Programming in Python
1st Homework Solutions

Karlo Knežević
karloknezevic.github.io

November 15, 2021

1 Solutions

The solutions are intentionally given in *pdf* format with the aim of writing scripts by yourself.

1.1 Task 1

```
import math

# enter parameters
#convert parameters to real numbers
a = float(input("Enter␣parameter␣a:"))
b = float(input("Enter␣parameter␣b:"))
c = float(input("Enter␣parameter␣c:"))

#calculate what is below the root
D = b * b - 4 * a * c

#if D is negative, remember it was
#negative and turn it into a positive
negative = False
if D <0:
        negative = True
        D *= -1 # convert D to a positive number

#calculate the first part of the solution
factor_1 = -b / (2 * a)
#calculate the second part of the solution
factor_2 = math.sqrt (D) / (2 * a)

#print the equation
print (f "The␣solutions␣of␣{a}␣x␣*␣x␣+␣{b}␣x␣+␣{c}␣=␣0␣are:")

#if the solution under the root is negative
if negative:
        #then it's a complex number
        #format it
        print (f "x1␣=␣{factor_1:␣.2f}␣+␣{factor_2:␣.2f}␣j")
        print (f "x2␣=␣{factor_1:␣.2f}␣-␣{factor_2:␣.2f}␣j")
else:
        #if the solution is realistic, just add the first and second
           factors
        print (f "x1␣=␣{factor_1␣+␣factor_2:␣.2f}")
```

```
        print (f "x2 = {factor_1 - factor_2:.2f}")
```

1.2 Task 2

```
input = input ("Enter a word or sentence:")

#I'm making a copy so I can later
#print the original sentence
copy = input[:]

#convert everything to lowercase
copy = copy.lower()

# throw out blanks (this is a task trick)
copy = copy.replace("", "")

#check palindrome
if copy == copy[:: - 1]:
        print (f"The entry {entry} is palindrome.")
else:
        print (f"Input is not a palindrome")
```

1.3 Task 3

```
entry = input ("Enter a word or sentence:")

even_indices = entry [:: 2]
odd_indices = entry [1 :: 2]
#starting from index 4
every_4th = entry [4 :: 4]
reverse = entry [:: - 1]

#printing characters | to see what was cut
#it is possible to cut a space so that it is
#visible
print (f"The characters on the even indices are |{even_indices}|")
print (f"The characters on the odd indices are |{odd_indices}|")
print (f"Chars in every 4th place are |{every_4th}|")
print (f"The entry entered in reverse order is |{reverse}|")
```

1.4 Task 4

```
entry = input("Enter an integer number:")

if not entry.isdecimal ():
        print ("You did not enter an integer!")
        print ("An error is likely to occur!")

#convert a string to a number
number = int(entry)

#inside curly braces can call functions: D
print (f"The least significant digit is {number % 10} or {entry [-1]}.")
print (f"The number has {len (entry)} digits.")
```

1.5 Task 5

```
#immediately load and convert types
```

```
operand_1 = float (input ("Enter␣the␣1st␣operand:"))
operator = input ("Enter␣an␣operator:")
operand_2 = float (input ("Enter␣the␣2nd␣operand:"))

if operator == "+":
        print (f"The␣sum␣{operand_1}␣+␣{operand_2}␣=␣{operand_1␣+␣
           operand_2}")
elif operator == "-":
        print (f"The␣difference␣{operand_1}␣-␣{operand_2}␣=␣{operand_1␣-
           ␣operand_2}")
elif operator == "*":
        print (f"The␣product␣{operand_1}␣*␣{operand_2}␣=␣{operand_1␣*␣
           operand_2}")
elif operator == "/":
        #division by 0
        if operand_2 == 0:
                print (f"{operand_1}␣/␣{operand_2}␣=␣ERROR")

        #if the second operand is not 0, then everything is fine
        if operand_2 != 0:
                print (f"The␣quotient␣{operand_1}␣/␣{operand_2}␣=␣{
                   operand_1␣/␣operand_2}")
else:
        print("Unknown␣operator!")
```

1.6 Task 6

```
first = float (input ("Enter␣1st␣number:"))
second = float (input ("Enter␣2nd␣number:"))
third = float (input ("Enter␣3rd␣number:"))

# suppose the first number is the smallest
smallest = first

#if the other is less than the smallest, then
# the second becomes the smallest
if second <smallest:
        smallest = second

#if the third is less than the smallest, then
#third becomes the smallest
if third <smallest:
        smallest = third

print (f"The␣smallest␣of␣{first},␣{second}␣and␣{third}␣is␣{smallest}.")
```

1.7 Task 7

```
#convert character strings immediately upon loading
#floating point number
x1 = float (input ("Enter␣x1:"))
y1 = float (input ("Enter␣y1:"))
x2 = float (input ("Enter␣x2:"))
y2 = float (input ("Enter␣y2:"))

#first of all there is a check of division by 0!
if x2 - x1 == 0:
        print (f"Direction␣cannot␣be␣determined.")
```

```
        print (f"An␣error␣is␣likely␣to␣occur!")

a = (y2 - y1) / (x2 - x1)
b = -x1 * a + y1

# I do this so that it doesn't happen to me in print + -
if b >= 0:
        print (f"The␣equation␣of␣line␣is:␣y␣=␣{a}␣x␣+␣{b}")
else:
        #if b is negative, then it prints automatically -
        print (f"The␣equation␣of␣line␣is:␣y␣=␣{a:.2f}x␣{b:.2f}")
```

1.8 Task 8

```
"""
Introductory comment: this task must first be graphically sketched
to be able to visualize what needs to be done in the printout.

For code readability, it is necessary to separate the calculation and
display on the screen.
"""

#user entry gross1 salary
gross1 = float (input ("Enter␣gross1␣amount␣[kn]:"))

"""
These are fixed amounts determined by law
Of the Republic of Croatia and laws
local self-government for the City of Zagreb.

It is possible to change these amounts.
"""
mio1_per = 0.15 # 1. pension pillar
mio2_per = 0.05 # 2. pension pillar
tax = 0.24 #tax in Croatia
surtax_zagreb = 0.18 #surtax in the City of Zagreb
tax_free_amount = 4000.00 #non-taxable amount without tax relief
health_insr = 0.165 #rate gross1 salary for health insurance


mio1 = mio1_per * gross1
mio2 = mio2_per * gross1

X1 = gross1 - mio1 - mio2

X2 = X1 - tax_free_amount

X3 = tax * X2

X4 = surtax_zagreb * X3

net = gross1 - mio1 - mio2 - X3 - X4

health = health_insr * gross1

gross2 = gross1 + health

# payroll width (this is subject to change)
```

```
width = 90

#number of columns in the payroll
columns = 6
#width of one column
#| -w- | -w- | -w- | -w- | -w- | -w- |
w = width // columns

#printing the title
print ("PAYROLL" .center (width, "_"))

#notice: the sum of the ws in a row must be 6 because it is 6 columns

#printing gross salary
print (f"{'':{w␣*␣5}}{'Gross':^{w}}")
print (f"{'':{w␣*␣5}}{gross1:>{w}.2f}")

#printing pension contributions
print (f"{'Contributions␣from':<{w*3}s}{'Rate':^{w}s}{'Base':^{w}s}{'
    Amount':^{w}s}")
print (f"{'MIO1':<{w}s}{'MIO␣I␣pension␣fund':<{w*2}s}{mio1_per␣*␣100:>{w
    }.2f}{gross1:>{w}.2f}{mio1:>{w}.2f}")
print (f"{'MIO2':<{w}s}{'MIO␣II␣pension␣fund':<{w*2}s}{mio2_per␣*␣100:>{
    w}.2f}{gross1:>{w}.2f}{mio2:>{w}.2f}")
print (f"{'':{w*5}}{mio1␣+␣mio2:>{w}.2f}")

#printing taxes
print (f"{'Tax':<{w␣*␣2}s}{'Deduction':^{w}s}{'Rate':^{w}s}{'Base':^{w}s
    }{'Amount':^{w}s}")
print (f"{'TAX':<{w}s}{'Tax':<{w}s}{tax_free_amount:>{w}.2f}{tax␣*␣
    100:>{w}.2f}{X2:>{w}.2f}{X3:>{w}.2f}")

#printing surtax
print (f"{'Surtax':<{w␣*␣3}s}{'Rate':^{w}s}{'Base':^{w}s}{'Amount':^{w}s
    }")
print (f"{'SURT':<{w}s}{'Surtax␣-␣City␣of␣Zagreb':<{w␣*␣2}s}{
    surtax_zagreb␣*␣100:>{w}.2f}{X3:>{w␣}.2f}{X4:>{w}.2f}␣")

#print of net salary
print (f"{'':{w␣*␣5}}{'Net':^{w}}")
print (f"{'':{w␣*␣5}}{net:>{w}.2f}")

#print of health contributions
print (f"{'Contributions␣to':<{w␣*␣3}s}{'Rate':^{w}s}{'Base':^{w}s}{'
    Amount':^{w}s}")
print (f"{'HEALTH':<{w}s}{'Contribute␣to␣health␣insurance':<{w␣*␣2}s}{
    health_insr␣*␣100:>{w}.2f}{gross1:>{w}.2f}{health:>{w}.2f}")
print (f"{'':{w␣*␣5}}{0.165␣*␣gross1:>{w}.2f}")

#print of total salary cost
print (f"{'':{w␣*␣5}}{'Salary␣cost':^{w}}")
print (f"{'':{w␣*␣5}}{gross2:>{w}.2f}")
```