ECOO 1999
Programming Contest

Boardwide
Contest

Problem 1 – A different Kind of Writing

Canadians are used to reading text from left to right, from the top line to the bottom. Some cultures write their lines from right to left, or from up to down.
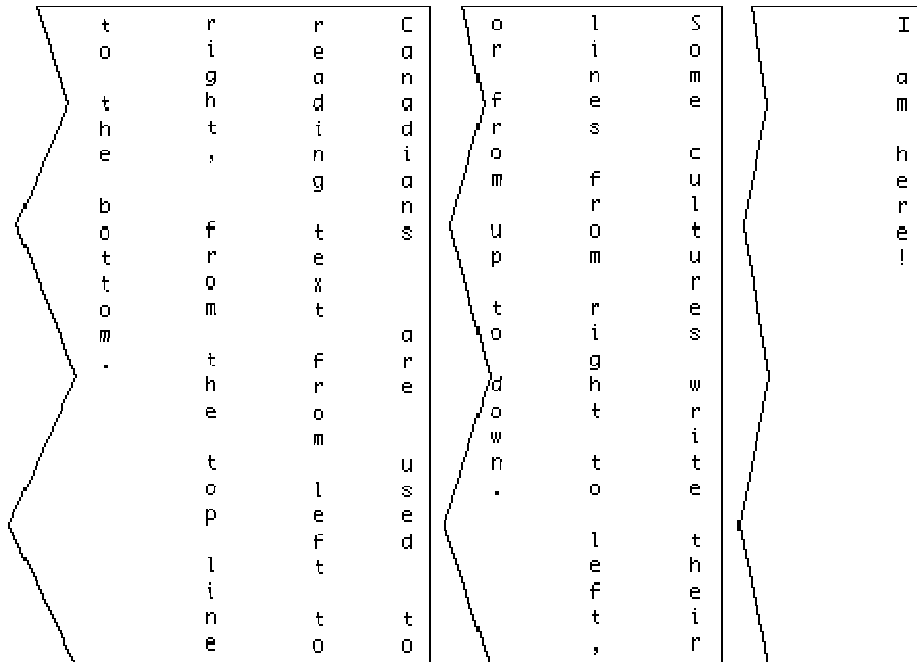
Your task is to read text from an input file and display it on the 80x25 screen in vertical lines that read from top to bottom, lines reading from right to left, as in the display to the right. Display the first line into column 79, the second line into column 74, the third into column 69, the fourth in column 64 and so on. Display each line except for the last fully justified: That is, row 1 and row 25 must always contain letters and not spaces. Spaces must me equally divided between words: The differences between spaces between words in the same line must never be more than 1: If for example the largest number of spaces between two words is 3, then all spaces in that line should be at least 2 (see for example the first line of the first sample below).

The input file (DATA11.txt for the first try and DATA12.txt for the second) contains 5 lines of no more than 250 alphanumeric characters. Display each of the 5 lines in the way described above, one set of data at a time. Each new data set should be displayed after one presses any key to proceed.

SAMPLE INPUT: (3 of 5 lines only)

Canadians are used to reading text from left to right, from the top line to the bottom.
Some cultures write their lines from right to left, or from up to down.
I am here!

SAMPLE OUTPUT:

```
    t       r       r       C       o       l       S               I
    o       i       e       a       r       i       o
    g       a       n       n       m       a
  t h       d       a       f       e       e                       m
  h t       i       d       r       s
  e ,       n       i       o       f       c                       h
          g       a       m       r       u                       e
  b f       t       n       u       o       l                       r
  o r       e       s       p       m       t                       e
  t o       x               r       u                       !
  t m       t       a       t       r       e
  o         a       r       o       i       s
  m . t     f       e       d       g       w
    h       r           o       h       r
    e       o       m       w       t       i
    t       u       n       t       t
  t         l       s       .       o       e
  o         e       d           l       t
  p         f               l       h
  l         t       t       e       e       e
  i       o           o       f       i
  n       t       t       o       t       r
  e       o       o       o       ,
```

Problem 2 – The Fibonacci Shuffle

Consider a deck of 52 cards: From top to bottom, 13 Spades, 13 Hearts, 13 Diamonds, 13 Clubs. Each set of 13 cards are in order: Ace, King, Queen, Jack, 10,9,8,7,6,5,4,3,2. Represent each of the 13 cards by a one-character symbol: A,K,Q,J,T,9,8,7,6,5,4,3,2 The cards occur in order in a brand new deck: on top is the Ace of Spaces (number 1), followed by its King (#2), and so on: The two of Spades is card #13, the Ace of Hearts is #14, and finally the two of Clubs is card #52.

