# **Solve a Linear Equation**

**Task:** The function is given a string representing a linear equation with one unknown. Find the value of x that is a valid solution of the equation. Return the string x=val. Three outcomes are possible:

* x\_val is an integer or a float (round x\_val to 2 decimals)
* "Infinite solutions" for situations 0\*x=0
* "No solution" for situations 0\*x=num, (num != 0)

**Examples:**

FindX("4x-7=x+11") ➞ "x=6"

FindX("3x=2x+x") ➞ "Infinite solutions"

FindX("3x=3x+2") ➞ "No solution"

**Notes:** All input numbers should only have at most 3 digits.

# **Quadratic Equation**

**Task:** Create a function to find only the root value of **x** in any quadratic equation . The function will take three arguments:

* **a** as the coefficient of **x^2**
* **b** as the coefficient of **x**
* **c** as the constant term

**Examples:**

quadraticEquation(1, -12, -28) ➞ 14

quadraticEquation(2, -7, 3) ➞ 3

quadraticEquation(1, 2, -3) ➞ 1

**Notes:**

* Quadratic equation is always guaranteed to have a root.
* Calculate only the root that sums the square root of the discriminant, not the one that subtracts it.
* Round the value / return only integer value.

# **How Many Solutions Does This Quadratic Have?**

A quadratic equation has either 0, 1, or 2 distinct solutions for real values of **x**.

**Task:** Given **a**, **b** and **c**, you should return the number of solutions to the equation.

**Examples:**

solutions(1, 0, -1) ➞ 2

// x² - 1 = 0 has two solutions (x = 1 and x = -1).

solutions(1, 0, 0) ➞ 1

// x² = 0 has one solution (x = 0).

solutions(1, 0, 1) ➞ 0

// x² + 1 = 0 has no real solutions.

**Notes:**

* You do not have to calculate the solutions, just return how many there are.
* “**a**” will always be non-zero.

# **Vertex of a Quadratic**

Every quadratic curve has a vertex point: the turning point where the curve stops heading down and starts going up.

**Task:** Given the values **a**, **b** and **c**, where **a** is never 0, you need to return the coordinates of the vertex. Return your answers rounded to 2 decimal places (a double).

**Examples:**

FindVertex(1, 0, 25) ➞ [0, 25]

// The vertex of y=x²+25 is at (0, 25).

FindVertex(-1, 0, 25) ➞ [0, 25]

// The vertex of y=-x²+25 is at (0, 25).

FindVertex(1, 10, 4) ➞ [-5, -21]

// The vertex of y=x²+10x+4 is at (-5, -21).

**Notes:**

* A quadratic equation can be expressed two ways:
  + Standard form:
  + Vertex form:
    - The vertex is
    - The coefficient controls whether the parabola opens upward or downward, as well as the speed of increase or decrease of the parabola.

# **Sum of Quadratic Roots**

**Task:** Given **a**, **b** and **c**, find the roots of the equation and then add them together.

**Examples:**

findRootsSum(2, 4, -6) ➞ -2.00

findRootsSum(3, 4, -3) ➞ -1.33

findRootsSum(4, 3, -8) ➞ -0.75

**Notes:**

* Round your answer to two decimal places.
* If there is only one real root return 1.
* If there are no real roots, return 0.

# **Quartic Equation**

**Task:** Create a function that returns the number of (real) solutions of . The function will take three arguments: **a** as the coefficient of , **b** as the coefficient of , and **c** as the constant term.

**Examples:**

quarticEquation(1, -5, 4) ➞ 4

quarticEquation(4, 3, -1) ➞ 2

quarticEquation(1, 10, 9) ➞ 0

**Notes:** Try substitution t=x^2.

# **Integral of a Function**

**Task:** Create a function that takes numbers **b**, **m**, and **n** as arguments and returns the definite integral of the function with respect to **x** from **x=m** **to x=n**, where **b**, **m**, and **n** are constants.

**Examples:**

Integral(0, 2, 5) ➞ 3

Integral(2, 4, 7) ➞ 279

Integral(5, 9, 3) ➞ -530712

**Notes:**

* **^** in the context of this challenge means "to the power of", also known as the "exponent" operator.
* Assume that **b** will be an integer greater than or equal to 0.
* **m** and **n** can be any integer, both positive and negative.

# **Maximum Travel Distance**

**Task:** Write a function that takes **fuel** (liters), **fuelUsage** (liters/100km), **passengers**, **airCon** (boolean) and returns maximum distance that car can travel.

* **fuel** is the number of liters of fuel in the fuel tank.
* **fuelUsage** is basic fuel consumption per 100 km (with the driver inside only).
* Every additional passenger is increasing basic fuel consumption by 5%.
* If the air conditioner is ON (true), its increasing the total (not basic) fuel consumption by 10%.

**Examples:**

totalDistance(70.0, 7.0, 0, false) ➞ 1000.0

totalDistance(36.1, 8.6, 3, true) ➞ 331.8

totalDistance(55.5, 5.5, 5, false) ➞ 807.3

**Notes:**

* **fuel** and **fuelUsage** are always greater than 1.
* **passengers** are always greater or equal to 0.
* Round your answer to the nearest tenth.

# **Clock Angle Problem**

**Task:** Given a time in “**hh:mm**” format, calculate the shorter angle between the hour and minute hand in an analog clock

**Examples:**

Input: 5:30 Output: 15°

Input: 9:00 Output: 90°

Input: 12:00 Output: 80

**Notes:**

* hh:60 should be considered as (hh+1):0.
* The idea is to consider the rate of change of the angle in degrees per minute. The hour hand of a 12-hour analog clock turns 360° in 12 hours, and the minute hand rotates through 360° in 60 minutes. So, we can calculate the angle in degrees of the hour hand minute hand separately and return their difference using the following formula:
  + **Degree(hr) = Hx (360/12) + (Mx360)/(12x60) Degree(min) = Mx(360/60)**
* Here, H is the hour, and M is the minutes past the hour. The angle should be in degrees and measured clockwise from the 12 o'clock position of the clock. If the angle is greater than 180° , take its difference with 360.