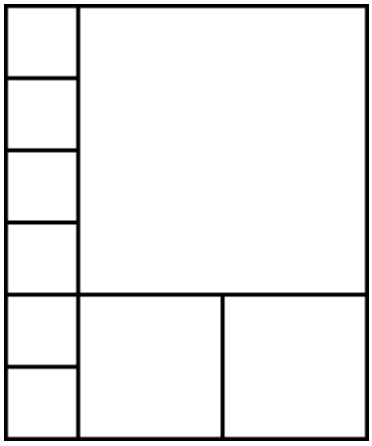
# ECOO 2009
## Programming Contest

## Board wide Contest

**to be written after March 20th**

# Problem 1: Power Tiles

You are given a rectangular floor that is to be paved with square tiles. The tiles all have special sizes: They have dimensions x foot by x foot, where x is a power of 2: 1, 2, 4, 8,…. A 5 foot x 6 foot rectangle could be paved with 30 tiles of dimension 1 x 1, however, we only need 9 tiles, if we use one 4x4, two 2x2 and 6 1x1 tiles (see the figure on the right)



Data11.txt (Data12.txt for the second try) contains five lines, each line contains a pair of positive integers representing the dimensions of the rectangle to be paved. Each dimension is an integer between 1 and 10,000.

Write a program that will find the minimum number of tiles needed to pave the floor, given its dimensions.

**Sample input:**
10 5
1000 1001
21 13
9999 888
345 1277

**Sample output:**
14 tiles are needed for a 10 by 5 floor
1358 tiles are needed for a 1000 by 1001 floor
42 tiles are needed for a 21 by 13 floor
4065 tiles are needed for a 9999 by 8888 floor
2046 tiles are needed for a 345 by 1277 floor

# Problem 2:Anagrams

Anagrams are the rearrangement of letters of a word or short sentence, so that it has a different meaning. For some examples, see the sample input below. Spaces and punctuation marks are naturally ignored, and so is the distinction between upper and lower case.

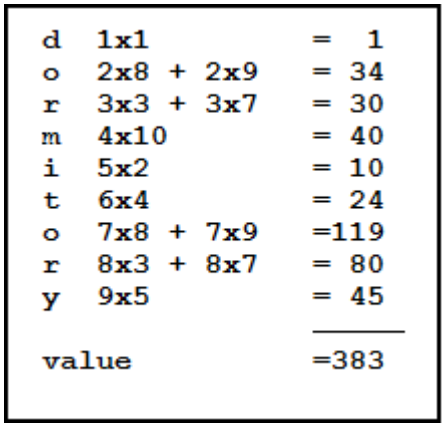
Anagrams may be assigned a value depending on the position of corresponding letters within the words: Take the example of : DORMITORY = DIRTY ROOM.
The first word contains 9 characters, and the values of the letters take on, according to their positions, 1, 2, 3, … , 9. Note that that the **O** and **R** have 2 different values. These same letters have different values again in the phrase: DIRTY ROOM. The letter **M** has in fact the value 10. By taking the product of the the corresponding positions, each letter has its own unique value. Since O may be found in two different places, the possible products will be added (see the figure (D=1, O=153, R=110, M=40, I=10, T=24, Y=45) and the value of the anagram is 383.

Data21.txt (Data22.txt for the second try) contains five lines, each line contains two phrases separated by the equals sign "=" surrounded by two spaces. Each phrase may contain extra punctuation marks or extra spaces, which are to be ignored in the criterion whether the two phrases are two parts of an anagram. However, they are not to be ignored in the criterion, of what the position is of each letter.

Write a program that will determine whether each line is a valid anagram, and that will determine the value of the anagram. If the line does not contain an anagram, your program should indicate this.

**Sample Input:**
Listen = Silent
Dormitory = Dirty Room
School Master = The Classroom
A telephone girl = Repeating "Hello"
Salmonella = so not a Meal

**Sample Output:**
Listen = Silent
is an anagram and its value is 84
Dormitory = Dirty Room
is an anagram and its value is 383
School Master = The Classroom
is an anagram and its value is 790
A telephone girl = Repeating "Hello"
is an anagram and its value is 1602
Salmonella = so not a Meal
is not an anagram



# Problem 3:Chemical Brew

Our chemical brew is a rectangle of characters as shown in the sample input. Individual compounds are made up of “atoms” (single letters). Atoms are bonded to each other to form a compound, if they are neighbours in the rectangle, above, below, to the left or right. Atoms that are only neighbours diagonally are not part of the same compound. The first rectangle in the sample contains therefore the following “molecules”:
7 HHO, 2 HH, and one of each: CCCCHHHHHHHHHSS, HHOO, OO, HHHOO, H, HHHHOO, CC

Write a program that will find among the various compounds in the rectangle the number of molecules of a given type.

Data31.txt (Data32.txt for the second try) contains five sets of data. The first line of each set contains two integers <=50, representing the length, x, and height, y, of each rectangle. The second line contains the compound for which you must search. The remaining y lines contain x characters each, spaces between compounds are represented by the character “*”.

