Experiment 1

Title of the Experiment: Solving Linear equations efficiently

Aim of the Experiment: Write python code to solve the following linear equation efficiently:

1.

2.

3. = 0.4y

Theory:

Linear Equation: It must be solvable in its pure form. This means that it can only have one variable, usually written as x. A two-variable equation would require multiple linear equations (a system of equations) to be solved. A linear equation consists of three primary components — constants, variables, and multipliers. Any number or combination of operations — addition, subtraction, multiplication, and division — are valid, with any parenthesis scope. As long as it abides by these definitions of a linear equation, it is solvable by our function.

- expression = equation.replace("="," - (")+")"

  This transforms the equation into an expression to be evaluated by moving the entire expression on the right side of the equation to the left side.
- grouped = eval(expression.replace(var,'1j')

  The eval function takes in an expression. By substituting an unknown variable x with the natively understood j (i), Python treats two categories of expression elements — variables and constants — as separate. When the expression is evaluated, the answer comes out to be a*j + b, which Python believes is a complex number. Because j was used as a substitute for x, the result is a simplified and easily solvable linear equation.
- return -grouped.real/grouped.imag

  returns the negative of the real component of the complex number divided by the imaginary component.

Example: **2x+3 = x-6**

2x+3 – (x-6) =0

x=-9

Pre lab Questions:

1. What is Python?
2. What is an interpreted Language
3. What are the supported data types in python

Post Lab Questions:

1. Name three features of python.
2. What is Jupyter Notebook?
3. Name the Python IDEs?

Program

```
# Expt 1c: Solving Linear equations
# Class: III Year EKE
# Course: 18ECS301J – Applied Programming Lab
# Student Name and Reg Number: ___________________________________

from math import *
def solve_linear(equation,var='x'):
   expression = equation.replace("=","-(")+")"
   grouped = eval(expression.replace(var,'1j'))
   return -grouped.real/grouped.imag
solve_linear("2*x+3=x-6")
```

Sample Output

```
-9.0
```

Result: