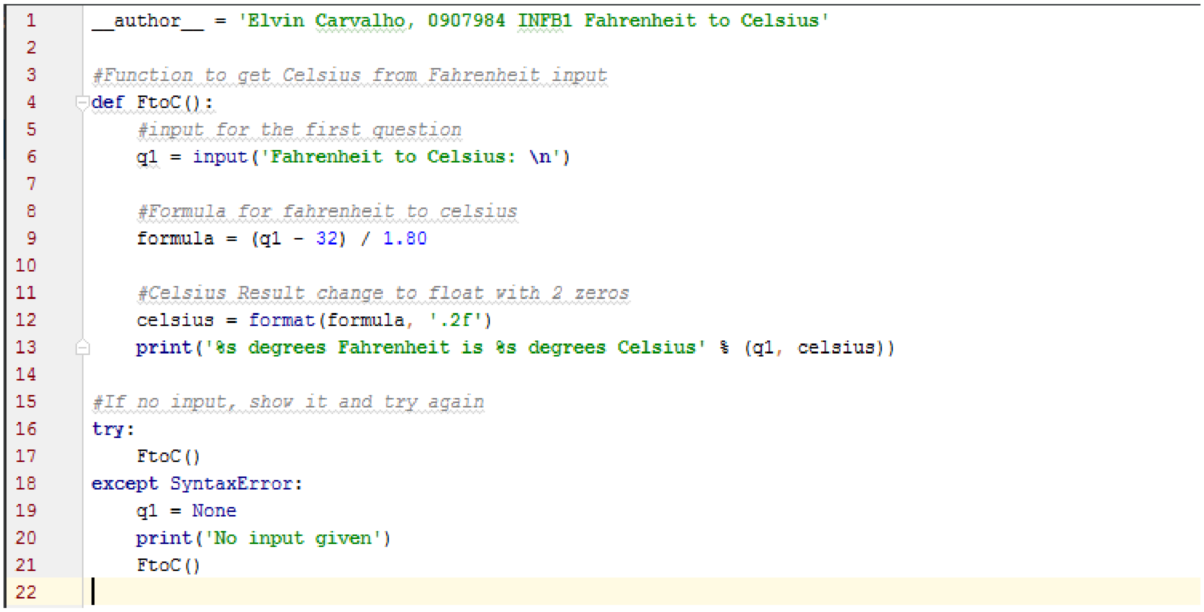
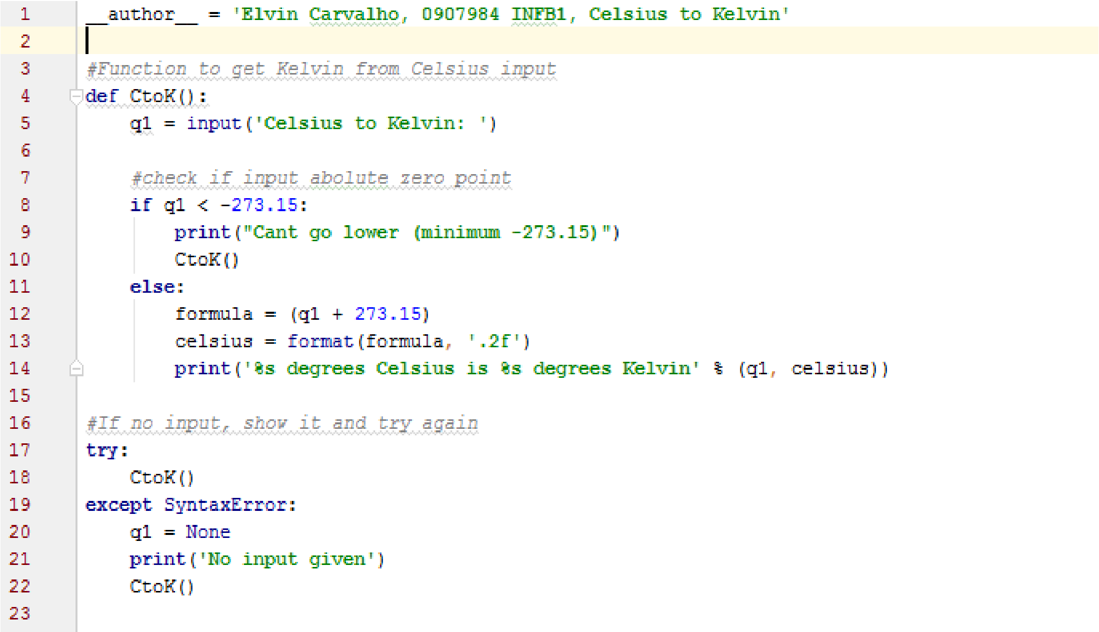
ASSIGNMENT 4