An automatic shuffling machine will do a Fibonacci shuffle* and then deal the four players, North, East, South and West each 13 cards the usual way: North is dealt the 1st, 5th, 9th and so on cards from the shuffled deck; East is dealt the 2nd, 6th, 10th etc. cards; South is dealt the 3rd, 7th, 11th etc, and West is dealt cards 4,8,12 and so on from the suffled deck.

A Fibonacci shuffle of degree n is a shuffle where the first n Fibonacci numbers are used, to split a deck of cards repeatedly into two stacks, from which a third stack is constructed.

Consider the Fibonacci numbers mod 52 (where each number is formed by adding the previous two numbers: 1,1,2,3,5,8,13,21,34,3,37,40,...) Each time a stack is split, the top set of x cards move to the left and the last set of 52-x cards to the right. A third stack is constructed by taking in succession one card off of the left and right stacks and placing them on the third. for example:
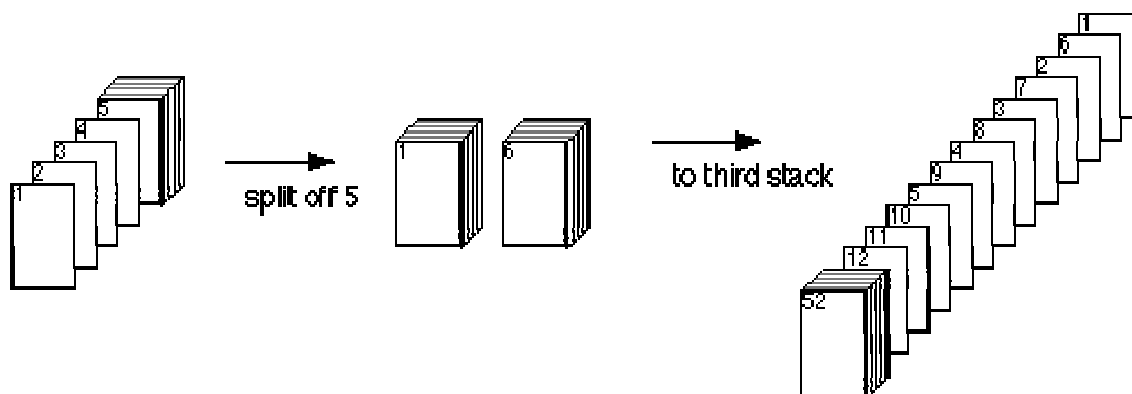
In a Fibonacci shuffle of degree 5, this procedure is performed 5 times, in succession, by splitting off first 1, then 1, followed by 2, 3 and 5. In a Fibonacci shuffle of degree 10 this procedure is performed 10 times, using the numbers 1,1,2,3,5,8,13,21,34,3.

The input file (DATA21.txt for the first try and DATA22.txt for the second) contains 5 numbers, representing the degree of the Fibonacci shuffle for 5 different card games.

The output will be 5 different screens displaying the hands of North, East, South and West as seen in the example on the back of this sheet. Each new screen will be displayed after pressing any key.

