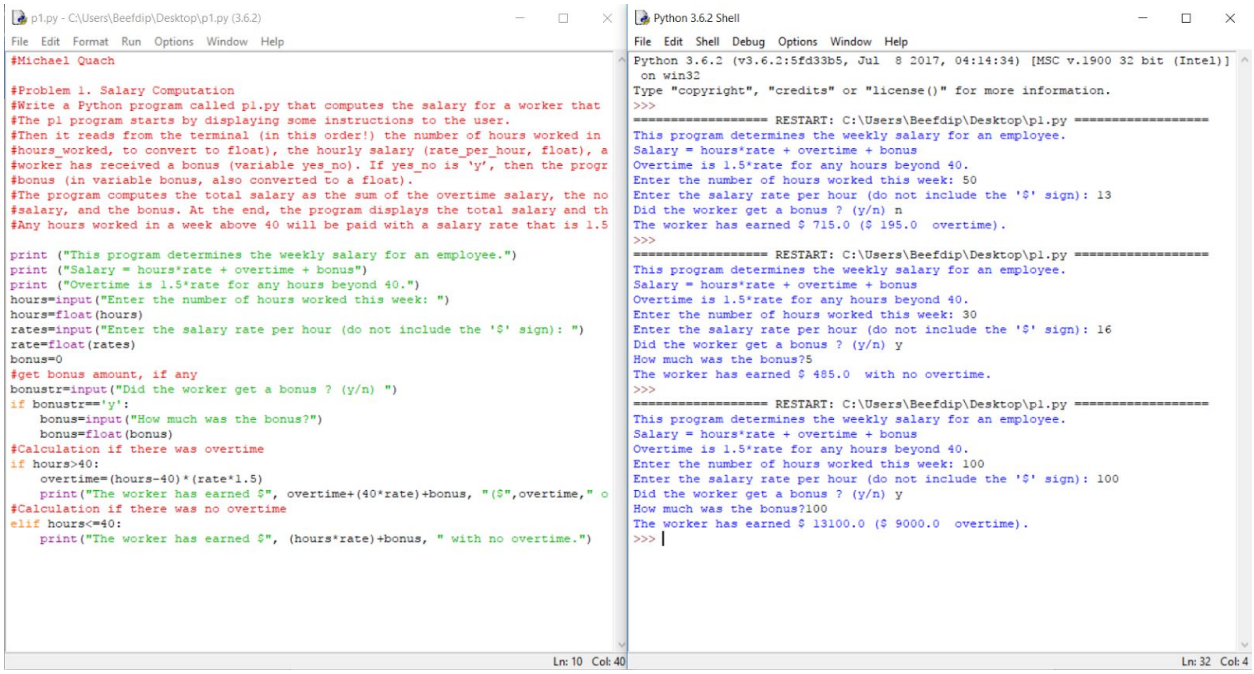
Homework 1
Michael Quach

Problem 1

```
#Michael Quach

#Problem 1. Salary Computation
#Write a Python program called p1.py that computes the salary for a worker that is paid weekly.
#The p1 program starts by displaying some instructions to the user.
#Then it reads from the terminal (in this order!) the number of hours worked in a week (variable
#hours_worked, to convert to float), the hourly salary (rate_per_hour, float), and whether the
#worker has received a bonus (variable yes_no). If yes_no is 'y', then the program reads the
#bonus (in variable bonus, also converted to a float).
#The program computes the total salary as the sum of the overtime salary, the non-overtime
#salary, and the bonus. At the end, the program displays the total salary and the overtime pay.
#Any hours worked in a week above 40 will be paid with a salary rate that is 1.5 times the
regular hourly rate.

print ("This program determines the weekly salary for an employee.")
print ("Salary = hours*rate + overtime + bonus")
print ("Overtime is 1.5*rate for any hours beyond 40.")
hours=input("Enter the number of hours worked this week: ")
hours=float(hours)
rates=input("Enter the salary rate per hour (do not include the '$' sign): ")
rate=float(rates)
bonus=0
#get bonus amount, if any
bonustr=input("Did the worker get a bonus ? (y/n) ")
if bonustr=='y':
   bonus=input("How much was the bonus?")
   bonus=float(bonus)
#Calculation if there was overtime
if hours>40:
   overtime=(hours-40)*(rate*1.5)
   print("The worker has earned $", overtime+(40*rate)+bonus, "($",overtime," overtime).")
#Calculation if there was no overtime
elif hours<=40:
   print("The worker has earned $", (hours*rate)+bonus, " with no overtime.")
```



Problem 2

#Michael Quach

##
##

#Write all code for this problem in a file p2.py.