EXERCISE 1a



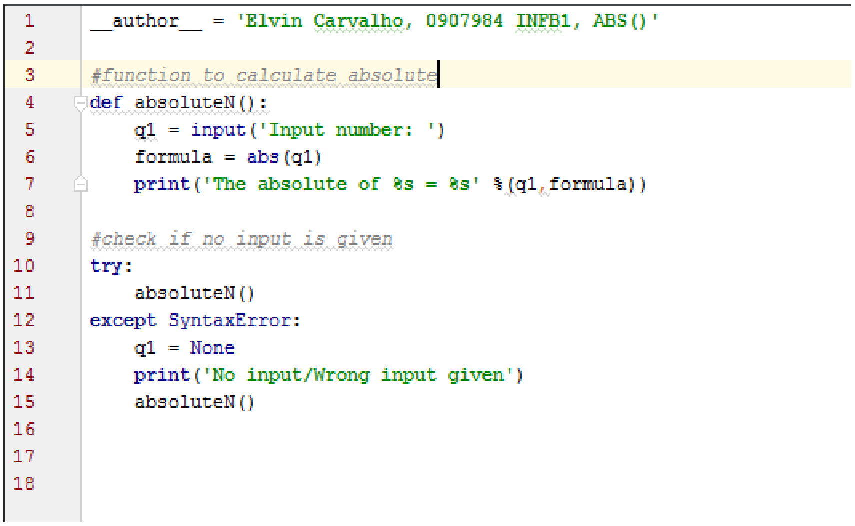
```python
__author__ = 'Elvin Carvalho, 0907984 INFB1 Fahrenheit to Celsius'

#Function to get Celsius from Fahrenheit input
def FtoC():
    #input for the first question
    q1 = input('Fahrenheit to Celsius: \n')

    #Formula for fahrenheit to celsius
    formula = (q1 - 32) / 1.80

    #Celsius Result change to float with 2 zeros
    celsius = format(formula, '.2f')
    print('%s degrees Fahrenheit is %s degrees Celsius' % (q1, celsius))

#If no input, show it and try again
try:
    FtoC()
except SyntaxError:
    q1 = None
    print('No input given')
    FtoC()

```

EXERCISE 1b



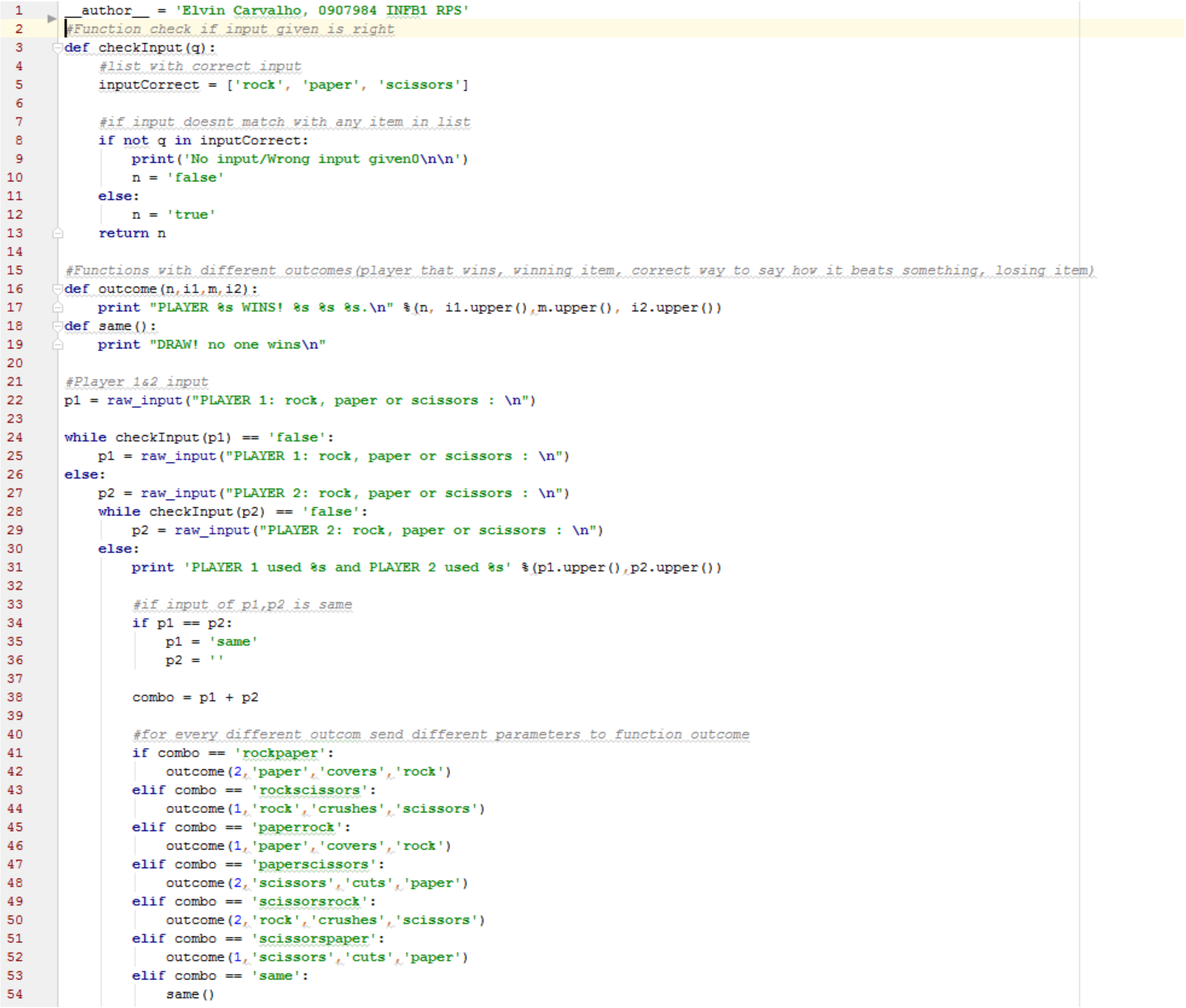
```python
__author__ = 'Elvin Carvalho, 0907984 INFB1, Celsius to Kelvin'

#Function to get Kelvin from Celsius input
def CtoK():
    q1 = input('Celsius to Kelvin: ')

    #check if input abolute zero point
    if q1 < -273.15:
        print("Cant go lower (minimum -273.15)")
        CtoK()
    else:
        formula = (q1 + 273.15)
        celsius = format(formula, '.2f')
        print('%s degrees Celsius is %s degrees Kelvin' % (q1, celsius))

#If no input, show it and try again
try:
    CtoK()
except SyntaxError:
    q1 = None
    print('No input given')
    CtoK()