SAMPLE INPUT: (2 of 5 numbers only)

9
3

SAMPLE OUTPUT:

```
 North      Spades
            Hearts
            Diamonds  4 3 2
            Clubs     A K Q J T 6 5 4 3 2

 East       Spades
            Hearts    A K Q J T 7 6 5 4
            Diamonds  9 8 7 6
            Clubs

 South      Spades    3 2
            Hearts    3 2
            Diamonds  A K Q J T 5
            Clubs     9 8 7

 West       Spades    A K Q J T 9 8 7 6 5 4
            Hearts    9 8
            Diamonds
            Clubs
                                        press any key to continue..
 North      Spades    J 7 3
            Hearts    Q 8 4
            Diamonds  K 9 5
            Clubs     A T 6 2

 East       Spades    K 8 4
            Hearts    K 9 5
            Diamonds  A T 6 2
            Clubs     J 7 3

 South      Spades    Q 9 5
            Hearts    A T 6 2
            Diamonds  J 7 3
            Clubs     Q 8 4

 West       Spades    A T 6 2
            Hearts    J 7 3
            Diamonds  Q 8 4
            Clubs     K 9 5
                                        press any key to continue..
```

Problem 3 – Sifter

Imagine a sifter for grain, like the one pictured below. If 64 grains are poured through the top, then equal amounts (32) will fall on either side of the triangular obstruction on the first level, each of which will further split at the next level, and so on. With powers of 2 (that can always be split in two, such as 64) the numbers behave like in a Pascal triangle. Odd numbers (2n+1) will however be split so that the larger (n+1) will pass on the right and the smaller (n) will pass on the left.
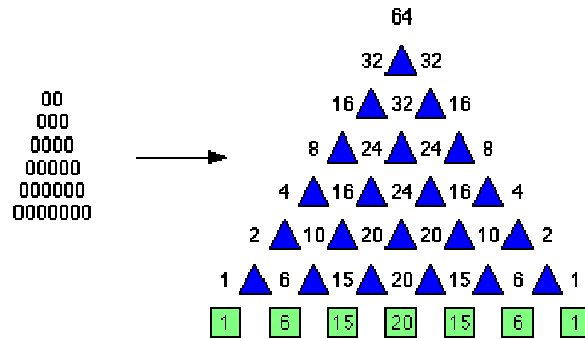
If some of the openings are closed, as in the example below, the grains will behave in a predictable way:
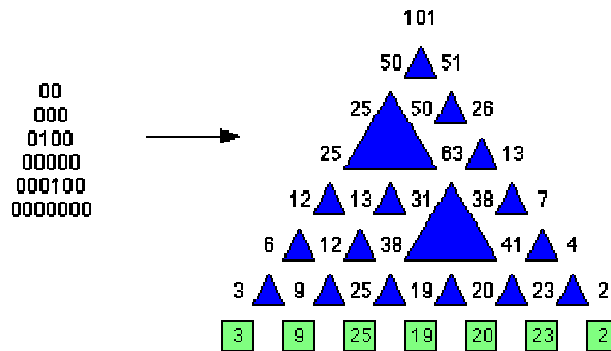
Note that a sifter is defined in the input file, as a set of zeros and ones. When two openings in a row are closed, the effect is equivalent to the situation, where the opening **above** the two is closed as well (see the next diagram)
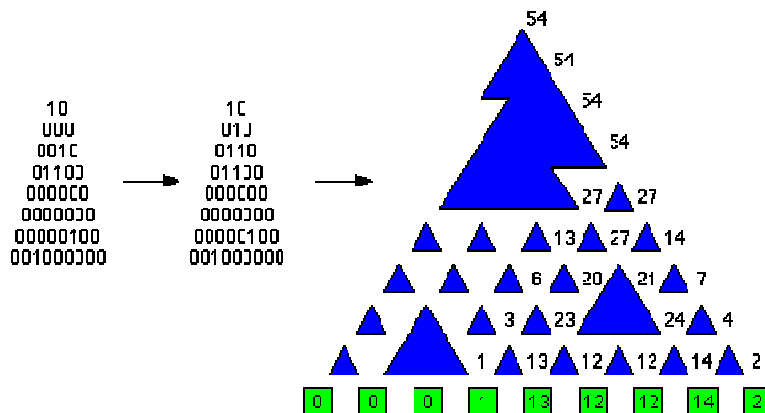
Note that a sifter of n levels has n+1 receptacles at the bottom

Your task is to write a program that will read from an input file data about the number of grains poured through the sifter, and the properties of the sifter itself. You will then print the contents of the receptacles at the bottom of the sifter. Note: In all sifters, all grain will reach the receptacles.

The input file (DATA31.txt for the first try and DATA32.txt for the second) contains 5 sets of data: Each set of data is composed of two numbers on separate lines: the number of grains followed by the number of levels of the sifter (n). Then follows, in text form, the n lines of 0's and 1's (without spaces)
The set of data will contain no blank lines.

SAMPLE INPUT: (2 of 5 sets)

```
101
6
00
000
0100
00000
000100
0000000
54
8
10
000
0010
01100
000000
0000000
00000100
001000000
```

SAMPLE OUTPUT:

```
3    9    25   19   20   23    2

0    0    0    1   13   12   12   14    2
```
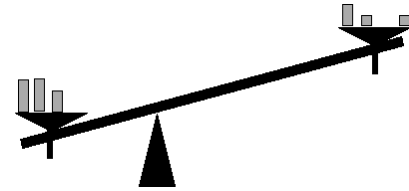
Problem 4 – Prime Number Madness

Consider equations of the form $a_1p_1 + a_2p_2 + ... + a_np_n = z$ where $p_1, p_2, ..., p_n$ are consecutive prime numbers, and $a_1, a_2, a_3, ... , a_n$ are any integer, 0 or greater.

Suppose you are given three positive integers, x,y,z . First find all prime numbers between x and y (inclusive). Then find all possible equations of the form listed above.

With this in mind, write a program that will read from a data file (DATA41.txt for the first try, DATA42.txt for the second) five sets of 3 positive integers x,y,z where $1 < x < y < z < 2^{15}$ and that lists on a cleared screen all solutions to the equation of the form listed above.

The output may scroll, if there are more lines of output than can be displayed on one screen. Pause while each of the five sets of data is displayed, proceeding only when any key is pressed.

SAMPLE INPUT:

```
7 16 100
34 48 300
3 10 20
97 120 600
1230 1260 10000
```

SAMPLE OUTPUT:

```
0*7 + 2*11 + 6*13 = 100
3*7 + 6*11 + 1*13 = 100
4*7 + 3*11 + 3*13 = 100
5*7 + 0*11 + 5*13 = 100
8*7 + 4*11 + 0*13 = 100
9*7 + 1*11 + 2*13 = 100          press any key to continue


7*37 + 1*41 + 0*43 + 0*47= 300
                         press any key to continue


0*3 + 4*5 + 0*7 = 20
1*3 + 2*5 + 1*7 = 20
2*3 + 0*5 + 2*7 = 20
5*3 + 1*5 + 0*7 = 20
                         press any key to continue

2*97 + 3*101 + 1*103 + 0*107 + 0*109 + 0*113 = 600
3*97 + 0*101 + 3*103 + 0*107 + 0*109 + 0*113 = 600
3*97 + 2*101 + 0*103 + 1*107 + 0*109 + 0*113 = 600
4*97 + 0*101 + 1*103 + 0*107 + 1*109 + 0*113 = 600
                          press any key to continue

0*1231 + 1*1237 + 5*1249 + 2*1259 = 10000
1*1231 + 2*1237 + 0*1249 + 5*1259 = 10000
                        press any key to continue
```

ECOO 2009
Programming Contest

Regional
Contest

to be written
April 25th

Problem 1: Weigh Scale

Namhu, a merchant in ancient Mesopotamia had a curious
weigh scale. One arm is twice as long as the other, with
the effect that if you placed one shekel on the longer arm,
you had to place two shekels on the shorter arm to keep
the scale in balance. Namhu had a limited amount of
weights: one half shekel, one shekel, a single 5-shekel weight and a single 10-shekel weight.
Luckily he had an unlimited number of 25-shekel weight.
If a customer wanted to buy 72 shekels of wheat, he placed seven 25-shekels on the short
arm and the 10-shekel, the 5-shekel and the half-shekel on the other side with the bag to be
filled with wheat.
That is because 175 shekels on the short side is equivalent to 87.5 shekels on the long side.
So to balance the 72 shekel wheat container, he needed 15.5 shekels on the long side.
(Namhu supplied the bag, for which he charged the same as the weight of the wheat.)