#1. The program consists of a main while loop (an infinite loop) in which the user is prompted to enter values for coefficients a, b, and c.

#Assume the user types valid float numbers from the terminal.

#The program converts the input to float type and then uses formula (1) above to compute solutions x1 and x2, as follows:

#a) if b2-4ac<0 then the solutions are complex numbers (i.e. not real) and the program displays #the string "no real solutions",

#b) if b2-4ac=0 then x1=x2 and the program displays: "one solution: " followed by the value x1.

#c) if b2-4ac>0 then the solutions are distinct and the program displays "two solutions: " #followed by the values of x1 and x2.

#To keep the problem simple we can assume that the user never enters a value for coefficient a that is equal to 0.

#To stop the loop and to end the program, the user types the enf-of-file key – CTRL-Z on Windows

#(CTRL-D on Linux/Mac/Unix) – when expected to enter coefficient a for a new iteration.

##
##

```
import pylab
import math
```

#NOTICE: Since you had us install Python via Anaconda, I couldn't figure out

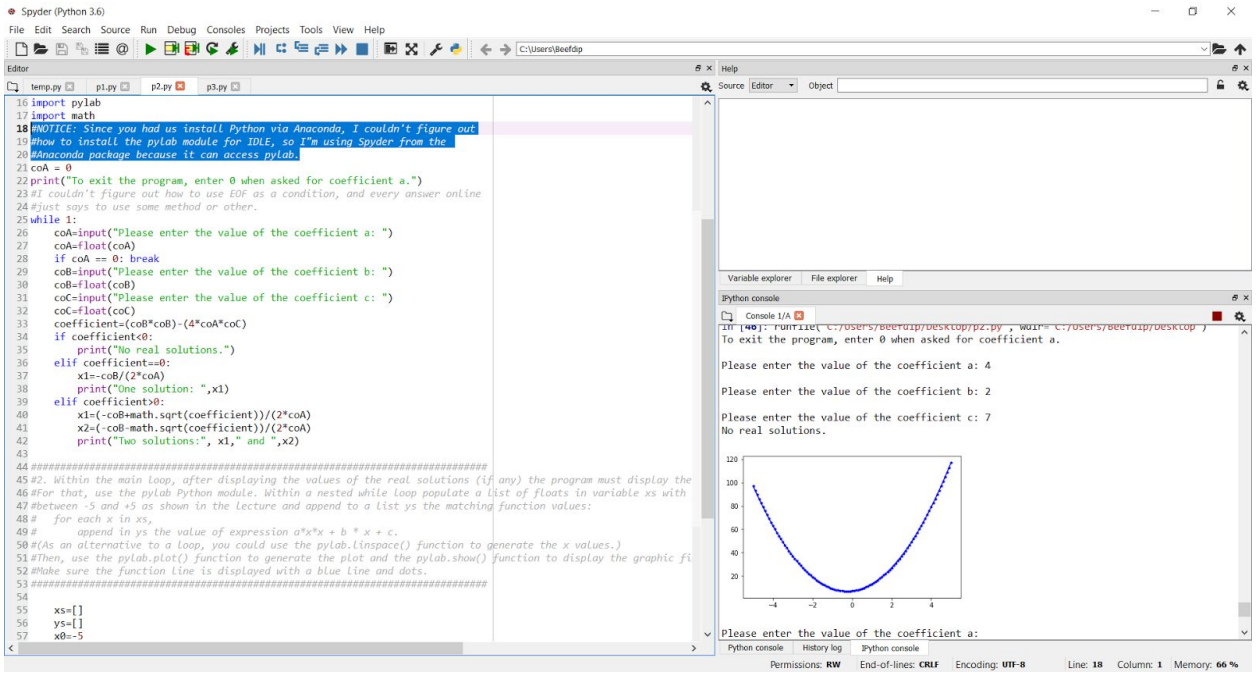
```
#how to install the pylab module for IDLE, so I"m using Spyder from the
#Anaconda package because it can access pylab.
coA = 0
print("To exit the program, enter 0 when asked for coefficient a.")
#I couldn't figure out how to use EOF as a condition, and every answer online just says to use
some method or other.
while 1:
    coA=input("Please enter the value of the coefficient a: ")
    coA=float(coA)
    if coA == 0: break
    coB=input("Please enter the value of the coefficient b: ")
    coB=float(coB)
    coC=input("Please enter the value of the coefficient c: ")
    coC=float(coC)
    coefficient=(coB*coB)-(4*coA*coC)
    if coefficient<0:
        print("No real solutions.")
    elif coefficient==0:
        x1=-coB/(2*coA)
        print("One solution: ",x1)
    elif coefficient>0:
        x1=(-coB+math.sqrt(coefficient))/(2*coA)
        x2=(-coB-math.sqrt(coefficient))/(2*coA)
        print("Two solutions:", x1," and ",x2)

##############################################################################
##
#2. Within the main loop, after displaying the values of the real solutions (if any) the program
must display the graphic of the quadratic function on the [-5, 5] domain.
#For that, use the pylab Python module. Within a nested while loop populate a list of floats in
variable xs with 100 float numbers
#between -5 and +5 as shown in the lecture and append to a list ys the matching function
values:
#   for each x in xs,
#       append in ys the value of expression a*x*x + b * x + c.
#(As an alternative to a loop, you could use the pylab.linspace() function to generate the x
values.)
#Then, use the pylab.plot() function to generate the plot and the pylab.show() function to display
the graphic figure.
#Make sure the function line is displayed with a blue line and dots.
##############################################################################
##
```

```
xs=[]
ys=[]
x0=-5
x1=+5
x=x0
n=100
dx=(x1-x0)/n

while x<=x1:
    xs.append(x)
    y=(coA*x*x)+(coB*x)+coC
    ys.append(y)
    x+=dx

pylab.plot(xs, ys, "b.-")
pylab.show()
```

