

```

1  # Lab 1: Math and String shtuff
2  # Whew, this was tedious.
3  # Andrea Smith
4  # CSCI 1913
5
6  def left(e):
7      return e[0]
8
9  def op(e):
10     return e[1]
11
12 def right(e):
13     return e[2]
14
15 def isInside(v,e): # v is a variable, e is an equation
16     if v == e:
17         return True
18     elif type(e) == str:
19         return False
20     else:
21         return isInside(v,left(e)) or isInside(v,right(e))
22
23 def solving(v,q):
24     if v == left(q):
25         return q
26     elif op(left(q)) == "+":
27         return solvingAdd(v,q)
28     elif op(left(q)) == "-":
29         return solvingSubtract(v,q)
30     elif op(left(q)) == "*":
31         return solvingMultiply(v,q)
32     elif op(left(q)) == "/":
33         return solvingDivide(v,q)
34
35 def solve(v,q):
36
37     if isInside(v, left(q)):
38         return solving(v,q)
39
40     elif isInside(v, right(q)):
41         return solving(v, [q[2], q[1], q[0]])
42
43     else:
44         return None
45
46 def solvingAdd(v,q):
47     a = left(left(q))      # gets farthest left var within the left

```

```

• expression (i.e. if left is  $x + b$ , will give  $x$ )
48 b = right(left(q))
49 c = right(q)
50
51 if isInside(v,a):
52     newEquation = (a, "=", (c, "-", b))
53     return solving(v,newEquation)
54 else:
55     newEquation = (b, "=", (c, "-", a))
56     return solving(v,newEquation)
57
58 def solvingSubtract(v,q):
59     a = left(left(q))      # gets farthest left var within the
• left expression (i.e. if left is  $x + b$ , will give  $x$ )
60 b = right(left(q))
61 c = right(q)
62
63 if isInside(v,a):
64     newEquation = (a, "=", (c, "+", b))
65     return solving(v,newEquation)
66 elif isInside(v,b):
67     newEquation = (b, "=", (a, "-", c))
68     return solving(v,newEquation)
69 else:
70     return None
71
72 def solvingMultiply(v,q):
73     a = left(left(q))      # gets farthest left var within the
• left expression (i.e. if left is  $x + b$ , will give  $x$ )
74 b = right(left(q))
75 c = right(q)
76
77 if isInside(v,a):
78     newEquation = (a, "=", (c, "/", b))
79     return solving(v,newEquation)
80 elif isInside(v,b):
81     newEquation = (b, "=", (c, "/", a))
82     return solving(v,newEquation)
83 else:
84     return None
85
86 def solvingDivide(v,q):
87     a = left(left(q))      # gets farthest left var within the
• left expression (i.e. if left is  $x + b$ , will give  $x$ )
88 b = right(left(q))
89 c = right(q)
90

```

```

90
91     if isInside(v,a):
92         newEquation = (a, "=", (c, "*", b))
93         return solving(v,newEquation)
94     elif isInside(v,b):
95         newEquation = (b, "=", (a, "/", c))
96         return solving(v,newEquation)
97     else:
98         return None
99
100 #
101 # TESTS. Test the equation solver for CSci 1913 Lab 1.
102 #
103 # James Moen
104 # 10 Sep 18
105 #
106 # Every test is followed by a comment which shows what must be
107 # • printed if your
108 # • code works correctly. It also shows how many points the test is
109 # • worth, for a
110 # total of 35 possible points.
111 #
112 print(isInside('x', 'x')) # True 1 point
113 print(isInside('x', 'y')) # False 1 point
114 print(isInside('x', ('x', '+', 'y'))) # True 2 points
115 print(isInside('x', ('a', '+', 'b'))) # False 2 points
116 print(isInside('+', ('a', '+', 'b'))) # False 2 points
117 print(isInside('x', (('m', '*', 'x'), '+', 'b'))) # True 2 points
118
119 print(solve('x', (('a', '+', 'x'), '=', 'c')))
120 # ('x', '=', ('c', '-', 'a')) 2 points
121
122 print(solve('x', (('x', '+', 'b'), '=', 'c')))
123 # ('x', '=', ('c', '-', 'b')) 2 points
124
125 print(solve('x', (('a', '-', 'x'), '=', 'c')))
126 # ('x', '=', ('a', '-', 'c')) 2 points
127
128 print(solve('x', (('x', '-', 'b'), '=', 'c')))
129 # ('x', '=', ('c', '+', 'b')) 2 points
130
131 print(solve('x', (('a', '*', 'x'), '=', 'c')))
132 # ('x', '=', ('c', '/', 'a')) 2 points
133
134 print(solve('x', (('x', '*', 'b'), '=', 'c')))
135 # ('x', '=', ('c', '/', 'b')) 2 points

```

```
135
136 print(solve('x', (('a', '/', 'x'), '=', 'c'))))
137 # ('x', '=', ('a', '/', 'c')) 2 points
138
139 print(solve('x', (('x', '/', 'b'), '=', 'c'))))
140 # ('x', '=', ('c', '*', 'b')) 2 points
141
142 print(solve('y', ('y', '=', (('m', '*', 'x'), '+', 'b'))))
143 # ('y', '=', (('m', '*', 'x'), '+', 'b')) 2 points
144
145 print(solve('x', ('y', '=', (('m', '*', 'x'), '+', 'b'))))
146 # ('x', '=', (('y', '-', 'b'), '/', 'm')) 2 points
147
148 print(solve('a', (('b', '+', 'c'), '=', ('d', '*', (('a', '/', 'e'),
    • '- ', 'f')))))
149 # ('a', '=', (((('b', '+', 'c'), '/', 'd'), '+', 'f'), '*', 'e')) 5
    • points
150
```