Namhu always placed the bag of wheat on the long side. Write a program to assist Namhu,
with the correct distribution of weights, given any value of weight of the wheat. If there is no
solution for a given weight, say so. If there is more than one solution, any one correct
solution will be accepted.

Data**11**.txt (Data**12**.txt for the second try) contains 5 integers on 5 separate lines
representing 5 different weights as in the sample below. Your output should be similar to
the output of the sample. All input data are between 1 and 999.

Sample Input
```
10
11
34
73
879
```

Sample Output
```
For a weight of 10:
short end: -- 2x25 --
long end: -- 10 -- 5 --

For a weight of 11:
short end: -- 1x25 --
long end: -- 1 -- 1/2 --

For a weight of 34:
short end: -- 4x25 --
long end: -- 10 -- 5 -- 1 --

For a weight of 73:
short end: -- 7x25 -- 1 --
long end: -- 10 -- 5 --

For a weight of 879:
short end: -- 71x25 -- 5 --
long end: -- 10 -- 1 -
```

Problem 2: Ramanujan Cubes

The mathematician Srinivasa Ramanujan, no doubt inspired by the Pythagorean equation $C^2 = A^2 + B^2$, examined the equation $X^3 = A^3 + B^3 + C^3$ and developed several interesting identities based on it.

6 is the smallest positive integer such that its cube is the sum of 3 other cubes of integers: $6^3=5^3+4^3+3^3$.

It is interesting to note that most numbers have this property.

It is your task to find the number of integers, x, that have the property, that it can be written as the sum of 3 positive integral cubes, in the range between two given integers a and b, including the limits a and b.

Data21.txt (Data22.txt for the second try) contains 10 lines: 5 pairs on lines, each containing a positive integer. The first integer of the pair is a, the lower limit of the range and the second integer is b, the upper limit of the pair. All input data are between 1 and 999.

Sample Input:
```
1 34
56 77
100 200
150 600
900 999
```

Sample Output:
```
There are 12 numbers between 1 and 34 whose cubes are Ramanujan Cubes
There are 13 numbers between 56 and 77 whose cubes are Ramanujan Cubes
There are 73 numbers between 100 and 200 whose cubes are Ramanujan Cubes
There are 367 numbers between 150 and 600 whose cubes are Ramanujan Cubes
There are 90 numbers between 900 and 999 whose cubes are Ramanujan Cubes
```

Problem 3: Power Cipher

A certain cipher uses the number 29 as a modulus. Letters are converted into numbers, then manipulated and turned back into letters. For this reason, we need 29 characters. Besides the 26 regular capital letters, we need three extra characters: * representing the space character; + representing the word (NOT letter) "A"; and % representing the word "the".

After a message has been capitalized, and any non-letter characters ignored, each letter is replaced by a number: A=0, B=1, etc. Z=25; *=26, +=27 and finally %=28. A secret number ,x, is picked and the first "letter" increased by: x^1, the second letter by x^2, and so on, and the nth letter by the number x^n. All this of course, mod 29. Since all results are numbers between 0 and 28, the numbers are next replaced by their equivalent characters.

Finally, a secret code word is embedded in an arbitrary place inside the message, followed by the letter corresponding to "x". Then each letter is increased by an arbitrary number, so that the secret code word (and "x") is no longer visible.

Let's follow the steps of "A SERPENT RAN INTO HIS BOOT" using the secret code "SECRET"
+*SERPENT*RAN*INTO*HIS*BOOT
27, 26, 18, 4, 17, 15, 4, 13, 19, 26, 17, 0, 13, 26, 8, 13, 19, 14, 26, 7, 8, 18, 26, 1, 14, 14, 19
Let's use x=5, then each of these numbers are changed by the addition of the following:
5, 25, 9, 16, 22, 23, 28, 24, 4, 20, 13, 7, 6, 1, 5, 25, 9, 16, 22, 23, 28, 24, 4, 20, 13, 7, 6 to get:
3, 22, 27, 20, 10, 9, 3, 8, 23, 17, 1, 7, 19, 27, 13, 9, 28, 1, 19, 1, 7, 13, 1, 21, 27, 21, 25
DW+UKJDIXRBHT+NJ%BTBHNBV+VZ

