S.A. Computer Olympiad
Second Round Start Div. 2007
For all grades up to and including Grade 10

Q1. Tee

Prepared by Donald Cook

Task

Your task is to write a program that will accept an odd integer as input. It will then output a capital 'T' using the lower case 't' character as shown in the figure below. The length of the top bar and the height of the 'T' are the same and are given by the integer input. Assume that only positive odd numbers will be entered.

```
  ttttttt
     t
     t
     t
     t
     t
     t
```

Figure1.

Sample run

Assume you run a test program with the integer 5

Input

Enter size of Tee: 5

Output

```
 ttttt
   t
   t
   t
   t
```

Constraints

The largest integer will be 21.

Test your program with

a. 3
b. 7
c. 11

Q2. Triangles

Proposed by Donald Cook

Prepared by Carl Hultquist

Task

Your task is to write a program that will accept three integers, which are the lengths of the sides of a valid triangle. Your program must then output what kind of triangle has been described, as follows:

- If all the sides are of different lengths, then this is a "scalene" triangle and your program should output SCALENE
- If two of the sides have the same length (and the third side has a different length), then this is an "isosceles" triangle and your program should output ISOSCELES
- If all three sides have the same length, then this is an "equilateral" triangle and your program should output EQUILATERAL

Constraints

- The largest value for the side of a triangle is 20.

Sample run

Input

```
3 4 5
```

Output

```
SCALENE
```

Test your program with

a. 3 4 5
b. 3 7 8
c. 6 6 6
d. 11 6 11
e. 8 8 6
f. 2 4 4

S.A. Computer Olympiad
Second Round Start Div. 2006

Q3. Digit Sums

Proposed by Donald Cook

Prepared by Max Rabkin

Description

A simple operation that you can perform on a number is to sum its digits. If the result has more than 1 digit, the process may be repeated until a single digit answer is given. For example, applying the operation to 673 we get 7 i.e. 6 + 7 + 3 = 16 and 1 + 6 = 7.

Task

Your task is to write a program to calculate the digit sums described. The will consist of a number of lines, each containing a single positive number less than 100,000. The last line of input will contain 0 – this line should not be processed. The output will consist of a single line containing the digit sum (a single digit number) for each line of input, with no spaces separating them.

Sample run

Note the input is terminated by 0.

Input

673
51
1000
99
0

Output

7619

Test your program with

a.
26
36
37
99
0

b.
11111
24567
91919
37459
0

c.
1
45
3645
99999
0

Q4. Dominoes

Proposed by Marco Gallotta

Prepared by Shen Tian

Description

There are N dominoes arranged in a circle. They will be referred to as domino 0 through to N-1. The i^{th} domino has the value a_i on it.

Normally if you were to knock over one of the dominoes, this would cause the rest of the dominoes to knock each other over one by one until all the dominoes have been knocked over. However, these dominoes do not act as normal dominoes. These are magic dominoes. When the i^{th} domino is knocked over, it tries to knock over domino $(i + a_i) \bmod n$ (where x mod y is equal to the remainder left when x is divided by y). However, if that domino has already been knocked over, the process ends and no more dominoes are knocked over. As a consequence, if we start with all the dominoes upright, we have to careful which domino we knock over. Only some dominoes, when knocked over, will start a chain reaction as described above that will knock over all the dominoes. We shall call these super dominoes.

Task

Write a program to determine, given N and all the $< a_i$ s, how many super dominoes there are.

Sample run

$N = 4$, $a_0 = 1$, $a_1 = 3$, $a_2 = 5$, $a_3 = 1$

When you knock over domino 2, it knocks over domino $2+5 \bmod 4 = 3$. $3+5 \bmod 4 = 0$, so domino 0 is knocked over. $0+1 \bmod 4 = 1$, thus domino 1, the last domino standing is knocked over. Domino 2 is a super domino.

If you knock over domino 1 however, domino $1+3 \bmod 4 = 0$ is knocked over. Domino 0 tries to knock over domino $0+1 \bmod 4 = 1$, which has already been knocked over. The process ends with dominoes 2 and 3 still standing.

One checks dominoes 0 and 3 similarly. Domino 2 is the only super domino in this case.

Input

A single integer N, followed by N integers for values of a_0 to a_{n-1}. Each integer has its own line.

Sample input:

4
1
3
5
1

Output

Output the number of super dominoes

Sample output:

1

Test your program with

a. 5
1
1
1
1
1

b. 2
7
5

c. 3
54
1
1

d. 5
8
4
12
4
7

Q5. Hidden Strings

Proposed by Bruce Merry

Prepared by Ben Steenhuisen & Charles Bradshaw

Description

Strings are just a series of characters 'strung' or joined together. Substrings are strings that are, in fact, just a part of a larger string. One might, for various reasons, wish to find if a string is merely a substring of another string, sometimes disregarding such things as case (UPPER and lower) or punctuation.

Task

Your task is to write a program that finds and prints all occurrences of a word (substring) within a piece of text. This word may be hidden, it may contain spaces or punctuation, and it might appear with different capitalization. The program must accept 2 strings, the first being the main string, and the second the substring that is to be searched for in the main string. If no substrings are found, 'No strings found' must be printed.

Constraints

The length of each string will be <255 characters.

Sample run

Now that we are left with only a single letter, we are done.

Input

It's behind the intercom. Put erasers to one side computer

Output

```
com. Put er
```

Test your program with

a. This suit is black!!
not

b. "You thought your secrets were safe. You were wrong."-Hackers
gh

c. Donald likes Mall shops where he and his friends discuss idealism all day long.
small