Kyle Bowman

Python Programming

1/19/2017

Homework 1

Problem 1-

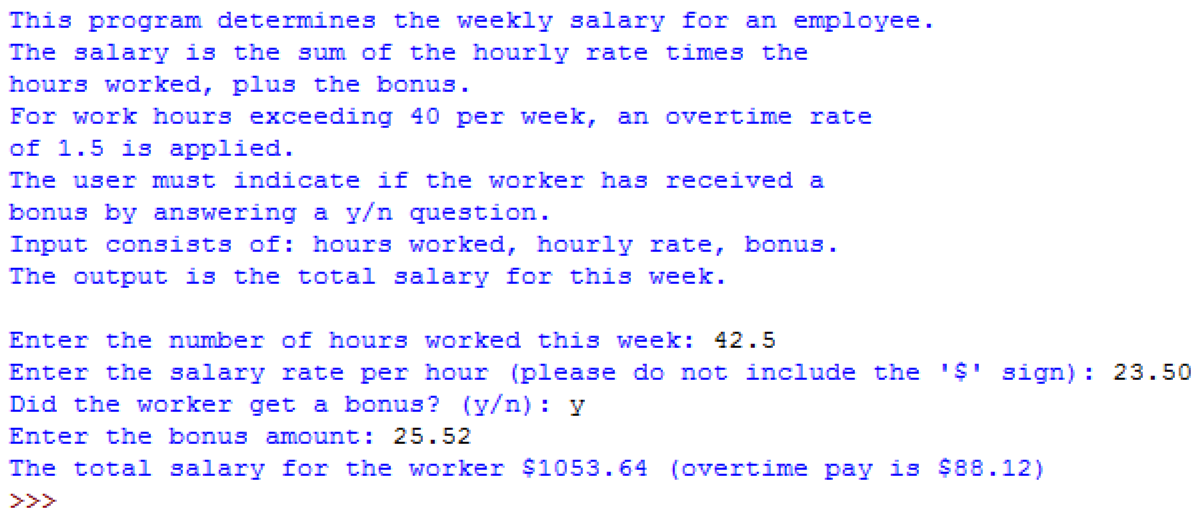
```python
print("This program determines the weekly salary for an employee.")
print("The salary is the sum of the hourly rate times the \nhours worked, plus the bonus.")
print("For work hours exceeding 40 per week, an overtime rate \nof 1.5 is applied.")
print("The user must indicate if the worker has received a \nbonus by answering a y/n question.")
print("Input consists of: hours worked, hourly rate, bonus.")
print("The output is the total salary for this week.\n")

hours_worked = float(input("Enter the number of hours worked this week: "))
rate_per_hour = float(input("Enter the salary rate per hour (please do not include the '$' sign): "))
yes_no = input("Did the worker get a bonus? (y/n): ")

if yes_no == 'y':
    bonus = float(input("Enter the bonus amount: "))
else:
    bonus = 0

if hours_worked > 40:
    overtime = (hours_worked - 40) * rate_per_hour * 1.5
    non_overtime = 40 * rate_per_hour
    salary = non_overtime + overtime + bonus
else:
    overtime = 0
    non_overtime = hours_worked * rate_per_hour
    salary = non_overtime + bonus

overtime = "{0:.2f}".format(overtime)
salary = "{0:.2f}".format(salary)
print("The total salary for the worker $" + salary + " (overtime pay is $" + overtime + ")")
```



```
This program determines the weekly salary for an employee.
The salary is the sum of the hourly rate times the
hours worked, plus the bonus.
For work hours exceeding 40 per week, an overtime rate
of 1.5 is applied.
The user must indicate if the worker has received a
bonus by answering a y/n question.
Input consists of: hours worked, hourly rate, bonus.
The output is the total salary for this week.

Enter the number of hours worked this week: 42.5
Enter the salary rate per hour (please do not include the '$' sign): 23.50
Did the worker get a bonus? (y/n): y
Enter the bonus amount: 25.52
The total salary for the worker $1053.64 (overtime pay is $88.12)
>>>
```