Inserting "SECRET" followed by 5=F: DW+UKJDIXRSECRETFBHT+NJ%BTBHNBV+VZ
Increase each letter by the same arbitrary number (say 7):
KAF+RQKPBYZLJYL*MIO*FUQGI*IOUI%F%D

Data**3**1.txt (Data**3**2.txt for the second try) contains 5 sets of data: Each set takes up two lines. The first line contains a code word and the second line the encoded message, containing fewer than 250 characters. Write a program that will decode each message as in the sample below.

Sample Input:
```
SONG
CDEQZ+XTSLMUZ+EZOBYJZNZGMJJIAMSXOCD**SGRDPE
SECRET
TC+MSZTFYGZBXMADSQCSEVS%W+RXKCKWPA
MOON
YMYTUDAVMTNO+AA%XHLKJUFRPBVNTS+Q
CREATURE
MRJWTIYA**NM+OKAB+OWPG%YBHXII+UGRA%
MORE
+JMXRCEHX+YVSI+ZL+ZLD*YVT%VMFXVI%QWOIIWL
```

Sample Output:
```
THERE ONCE WAS AN OLD MAN WITH A FLUTE
A SERPENT RAN INTO HIS BOOT
BUT HE PLAYED DAY AND NIGHT
TILL THE SERPENT TOOK FLIGHT
AND AVOIDED THAT MAN WITH HIS FLUTE
```

Problem 4: Checkers

Checkers is a game played on an 8x8 board. Checkers may only move diagonally, and when an opponent's checker piece may be taken by jumping across the piece diagonally as is shown. This may only happen, if the finishing spot of the jump is empty, as shown. The diagram shows two different kinds of moves: On the left, one black checker is taken. On the right, two black checkers are taken, because after finishing the first jump, the white piece is in position to take another one. Checkers may only move and jump diagonally forwards, as shown. For the black checkers, forward is down and for the white checkers, forward is up.
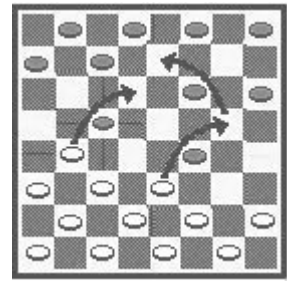
Data41.txt (Data42.txt for the second try) contains five sets of 8 lines of 8 characters. Each set represents a checker board containing checkers. White checkers are represented by W, Black checkers are represented by B. Squares that contain no checkers are represented by a *, as shown in the sample below.

Write a program that will find the largest number of checkers that can be taken either by black or by white, and that identifies the mover by its position: bottom left is 11, which is short for (1,1); bottom right is 18 which is short for (8,1) and top right is 88, shorthand for (8,8). If there is a tie for largest, then any one of these tied solutions will be correct. You may assume that there will always be at least one valid jump on the board.

Sample Input *(note: the data file contains this data in 40 lines of 8 characters each)*

```
*B*B***B      *B*B****      *B*B****      *B*B***B      *B*B****
**W***B*      **W***B*      **W*****      **B*B*B*      **W*B*B*
********      ********      *****B**      ***B*B**      *****B**
**W*B***      **W*B***      **W***W*      **W*B***      **W*B***
***W****      *B*W****      *B*B****      ***W***W      *B*W*W**
**W***B*      **W***B*      ******B*      **W***W*      **W***B*
*****W**      *****W**      *B***W**      *****W**      *****W**
W*W*W*W*      W*W*W*W*      W*W*W*W*      W*W*W*W*      W*W*W*W*
```

Sample Output
```
Black 28 takes 3 checkers
White 44 takes 2 checkers
White 11 takes 3 checkers
Black 46 takes 2 checkers
White 64 takes 2 checkers
```

ECOO 2009

Programming Contest

Final

Contest

May 9th

Problem 1 - Friendly Zigzag

To the left is a zigzag puzzle of three words as in the diagram. The first and last words are given, the one in-between word must be found.