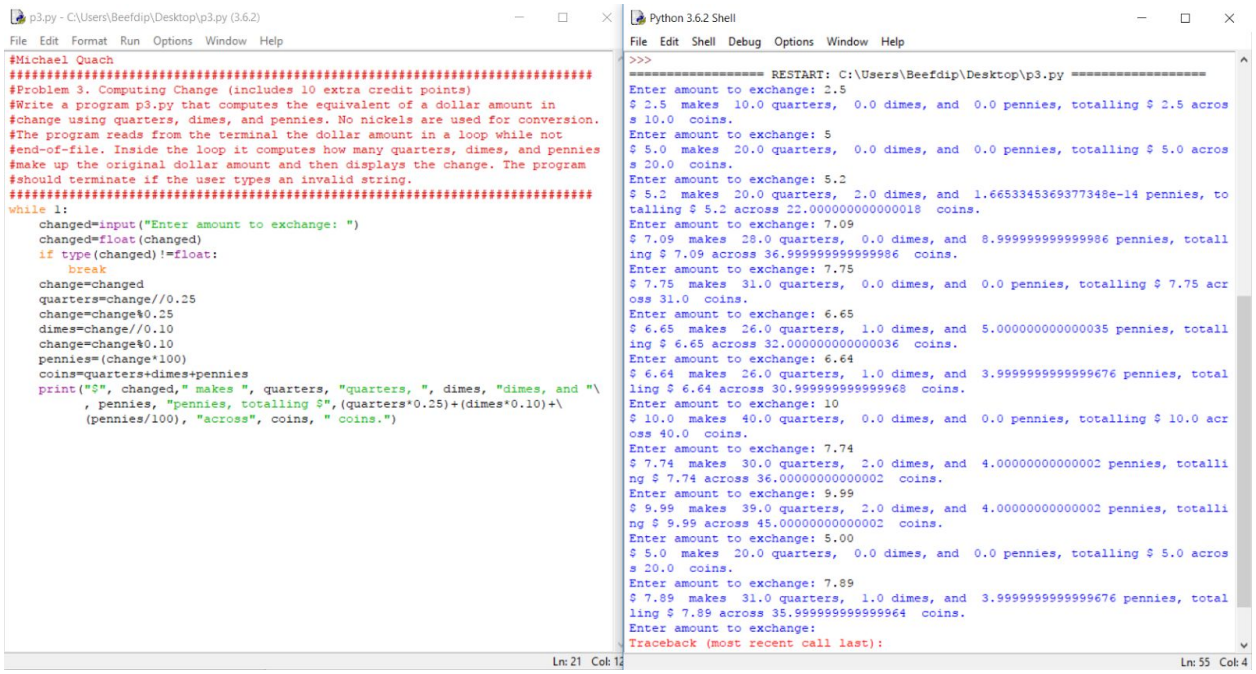


Problem 3

Algorithm:

input=user_input;
number_of_quarters = input/0.25 [without remainder]
input -= number_of_quarters*0.25
number_of_dimes = input
Input -= number_of_dimes*0.10
Number_of_pennies = input*100

```
#Michael Quach
################################################################################
##
#Problem 3. Computing Change (includes 10 extra credit points)
#Write a program p3.py that computes the equivalent of a dollar amount in
#change using quarters, dimes, and pennies. No nickels are used for conversion.
#The program reads from the terminal the dollar amount in a loop while not
#end-of-file. Inside the loop it computes how many quarters, dimes, and pennies
#make up the original dollar amount and then displays the change. The program
#should terminate if the user types an invalid string.
################################################################################
##
while 1:
    changed=input("Enter amount to exchange: ")
    changed=float(changed)
    if type(changed)!=float:
        break
    change=changed
    quarters=change//0.25
    change=change%0.25
    dimes=change//0.10
    change=change%0.10
    pennies=(change*100)
    coins=quarters+dimes+pennies
    print("$", changed," makes ", quarters, "quarters, ", dimes, "dimes, and "\
        , pennies, "pennies, totalling $",(quarters*0.25)+(dimes*0.10)+\
        (pennies/100), "across", coins, " coins.")
```



p3.py - C:\Users\Beefdip\Desktop\p3.py (3.6.2)

File Edit Format Run Options Window Help

```
#Michael Quach
################################################################################
#Problem 3. Computing Change (includes 10 extra credit points)
#Write a program p3.py that computes the equivalent of a dollar amount in
#change using quarters, dimes, and pennies. No nickels are used for conversion.
#The program reads from the terminal the dollar amount in a loop while not
#end-of-file. Inside the loop it computes how many quarters, dimes, and pennies
#make up the original dollar amount and then displays the change. The program
#should terminate if the user types an invalid string.
################################################################################
while 1:
    changed=input("Enter amount to exchange: ")
    changed=float(changed)
    if type(changed)!=float:
        break
    change=changed
    quarters=change//0.25
    change=change%0.25
    dimes=change//0.10
    change=change%0.10
    pennies=(change*100)
    coins=quarters+dimes+pennies
    print("$", changed," makes ", quarters, "quarters, ", dimes, "dimes, and "\
          , pennies, "pennies, totalling $",(quarters*0.25)+(dimes*0.10)+\
          (pennies/100), "across", coins, " coins.")
```

Ln: 21 Col: 12

Python 3.6.2 Shell

File Edit Shell Debug Options Window Help

```
>>>
========== RESTART: C:\Users\Beefdip\Desktop\p3.py ==========
Enter amount to exchange: 2.5
$ 2.5  makes  10.0 quarters,  0.0 dimes, and  0.0 pennies, totalling $ 2.5 acros
s 10.0  coins.
Enter amount to exchange: 5
$ 5.0  makes  20.0 quarters,  0.0 dimes, and  0.0 pennies, totalling $ 5.0 acros
s 20.0  coins.
Enter amount to exchange: 5.2
$ 5.2  makes  20.0 quarters,  2.0 dimes, and  1.6653345369377348e-14 pennies, to
talling $ 5.2 across 22.000000000000018  coins.
Enter amount to exchange: 7.09
$ 7.09  makes  28.0 quarters,  0.0 dimes, and  8.999999999999986 pennies, totall
ing $ 7.09 across 36.999999999999986  coins.
Enter amount to exchange: 7.75
$ 7.75  makes  31.0 quarters,  0.0 dimes, and  0.0 pennies, totalling $ 7.75 acr
oss 31.0  coins.
Enter amount to exchange: 6.65
$ 6.65  makes  26.0 quarters,  1.0 dimes, and  5.000000000000035 pennies, totall
ing $ 6.65 across 32.000000000000036  coins.
Enter amount to exchange: 6.64
$ 6.64  makes  26.0 quarters,  1.0 dimes, and  3.9999999999999676 pennies, total
ling $ 6.64 across 30.999999999999968  coins.
Enter amount to exchange: 10
$ 10.0  makes  40.0 quarters,  0.0 dimes, and  0.0 pennies, totalling $ 10.0 acr
oss 40.0  coins.
Enter amount to exchange: 7.74
$ 7.74  makes  30.0 quarters,  2.0 dimes, and  4.00000000000002 pennies, totalli
ng $ 7.74 across 36.00000000000002  coins.
Enter amount to exchange: 9.99
$ 9.99  makes  39.0 quarters,  2.0 dimes, and  4.00000000000002 pennies, totalli
ng $ 9.99 across 45.00000000000002  coins.
Enter amount to exchange: 5.00
$ 5.0  makes  20.0 quarters,  0.0 dimes, and  0.0 pennies, totalling $ 5.0 acros
s 20.0  coins.
Enter amount to exchange: 7.89
$ 7.89  makes  31.0 quarters,  1.0 dimes, and  3.9999999999999676 pennies, total
ling $ 7.89 across 35.999999999999964  coins.
Enter amount to exchange:
Traceback (most recent call last):
```

Ln: 55 Col: 4