```

EXERCISE 1c



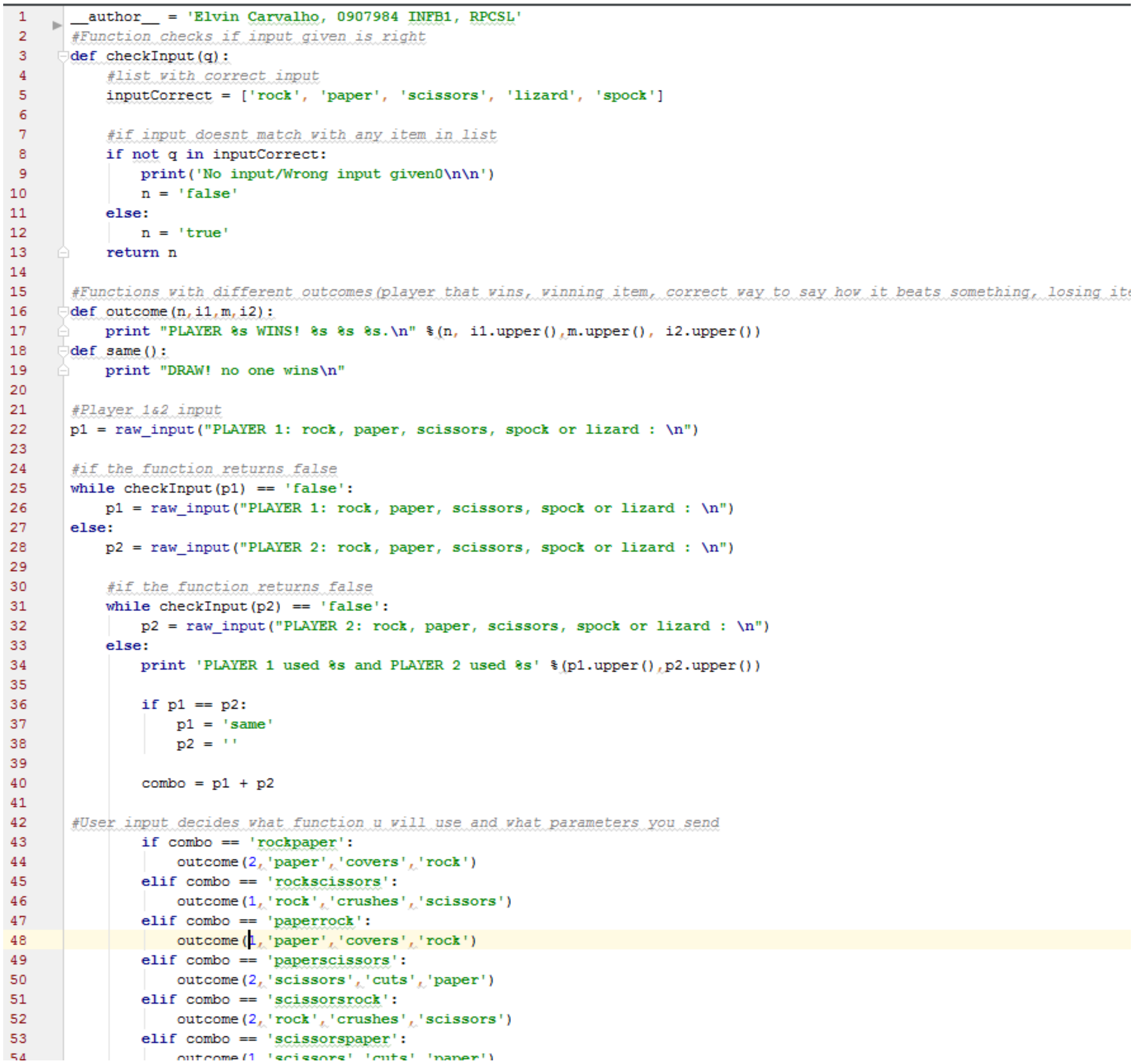
```python
__author__ = 'Elvin Carvalho, 0907984 INFB1, ABS()'

#function to calculate absolute
def absoluteN():
    q1 = input('Input number: ')
    formula = abs(q1)
    print('The absolute of %s = %s' %(q1,formula))

#check if no input is given
try:
    absoluteN()
except SyntaxError:
    q1 = None
    print('No input/Wrong input given')
    absoluteN()
```

EXERCISE 2a



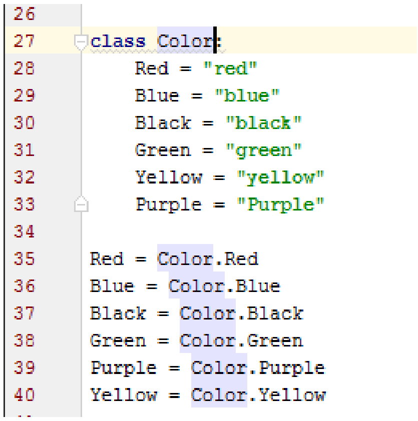
```python
__author__ = 'Elvin Carvalho, 0907984 INFB1 RPS'
#Function check if input given is right
def checkInput(q):
    #list with correct input
    inputCorrect = ['rock', 'paper', 'scissors']

    #if input doesnt match with any item in list
    if not q in inputCorrect:
        print('No input/Wrong input given0\n\n')
        n = 'false'
    else:
        n = 'true'
    return n

#Functions with different outcomes(player that wins, winning item, correct way to say how it beats something, losing item)
def outcome(n,i1,m,i2):
    print "PLAYER %s WINS! %s %s %s.\n" %(n, i1.upper(),m.upper(), i2.upper())
def same():
    print "DRAW! no one wins\n"

#Player 1&2 input
p1 = raw_input("PLAYER 1: rock, paper or scissors : \n")

while checkInput(p1) == 'false':
    p1 = raw_input("PLAYER 1: rock, paper or scissors : \n")
else:
    p2 = raw_input("PLAYER 2: rock, paper or scissors : \n")
    while checkInput(p2) == 'false':
        p2 = raw_input("PLAYER 2: rock, paper or scissors : \n")
    else:
        print 'PLAYER 1 used %s and PLAYER 2 used %s' %(p1.upper(),p2.upper())

        #if input of p1,p2 is same
        if p1 == p2:
            p1 = 'same'
            p2 = ''

        combo = p1 + p2

        #for every different outcom send different parameters to function outcome
        if combo == 'rockpaper':
            outcome(2,'paper','covers','rock')
        elif combo == 'rockscissors':
            outcome(1,'rock','crushes','scissors')
        elif combo == 'paperrock':
            outcome(1,'paper','covers','rock')
        elif combo == 'paperscissors':
            outcome(2,'scissors','cuts','paper')
        elif combo == 'scissorsrock':
            outcome(2,'rock','crushes','scissors')
        elif combo == 'scissorspaper':
            outcome(1,'scissors','cuts','paper')
        elif combo == 'same':
            same()
```

EXERCISE 2b



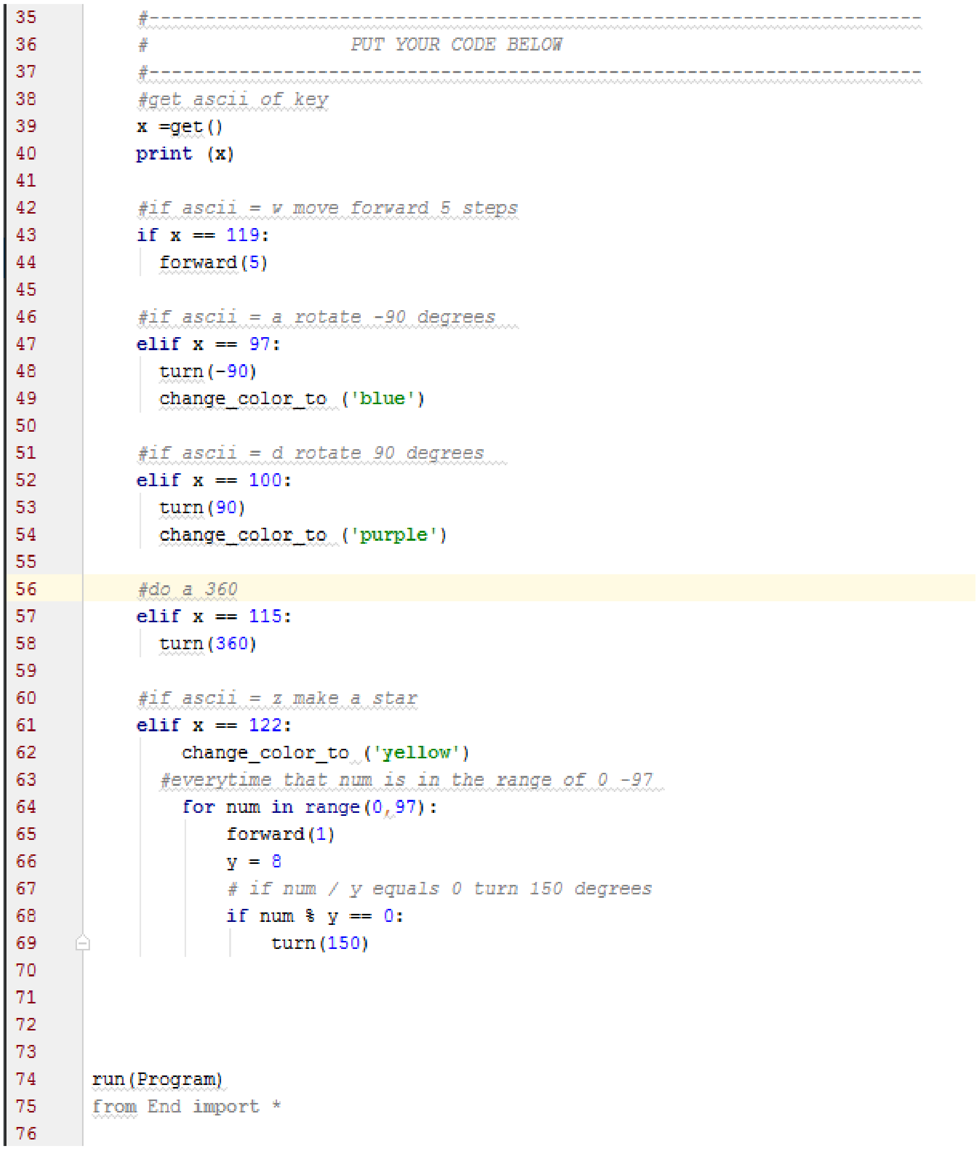
```python
__author__ = 'Elvin Carvalho, 0907984 INFB1, RPCSL'
#Function checks if input given is right
def checkInput(q):
    #list with correct input
    inputCorrect = ['rock', 'paper', 'scissors', 'lizard', 'spock']

    #if input doesnt match with any item in list
    if not q in inputCorrect:
        print('No input/Wrong input given0\n\n')
        n = 'false'
    else:
        n = 'true'
    return n

#Functions with different outcomes(player that wins, winning item, correct way to say how it beats something, losing it
def outcome(n,i1,m,i2):
    print "PLAYER %s WINS! %s %s %s.\n" %(n, i1.upper(),m.upper(), i2.upper())
def same():
    print "DRAW! no one wins\n"

#Player 1&2 input
p1 = raw_input("PLAYER 1: rock, paper, scissors, spock or lizard : \n")

#if the function returns false
while checkInput(p1) == 'false':
    p1 = raw_input("PLAYER 1: rock, paper, scissors, spock or lizard : \n")
else:
    p2 = raw_input("PLAYER 2: rock, paper, scissors, spock or lizard : \n")

    #if the function returns false
    while checkInput(p2) == 'false':
        p2 = raw_input("PLAYER 2: rock, paper, scissors, spock or lizard : \n")
    else:
        print 'PLAYER 1 used %s and PLAYER 2 used %s' %(p1.upper(),p2.upper())

        if p1 == p2:
            p1 = 'same'
            p2 = ''

        combo = p1 + p2

#User input decides what function u will use and what parameters you send
        if combo == 'rockpaper':
            outcome(2,'paper','covers','rock')
        elif combo == 'rockscissors':
            outcome(1,'rock','crushes','scissors')
        elif combo == 'paperrock':
            outcome(1,'paper','covers','rock')
        elif combo == 'paperscissors':
            outcome(2,'scissors','cuts','paper')
        elif combo == 'scissorsrock':
            outcome(2,'rock','crushes','scissors')
        elif combo == 'scissorspaper':
            outcome(1,'scissors','cuts','paper')
```

EXERCISE 3a Start.py



```python

class Color:
    Red = "red"
    Blue = "blue"
    Black = "black"
    Green = "green"
    Yellow = "yellow"
    Purple = "Purple"

Red = Color.Red
Blue = Color.Blue
Black = Color.Black
Green = Color.Green
Purple = Color.Purple
Yellow = Color.Yellow
```

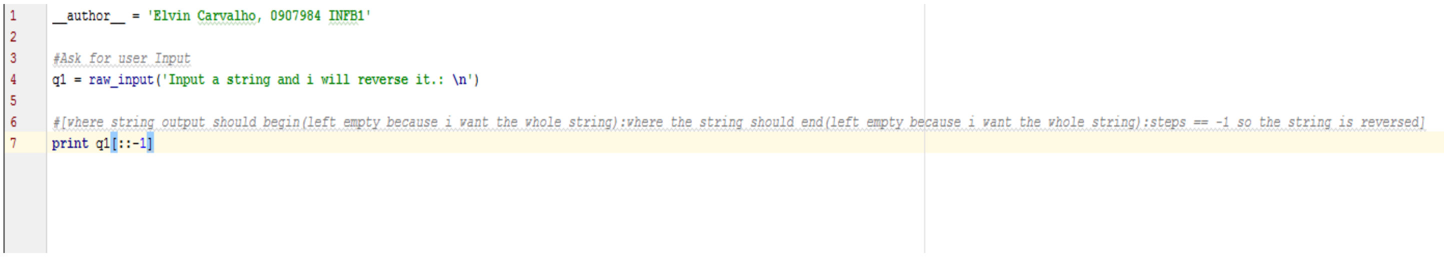
EXERCISE 3a Program.py



```python
    #-----------------------------------------------------------------------
    #                   PUT YOUR CODE BELOW
    #-----------------------------------------------------------------------
    #get ascii of key
    x =get()
    print (x)

    #if ascii = w move forward 5 steps
    if x == 119:
      forward(5)

    #if ascii = a rotate -90 degrees
    elif x == 97:
      turn(-90)
      change_color_to ('blue')

    #if ascii = d rotate 90 degrees
    elif x == 100:
      turn(90)
      change_color_to ('purple')

    #do a 360
    elif x == 115:
      turn(360)

    #if ascii = z make a star
    elif x == 122:
        change_color_to ('yellow')
      #everytime that num is in the range of 0 -97
        for num in range(0,97):
            forward(1)
            y = 8
            # if num / y equals 0 turn 150 degrees
            if num % y == 0:
                turn(150)




run(Program)
from End import *