Imagine that Renata, Aaron's girlfriend, because of some silly disagreement, wants to be as far as possible from Adam in the Zigzag. You must therefore find one person to sit between them. It is not sufficient to just pick any person that fits the zigzag, you must pick the person who will separate the two as far as possible. The result must be that the first letters of each of the two friends (e.g. R for Renata and A for Aaron) are as far removed as possible. In this example, the zigzag *itself* is formed by the letters: **R**ENATADRIANORA**A**. Note then that the A of AARON is 13 spaces away from the R of RENATA.

Data11.txt (Data12.txt for the second try) contains an integer, x, representing the number of names that follow. Then following x lines containing the x names in capital letters. The remaining 5 lines contain two integers each, between 1 and x, representing the positions of the names of the friends in the list on either end of 5 zigzags. Note that x<100, and all names contain less than 15 letters and no spaces or other special characters.

Write a program that creates zigzags according the specifications above, and state the distance between the two names as shown in the sample. You may use any name in the list provided no name is repeated more than once in any zigzag, except when the same name occurs more than once in the list. There may be several equivalent solutions to the problem, however, the maximum distance is of course unique and there is always at least one solution. You may show each of the 5 solutions one at a time using some input control, or all at once.

Sample Input
(note, to save space these data are on several columns. The data file will contain just one column)

```
36          BERNADETTE  ELYSE     LARRY     RENATA     WHITNEY
AARON       DANA        EMMA      MARIA     ROCHELLE   WILLIAM
ADAM        DANIEL      ERIC      MATTHEW   RYAN       1 28
ADRIAN      DAVID       GIULIA    MICHAEL   SABRINA    23 15
ALEX        DEAN        JULIA     MICHAEL   SHANE      9 33
ALEXANDRA   DOMENIC     KAYLA     PETER     STEPHANIE  24 31
ANDREA      DOMENICO    LIANNA    RANY      STEPHEN    32 7
```

Sample Output *(outputs placed beside each other to save space)*

```
13             13              19                18                 18
RENATA          A               B                 A                  S
    D           L           DANIEL            MICHAEL                T
    R       MATTHEW             R                 E              SHANE
    I           X               N                 X                  P
    A           A               A                 A                  H
AARON           N               D                 N                  A
                D               E                 D                  N
                R               T                 R                  I
            EMMA                T             SABRINA      BERNADETTE
                            STEPHANIE
```

Problem 2 - Flying Monkeys

On a watery planet, far far away, there live flying monkeys and crocodiles. The monkeys live in the trees that rise out of the water, and the crocodiles patrol the seas. The monkeys are called flying monkeys, because when they jump from tree to tree they manage to sail across. The distance they can sail is exactly equal to the height of the tree they jumped from. Occasionally a monkey sails to a short tree, a tree too short to jump back, and the consequences are of course tragic for the monkey: Either it is stuck there forever, or it gets eaten when it attempts to get back. Generally the monkeys know that it is safe to jump to a tree that is no further away than the height of the launching off tree. Given troupes of monkeys inhabit a range of trees that are close together, and simply ignore the trees that are out of range. Likely those trees are inhabited by a different troupe of monkeys.
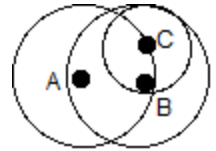
In the diagram there are three trees, A,B,C. The circles represent the distance a monkey can fly to get to the next tree. A monkey jumping from A to C cannot get back to A and would have been stuck there, if it was not for the fact that it can jump to B and from B back to A. This set of 3 trees are part of the same range.

Data21.txt (Data22.txt for the second try) contains five sets of data. The first line of each set contains an integer, k (k<20), denoting the number of trees in the set. Then follow the trees' data, each tree on a separate line. Those k lines contain 3 integers between 1 and 200 inclusive representing the x-value, the y-value and the height of a tree. These trees may or may not be part of the same range.

Write a program that finds, how many trees belong to the range that contains tree #1 in the set.

Sample Input
(note, to save space these data are on several columns. The data file will contain just one column)

5	14 89 56	87 45 65	65 86 45	30 61 29	11 49 51	30 55 28
73 67 65	8 88 68	12	38 75 67	92 69 50	8 73 29	20 57 64
68 75 29	73 65 56	94 70 20	24 75 54	1 92 27	99 54 48	90 50 30
78 81 34	18 43 62	3 91 51	55 13 61	87 32 24	18 96 58	
88 13 43	44 74 27	47 13 58	73 89 60	30 27 33	86 34 42	
32 8 33	1 1 40	48 13 49	49 8 44	45 25 55	50 54 59	
10	62 16 44	45 26 25	8	40 11 59	28 5 52	
15 96 25	57 45 39	90 82 34	43 94 47	11	5 63 24	

Sample Output
```
There are 3 of 5 trees in the range.
There are 4 of 10 trees in the range.
There are 7 of 12 trees in the range.
There are 1 of 8 trees in the range.
There are 11 of 11 trees in the range.
```

Problem 3 - Multiplication Cryptarithms

A cryptarithm is an equation made up of letters, where each letter represents a digit. For the sake of this question we will only be using only non-zero digits.

For example, "ECOO * IS = BETTER" is a cryptarithm for it represents the equation:
"5477 * 28 = 153356".
Note in particular that the "E" in the word "ECOO" has the same value as the "E" in the word "BETTER".

Data31.txt (Data32.txt for the second try) contains five lines, each line contains one multiplication cryptarithm as in the example below: Three words separated by "space star space" and "space equals space".

Write a program that will solve the five cryptarithms within 30 seconds. (If you are writing in Turing, you may have 60 seconds) If there are more than one solution, any solution is acceptable. If there are no solutions, state: "There are no solutions".

Sample input:
ECOO * IS = BETTER
TOM * THE = BITER
POSERS * R = AWESOME
RIDDLE * L = PROBLEM
TWO * TWO = SQUARE

Sample output:
ECOO * IS = BETTER
5477 * 28 = 153356

TOM * THE = BITER
143 * 176 = 25168

POSERS * R = AWESOME
913863 * 6 = 5483178

RIDDLE * L = PROBLEM
976624 * 2 = 1953248

TWO * TWO = SQUARE   or: TWO * TWO = SQUARE
567 * 567 = 321489   854 * 854 = 729316

Problem 4 - Crucisomma

Crucisomma is a number puzzle found in an Italian newspaper. It contains squares as shown in the example with six equations (three horizontal and three vertical) each containing three terms on the left of the equation and two operators. The missing terms are the digits 1,2,3,4,5,6,7,8,9 in some order. In the example, the right hand sides of the three horizontal equations are respectively: 16,7,-1; the right hand sides of the three vertical equations are respectively 18,1,0.

Data41.txt (Data42.txt for the second try) contains five sets of 2 lines. Each set represents a crucisomma puzzle. The first line contains the 12 operator symbols in the order in which they occur in the puzzle: from top left to bottom right, following each row from top to bottom. The second line contains the 6 right hand values, in order as shown in this example:

```
x−x++:+:−−x−
16 7 −1 18 1 0
```

*Note that **x** stands for multiplication and : for division.*

Write a program that will read the data, solve the puzzle and print out the 9 digits in a three by three square as shown in the sample solution. One important item worthy of note: Order of operations is ALWAYS the order in which they occur. For example: 5+1x6 =36. As well, intermediate answers, that is, the result of the first operation of the two is always an integer.

Sample Input
```
x−x++:+:−−x−
16 7 −1 18 1 0
+:−+−x−x:++−
1 1 3 30 1 14
+−−:x:+−x:+:
11 12 2 0 20 12
−xx:+−+:+−x:
5 11 6 14 15 10
x−xx:+−+−+:+
4 11 10 17 37 13
```

Sample Output *(solutions placed beside each other to save space)*
```
375     729     784     765     256
624     143     629     418     481
189     568     153     293     937
```