’re bored. Very, very bored. You start absentmindedly dropping your pencil. Sometimes it crosses the edge of the floor tiles, sometimes it doesn’t. You’re so incredibly bored that you’re actually interested in this. The teacher can’t possibly get upset with you if you start programming...

Problem Statement

Write a program that estimates the probability that a dropped pencil will cross one of a pair of parallel lines. The pencil will be dropped so that the midpoint lands between the lines.

The pencil length, \( l \), will always be less than the distance between the lines, \( D \). The basic setup is shown in figure 2.

![Figure 2: A dropped pencil between two lines](image)

Notes

Only an approximation is necessary. You may be asked to repeat an experiment to get a feeling for where the number is.

Examples

Example 1:
Enter the length and the distance: 11
The probability that the needle crosses a line is 0.637.
6. Repeating Expansions

Ed Sessive scribbles wildly in his notebook. He's just realized that every rational number's decimal expansion must repeat. He's also just realized that his pen is almost out of ink. He doesn't want to stop playing with his new discovery. And there's someone with a computer sitting just a few feet away.

The first thing you notice is a loud rustling of paper. You worry, knowing Ed's tendency to become obsessive. You look over and see him walking towards you, his hands stretched towards your laptop. You state firmly, "Um, no." He looks as if you've stabbed him with a nasty knife and he's going to die very slowly and miserably at your feet. You give up and ask what he wants. It sounds simple enough...

**Problem Statement**

Write a program which takes a numerator and a denominator, both positive integers less than 32000, and prints the repeating group of the decimal expansion.

**Notes**

The repeating digits displayed must begin with the first digit that appears after any non-repeating portion of the expansion. For instance, $\frac{1}{11} = .09090909\ldots$, so 0909 is displayed.

Each fraction will repeat within the first 100 digits of its expansion.

And no, you don't have to have the pluralization correct.

**Examples**

Example 1:
Enter a numerator and a denominator: 15
Repeating digit: 0

Example 2:
Enter a numerator and a denominator: 12 13
Repeating digits: 923076

Example 3:
Enter a numerator and a denominator: 11 42
Repeating digits: 619047
7. Whoa, big!

There are lots of big numbers all around us. For example, there are around 602,204,500,000,000,000,000,000,000 molecules in a mere eighteen grams of water. The Andromeda galaxy, the nearest to our own, lies about 1,470,000,000,000,000,000 miles away.

Unfortunately, your bank balance does not rank among these large numbers. Every little job, even figuring out something silly like how many feet deep Florida would be in silver dollars, can help. So when a group of scientists approach you about exactly that, you happily agree...

Problem Statement

Write a program to illustrate the magnitude of a number $N$, where $N$ is a large real number given in scientific notation. Calculate, to the nearest integer, how many feet deep a pile of $N$ silver dollars covering the state of Florida would be.

Figure 3: Silver dollars are stacked in a regular grid.

Assume that Florida has 58,560 square miles of land, while a silver dollar is 1\(\frac{1}{2}\) inches in diameter and \(\frac{3}{32}\) inch in thickness. Also assume that the silver dollars will be stacked neatly in columns placed edge-to-edge (see figure 3).

Notes

Numbers will be given in scientific notation with $E$ denoting the exponent. The input 1.54E21 would be the number $1.54 \times 10^{21}$.

The example’s number is intentionally incorrect.

You can assume that the number of feet deep is less than 30,000 and greater than zero.

Fractional values of .5 are rounded up.

Examples

Example 1:
Enter a huge number: **1.54E21**
Florida would be piled be 13 feet deep in 1.5400e+21 silver dollars.
8. Tree Surgeon

Gator Tree Surgeons isn’t your typical landscaping company. They don’t trim trees they put them together. Also they don’t work on typical trees. Instead they specialize in the very rare *ficus binarius*, (a binary tree to us laypeople). One of the most common jobs for Gator Tree Surgeons (GTS) is to repair trees that have been damaged in accidents. To do this GTS uses old records of the tree’s structure to help re-create the correct tree. Here’s how they do it.

Problem Statement

There are three common listings of binary trees. These are the preorder (node-left-right), inorder (left-node-right), and postorder (left-right-node) traversals. Each of these traversals represents a simple recursive algorithm for visiting each node of a binary tree, and (in this case) printing it. Any binary tree can be uniquely identified by any 2 of these traversals. Therefore, from knowing the preorder and the inorder you can reconstruct the tree. The same is true for the inorder and postorder, or preorder and postorder.

Input will be the inorder and postorder representations of the tree. These will be strings of characters, with each character representing a node in the tree.

Notes

Realize that the nodes may not be in alphabetical order. There are many possible ways of ordering a binary tree, and this is only one of them.

It is not necessary to show the tree, but it may be helpful to see an example. The example represents the tree found in figure 4.

All node values will be unique. There will be at most 20 nodes.

![Example Tree](image)

Figure 4: Example Tree

Examples

Example 1:
Enter the postorder traversal: **GDLHIEBMJNKFC**
Enter the inorder traversal : **DGBHLEIACJMFKN**
The preorder traversal is : **ABDGEHLICFJMKN**
9. Words, words, words

Cathy Chino loves coffee and word searches and absolutely despises sleep. One night (or possibly morning), she goes to her favorite 24-hour coffee shop with a book of word search puzzles. As soon as she gets her quintuple cup of espresso, she sits down and starts working on the one shown below:

```
adaddonot
gliaredhi
xodrecord
egikauwao
siendwren
schemeyru
urcsramyt
sensezrgn
qdifftick
```

darkness scheme logic record dowry
inch ordo sussex gyre rear
...

In a haze induced by caffeine overdose and sleep deprivation, Cathy invents a new, easier type of word search (which makes perfect sense to her at the time). There are simply too many letters, so she decides that removing them could only simplify the puzzle.

Each time she finds a word, she takes all the letters out of the block. Soon, she finds that she cannot finish the puzzle. Some of the letters she needs have mysteriously disappeared! After looking all around the cafe for them, she remembers that she removed them from the puzzle. It suddenly appears interesting to find out just how many words she can remove before being blocked.

You, of course, are sitting at the next table with your trusty laptop, working diligently on some obscure little problem of your own. You need a break, so you offer to help, in exchange for a cup...

**Problem Statement**

Write a program which finds and removes strings from the block of characters above. The strings may appear forward, backward, up, down, or diagonally. Case does not matter, and the strings must not wrap around the block. When a string is removed, it is essentially replaced by blanks; the block remains a 9-by-9 block.

The input will consist of a sequence of words which are to be removed sequentially. The output will be either the first word which cannot be removed or a message indicating that all the words can be removed successfully.

**Notes**

The input strings will occur at most once in the block of characters, although they may not appear at all (hey, it’s late, you can come up with some pretty strange words).

At most six words will be given, all of which will fit within the block.
Examples

Example 1:
Enter words:  wren derail rear tick
Cannot remove "rear".
Example 2:
Enter words:  wren tick scheme
Successfully removed all words.