```

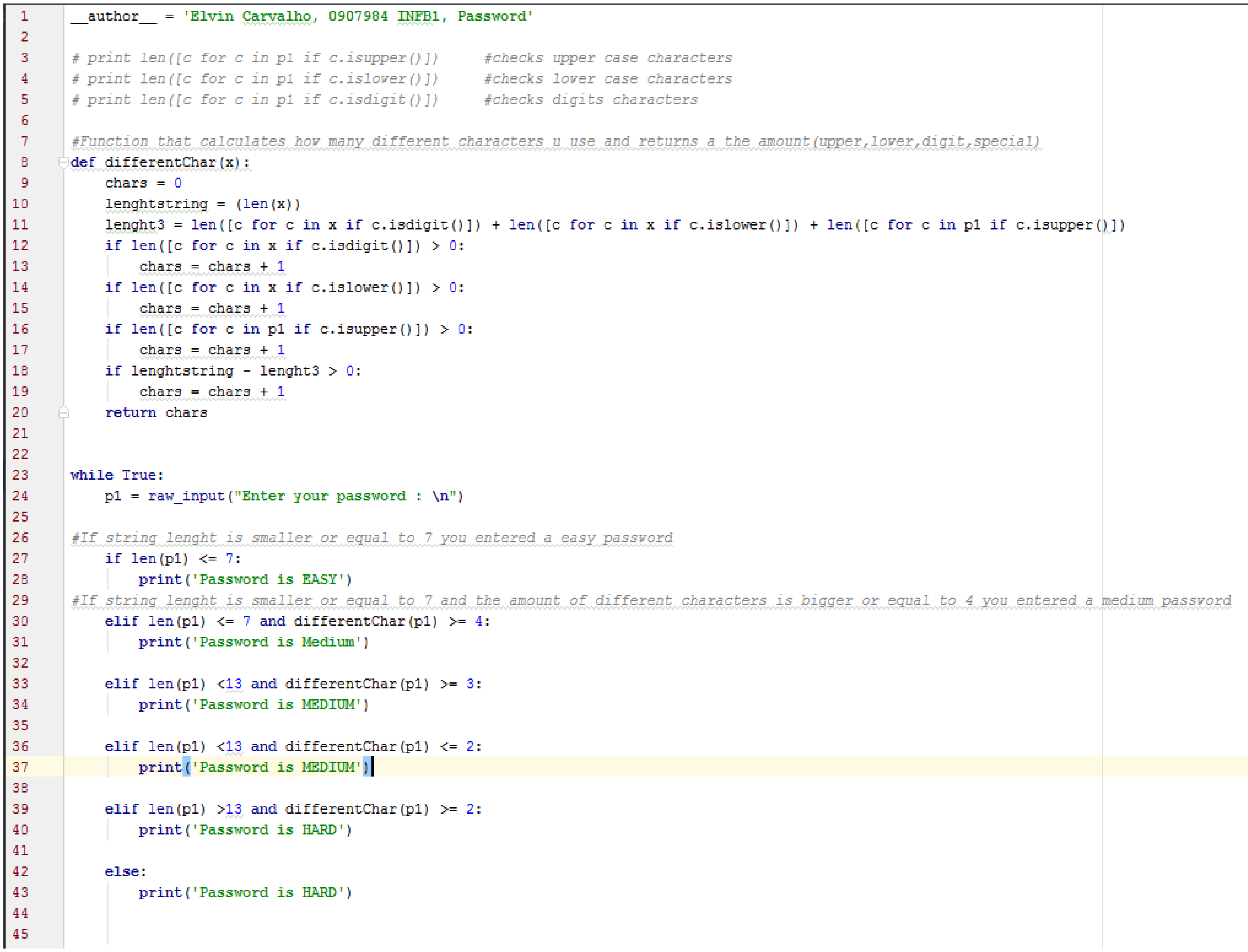
ASSIGNMENT 5

EXERCISE 1a



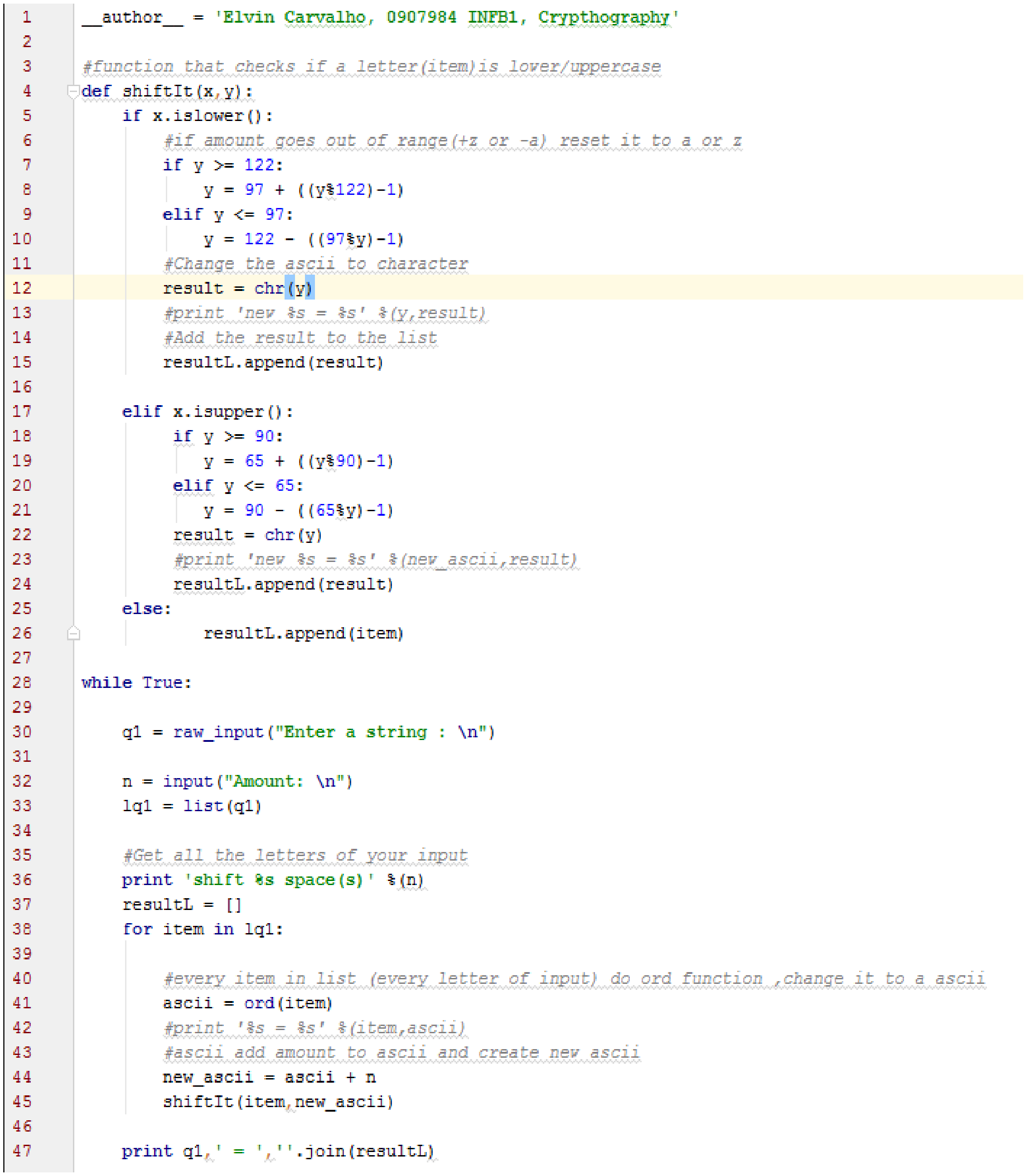
```python
__author__ = 'Elvin Carvalho, 0907984 INFB1'

#Ask for user Input
q1 = raw_input('Input a string and i will reverse it.: \n')

#[where string output should begin(left empty because i want the whole string):where the string should end(left empty because i want the whole string):steps == -1 so the string is reversed]
print q1[::-1]
```

EXERCISE 2a



```
__author__ = 'Elvin Carvalho, 0907984 INFB1, Password'

# print len([c for c in p1 if c.isupper()])     #checks upper case characters
# print len([c for c in p1 if c.islower()])     #checks lower case characters
# print len([c for c in p1 if c.isdigit()])     #checks digits characters

#Function that calculates how many different characters u use and returns a the amount(upper,lower,digit,special)
def differentChar(x):
    chars = 0
    lenghtstring = (len(x))
    lenght3 = len([c for c in x if c.isdigit()]) + len([c for c in x if c.islower()]) + len([c for c in p1 if c.isupper()])
    if len([c for c in x if c.isdigit()]) > 0:
        chars = chars + 1
    if len([c for c in x if c.islower()]) > 0:
        chars = chars + 1
    if len([c for c in p1 if c.isupper()]) > 0:
        chars = chars + 1
    if lenghtstring - lenght3 > 0:
        chars = chars + 1
    return chars


while True:
    p1 = raw_input("Enter your password : \n")

#If string lenght is smaller or equal to 7 you entered a easy password
    if len(p1) <= 7:
        print('Password is EASY')
#If string lenght is smaller or equal to 7 and the amount of different characters is bigger or equal to 4 you entered a medium password
    elif len(p1) <= 7 and differentChar(p1) >= 4:
        print('Password is Medium')

    elif len(p1) <13 and differentChar(p1) >= 3:
        print('Password is MEDIUM')

    elif len(p1) <13 and differentChar(p1) <= 2:
        print('Password is MEDIUM')

    elif len(p1) >13 and differentChar(p1) >= 2:
        print('Password is HARD')

    else:
        print('Password is HARD')
```

EXERCISE 3a



```
__author__ = 'Elvin Carvalho, 0907984 INFB1, Crypthography'

#function that checks if a letter(item)is lower/uppercase
def shiftIt(x,y):
    if x.islower():
        #if amount goes out of range(+z or -a) reset it to a or z
        if y >= 122:
            y = 97 + ((y%122)-1)
        elif y <= 97:
            y = 122 - ((97%y)-1)
        #Change the ascii to character
        result = chr(y)
        #print 'new %s = %s' %(y,result)
        #Add the result to the list
        resultL.append(result)

    elif x.isupper():
         if y >= 90:
            y = 65 + ((y%90)-1)
         elif y <= 65:
            y = 90 - ((65%y)-1)
         result = chr(y)
         #print 'new %s = %s' %(new_ascii,result)
         resultL.append(result)
    else:
            resultL.append(item)

while True:

    q1 = raw_input("Enter a string : \n")

    n = input("Amount: \n")
    lq1 = list(q1)

    #Get all the letters of your input
    print 'shift %s space(s)' %(n)
    resultL = []
    for item in lq1:

        #every item in list (every letter of input) do ord function ,change it to a ascii
        ascii = ord(item)
        #print '%s = %s' %(item,ascii)
        #ascii add amount to ascii and create new ascii
        new_ascii = ascii + n
        shiftIt(item,new_ascii)

    print q1,' = ',''.join(resultL)
```