Problem 2-

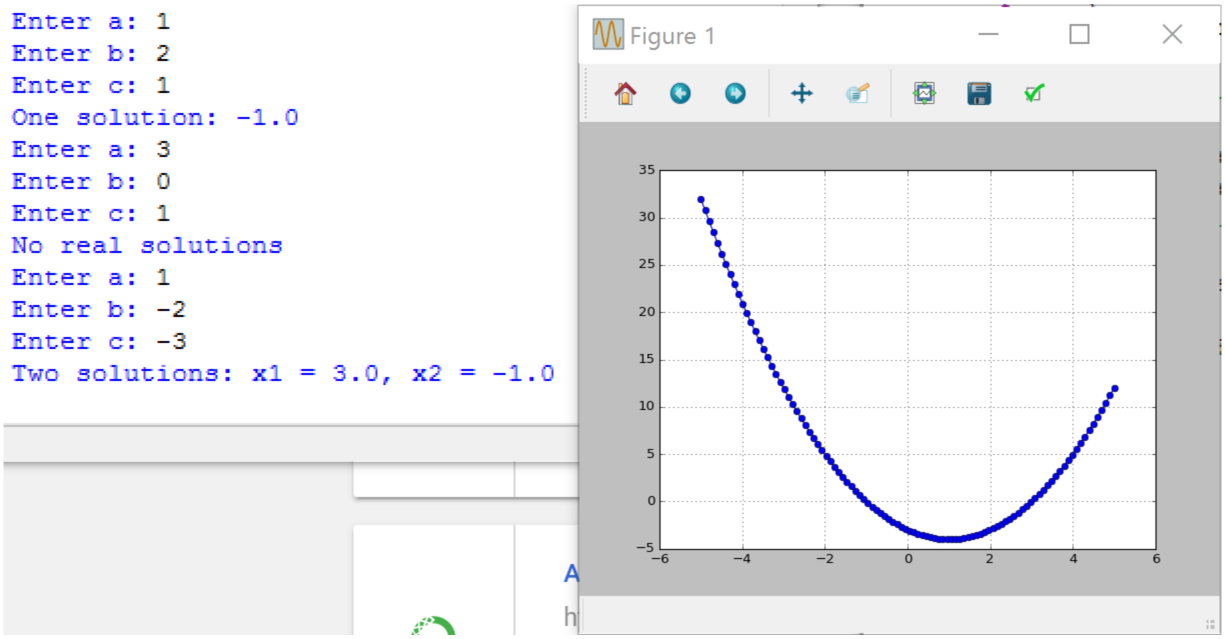
```
import pylab
import math

while True:
    a = float(input("Enter a: "))
    b = float(input("Enter b: "))
    c = float(input("Enter c: "))

    under_the_root = float(b * b - (4 * a * c))

    if under_the_root < 0:
        print("No real solutions")
    elif under_the_root == 0:
        x_1 = (-b)/(2 * a)
        print("One solution: {0}".format(x_1))
    else:
        x_1 = (-b + math.sqrt(under_the_root))/(2 * a)
        x_2 = (-b - math.sqrt(under_the_root))/(2 * a)
        print("Two solutions: x1 = {0}".format(x_1) + ", x2 = {0}".format(x_2))

    x = pylab.linspace(-5, 5, 100)
    quad_form = a * x * x + b * x + c
    pylab.plot(x, quad_form, '-bo')
    pylab.grid()
    pylab.show()
```



```
Enter a: 1
Enter b: 2
Enter c: 1
One solution: -1.0
Enter a: 3
Enter b: 0
Enter c: 1
No real solutions
Enter a: 1
Enter b: -2
Enter c: -3
Two solutions: x1 = 3.0, x2 = -1.0
```

Problem 3-

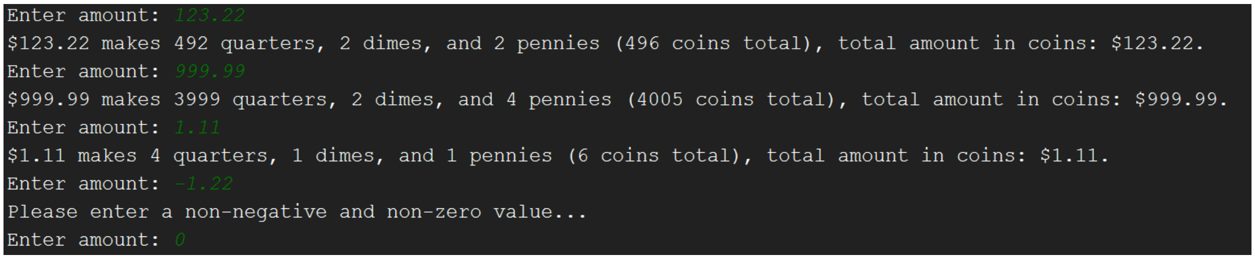
Algorithm-

1. Begin infinite loop
2. Get user input for the amount to get change for
3. Check the value entered and check if it larger than or equal to zero. If not, tell user to enter values ago and go back to step 2. If it is at least 0 or larger, go to step 4.
4. Begin greedy algorithm starting with largest value, 0.25, and going down to 0.10 then finally to 0.01 to. We check if we can give change (the value of the amount remaining must be larger than or equal to the value of the coin). This check is done and the proper amount of change given then the amount is updated. We repeat this for all the values of change.
5. We generate the count amount by adding all 3 different coin amounts together
6. Finally, we print out all relevant information.

```python
while True:
    # Initialize the variables to zero at the beginning of each loop
    amount = float(input("Enter amount: "))
    original_amount = amount

    if amount < 0:
        print("Please enter a non-negative and non-zero value...")
    elif amount >= 0:
        # use round when doing floating point arithmetic/algebra to make up for limitations
        if amount >= 0.25:
            quarter_count = int(amount / 0.25)
            amount = round(amount % 0.25, 2)
            # amount = round(amount - quarter_count * 0.25, 2)
        if amount >= 0.10:
            dime_count = int(amount / 0.10)
            amount = round(amount % 0.10, 2)
        if amount >= 0.01:
            penny_count = int(amount / 0.01)
            amount = round(amount % 0.01, 2)

    coin_count = quarter_count + dime_count + penny_count
    if original_amount > 0:
        # format string to print in specific order and to display proper decimal places
        print("${:.2f} makes {} quarters, {} dimes, and {} pennies ({} coins total), \
total amount in coins: ${:.2f}.".format(original_amount, quarter_count, dime_count,
                                        penny_count, coin_count, original_amount))
```



```
Enter amount: 123.22
$123.22 makes 492 quarters, 2 dimes, and 2 pennies (496 coins total), total amount in coins: $123.22.
Enter amount: 999.99
$999.99 makes 3999 quarters, 2 dimes, and 4 pennies (4005 coins total), total amount in coins: $999.99.
Enter amount: 1.11
$1.11 makes 4 quarters, 1 dimes, and 1 pennies (6 coins total), total amount in coins: $1.11.
Enter amount: -1.22
Please enter a non-negative and non-zero value...
Enter amount: 0
```