Introduction to Programming with Python

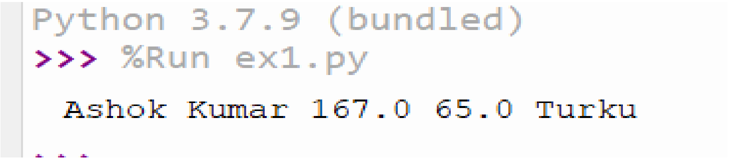
Weekly Programming Assignment – Week 6

All solution files must be submitted at CodeGrade enabled Link for grading.

All solutions must be uploaded on or before 20th October 2021 at 11:59 PM

Exercise 1

Define a procedure called ***myInfo()*** that takes filename as input and display its contents. Should not use **with open()**. Then Write a main program that creates a new file called ***"myInfo.txt"*** and write your information including, name, height, weight, and study campus in that file as one line with single space. Invoke *myInfo()* procedure to print **"myInfo.txt"** contents. The sample run is here:

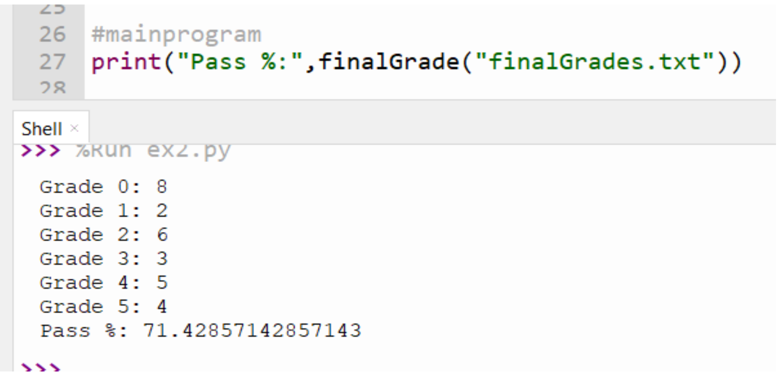


```
Python 3.7.9 (bundled)
>>> %Run ex1.py

 Ashok Kumar 167.0 65.0 Turku
```

Exercise 2

The file ***"finalGrades.txt"*** contains course grades received by all students for the course "Introduction to Programming". Define a function **finalGrade()** that reads all grades stored in that file and prints the total number of students that received 0, 1, 2, 3, 4, 5 grades respectively. Then it returns the total number of students that successfully completed(pass) the course (0 means fail) in percentage. The text file is attached. The final run is here.

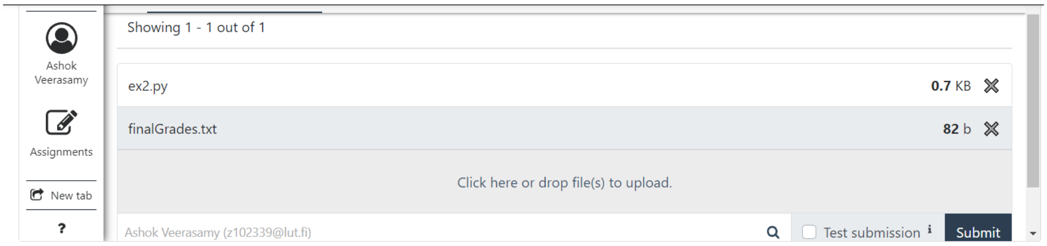


```
#mainprogram
print("Pass %:",finalGrade("finalGrades.txt"))
```

```
>>> %Run ex2.py

 Grade  0:  8
 Grade  1:  2
 Grade  2:  6
 Grade  3:  3
 Grade  4:  5
 Grade  5:  4
 Pass %: 71.42857142857143
```

Submit both your code (.py) and finalGrades.txt files. **Don't edit** the contents written in **"finalGrades.txt"**. Sample screen shot for submission procedure given here.


Ashok Veerasamy
Assignments
New tab
?
Click here or drop file(s) to upload.
Ashok Veerasamy (z102339@lut.fi)
□ Test submission
Submit

Showing 1 - 1 out of 1

ex2.py 0.7 KB

finalGrades.txt 82 b

Exercise 3

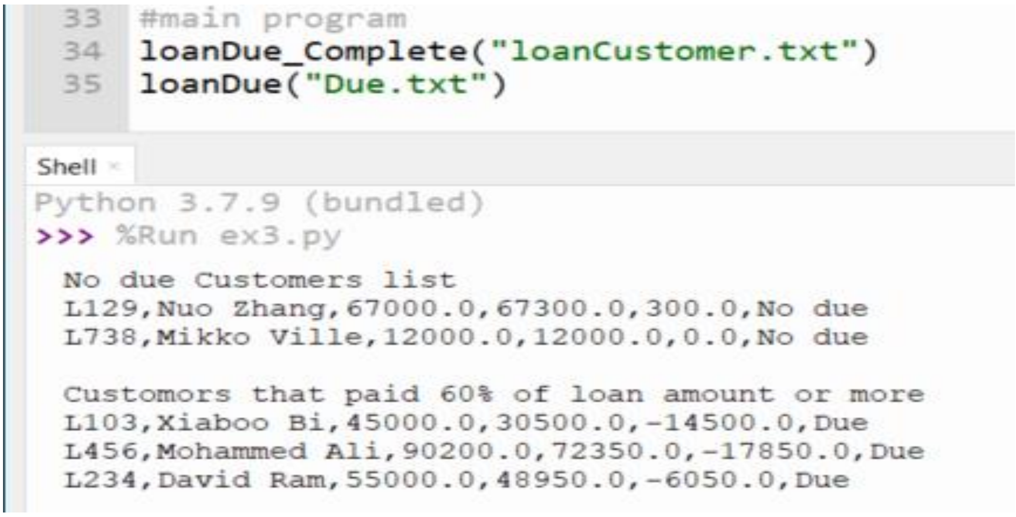
The *"loanCustomer.txt"* file contains customer id, name, loan amount granted, amount paid by customer. Refer The file is attached with the assignment [don't edit the contents of this file- use it for reading only]