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

```
20 16
OHH
*******HOH**********
*****************CC*
********************
****H******HH*******
H***O*******O*******
H***H***************
HH*******O****O*****
HH**HH***O***HOH****
CH*****HO*****H*****
CH******H****H******
SSCH*******OH*******
CH*********H****HH**
**H**O********HO****
**HO*HHO*******H****
****************OH**
****************HHHO
```

```
30 7
HCHHH
******HH*********H*********CCH
HHHHHH**OO*****HHC*********H**
**CC**HH******H**H***********
CH**HHC*******H************HOH
H*********OO**H*******OO**O***
H****HH******H*****HH****O**H
H*****C*******C****CHH*HOH***H
10 6
OO
OOO*CC**HH
**H*****H*
***HH*HH**
CH***O**H*
*****O***H
OHH****HHO
```

```
15 15
HHCHHCHH
*********C*****
*********C*****
***CC***H*OO***
**HHH***H******
**HHH****H*****
*********H**CC*
**********HH***
***************
****CHHHHHHC***
**H***********C
*HCH**********H
*HCH*CC*******H
**H*H******O**H
****H******O**H
************CHH
```

```
5 10
HH
HHH*O
****O
**H**
**H**
HH*HH
***OO
H*H**
H*H**
*H*HH
H*O**
```

**Sample Output**
there are 7 of OHH
there are 5 of HCHHH
there are 1 of OO
there are 4 of HHCHHCHH
there are 5 of HH

# Problem 4:Vortex Cipher

The following secret code is based on the conversion of individual words:
Take an ordinary message, and first change all the letters to lower case letters, leaving all other characters, including spaces untouched.
Then consider each word according to the number of characters in it.

- If the word contains one single letter, replace it by the letter that comes before it in the alphabet, replacing b by a, c by b, etc, z by y. Replace a by z.
- If the word contains two letters, then switch the letters and replace each by the next letter in the alphabet: replacing the a by b, b by c, etc , y by z. replace the z by a.
- If the word contains an odd number of letters, 3 or more, then there is one single middle letter. Replace the middle letter by the first and last letter of the word and place the middle letter at the start. The words “tom” and “topples” become respectively: “otm” and “poptsle. Then replace each of the three letters involved by their predecessor in the alphabet, and so “tom” becomes “otm” first and then “nsl” and “topples” becomes “poptsle” first and then “oopsrle”.
- If the word contains an even number of letters, 4 or more, then there are two middle letters. The first middle letter is switched with the first letter, and the second middle letter is switched with the last letter. All four letters involved are then replaced by their successor in the alphabet. The words “mary” and “rosemary” for example will be replaced first by “amyr” and “eosryarm” and then “bnzs” and “fosszarn”.

Data41.txt (Data42.txt for the second try) contains five lines of less than 250 characters. Each line represents one secret message that had the Vortex Cipher applied to it. You must decipher each line.

**Sample Input:**
uimflu pc fqqf zgr pmut dgq dhroe
mzc docms's olxp dhvdr pu jgeo iunf.
zekdv iunf nlzdn mzc iuzf'mm pdfn pifn,
orichinh dhsqi hasrl iecenj iunf.
uimflu pc fqqf fgmm bgut msbqef

**Sample Output:**
little bo peep has lost her sheep
and doesn't know where to find them.
leave them alone and they'll come home,
bringing their tails behind them.
little bo peep fell fast asleep

# 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**1**1.txt (Data**1**2.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.

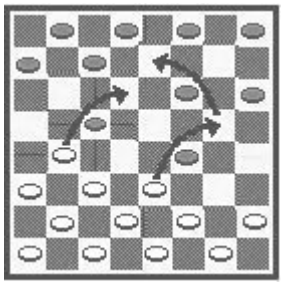
**Sample Input:**

```
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)*

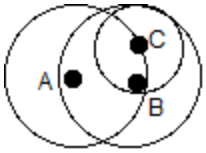
```
 36          BERNADETTE  ELYSE   LARRY    RENATA     WHITNEY
 AARON       DANA        EMMA    MARIA    ROCHELLE   WILLIAM
 ADAM        DANIEL      ERIC    MATTHEW  RYAN       1 28
 ADRIAN      DAVID       GIULIA  MICHAEL  SABRINA    23 15
 ALEX        DEAN        JULIA   MICHAEL  SHANE      9 33
 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

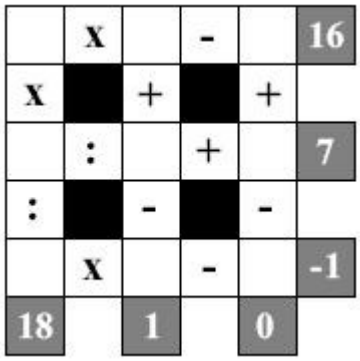
RIDDLE * L = PROBLEM
976624 * 2 = 1953248

TWO * TWO = SQUARE
567 * 567 = 321489

or:

TWO * TWO = SQUARE
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
```