

## PROGRAMMING CHALLENGE 21: LINEAR EQUATION

### Solving a Linear Equation

Input file: equation.txt

Save your file as: linear.py

A string containing integers, a lowercase letter, plus, minus and equal symbols is given in the input file. It can be viewed as a linear equation with one unknown variable. The unknown variable may appear on either side or both sides of the equation. Your task is to write a program which will read in the string from the input file, solve the equation and display the value of the variable.

Some examples of the string are given below:

4+3x=8	the unknown x appears on the left hand side of the equation
6a-5+1=2-4a	the unknown a appears on both sides of the equation
-5+12y=0	the unknown y appears on the left hand side and the right hand side is zero. Of course, zero may appear on the left instead.

For this problem, we have the following assumptions and requirements:

1. Other than + (plus), - (minus), = (equal) symbols and a letter, the string does not contain the blank character and other symbols.
2. The given equation is always solvable. Ill-formed equation or equation that has no definite answer, e.g.  $x = x$ , will not be given in the input.
3. The coefficient of the unknown variable is given before the variable. The output of your program should have 3 decimal places.

### Sample Input

6a-5+1=2-2a

### Sample Output

a= 0.750 #

## PROGRAMMING CHALLENGE 22: RUSSIAN PEASANT ALGORITHM

The Russian peasant algorithm is an alternative method to perform multiplication of numbers by consecutive application of doubling numbers, halving numbers and addition. Consider the multiplication of 57 by 86 (= 4902):

Write each number side by side:

57      86

Double the first number and halve the second number (by performing integer division by 2 i.e. drop the remainder).

If the second number or halving result is even, cross out this entire row.

Keep doubling, halving, and crossing out until the halving result is 1.

57	86
114	43
228	21
456	10
912	5
1824	2
3648	1

Add the remaining numbers in the first column.

The total is the product of the original numbers.

114	
228	
912	
3648	
4902	

### Task 1

Implement the Russian peasant multiplication algorithm. Get user input of 2 numbers with validation.

**Evidence 1:** Program code and screenshot of user input 57 and 86. [10]

The Russian peasant algorithm is actually related to binary numbers.

Doubling a decimal number is to shift its binary equivalent left, while halving a decimal number is to shift its binary equivalent right.

Consider the multiplication of 13 by 12 (= 156)

In Binary	In Decimal
1101 1100	13 12
11010 0110	26 6
110100 0011	52 3
1101000 0001	104 1

110100 + 1101000 = 10011100 = 156 (in decimal)

### Task 2

Implementing a stack class, write a `dec2bin()` method to convert a decimal number to its binary equivalent. Test your method with the numbers 57 and 86.

**Evidence 2:** Program code and screenshot. [10]

### Task 3

Implement the Russian peasant multiplication algorithm using binary numbers. Your program should accept 2 decimal numbers as input, convert them to binary using your `dec2bin()` method in task 2, and then perform addition of binary numbers to output a binary string as the result.

**Evidence 3:** Program code and screenshot. [10]