Create two subprograms namely:

i. ***loanDue_Complete()*** that takes file name as a parameter and write customer information that paid loan amount fully including id, name, amount granted, amount paid, balance (should be computed- 0 or excess), and status ("No due"). The no due customer data must be written into a new file called ***"noDue.txt".*** On the other hand, the information about customer that have not fully paid should be written in another new file called **"Due.txt".** Then print only the "No due" customer details [***noDue.txt]***. [refer the expected output given below]

ii. ***loanDue()*** that takes file name as a parameter which reads customer details from **"Due.txt"** and displays the customers that paid >=60% of loan amount.

Test your created subprograms with the main program given here. Submit code (.py) and *loanCustomer.txt* files only. Don't submit other text files.



```
#main program
loanDue_Complete("loanCustomer.txt")
loanDue("Due.txt")
```

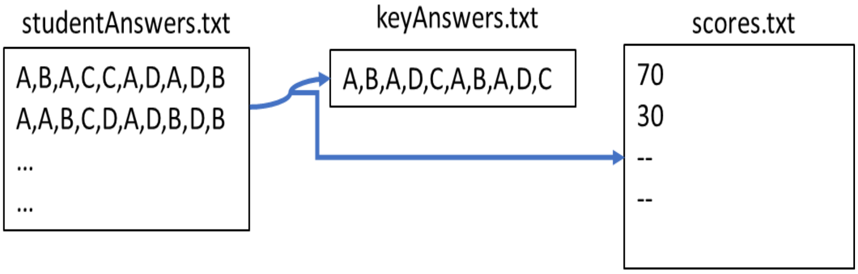
```
Python 3.7.9 (bundled)
>>> %Run ex3.py

 No due Customers list
 L129,Nuo Zhang,67000.0,67300.0,300.0,No due
 L738,Mikko Ville,12000.0,12000.0,0.0,No due

 Customors that paid 60% of loan amount or more
 L103,Xiaboo Bi,45000.0,30500.0,-14500.0,Due
 L456,Mohammed Ali,90200.0,72350.0,-17850.0,Due
 L234,David Ram,55000.0,48950.0,-6050.0,Due
```

Exercise 4

The students' answers for multiple choice exam are stored in a file called ***"studentAnswers.txt"***. The key answer for each question is stored in another file called ***"keyAnswers.txt"***. Do not modify the contents of these text files. Write code that evaluates student answers by using that key answer text given in another file. Then write the final score computed for each student in a new file called ***"scores.txt"***. Then print the final score by invoking the procedure called ***"printScores()"*** at main program. Check the example given below. Each row represents a student. In total 10 questions were given for the exam.



studentAnswers.txt

```
A,B,A,C,C,A,D,A,D,B
A,A,B,C,D,A,D,B,D,B
...
...
```

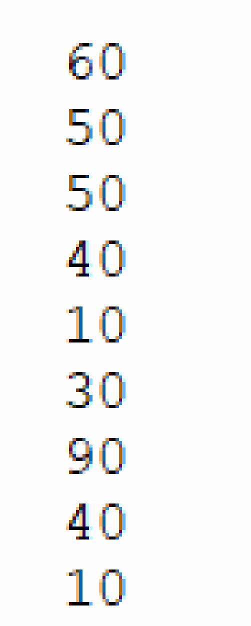
keyAnswers.txt

```
A,B,A,D,C,A,B,A,D,C
```

scores.txt

```
70
30
--
--
```

The expected output will be:

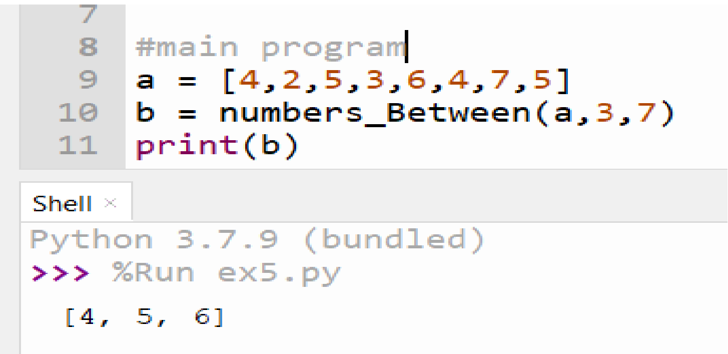


```
60
50
50
40
10
30
90
40
10
```

Test your created subprogram with the main program given above. Submit your code (.py), ***"studentAnswers.txt"***, and ***"keyAnswers.txt"*** files only. Don't submit ***"scores.txt"*** file.

Exercise 5 : Self-study exercise: Lists

Write a function ***numbers_Between(list1, x, y)*** that returns a new list which contains values that are bigger than x and smaller than y. [Assume x is always smaller than y]. Use the main program given below to test your function.



```
#main program
a = [4,2,5,3,6,4,7,5]
b = numbers_Between(a,3,7)
print(b)
```

```
Python 3.7.9 (bundled)
>>> %Run ex5.py
  [4, 5, 6]
```

Exercise / task Number	Codegrade link_Moodle for file solution files upload	Points / Marks
1	Exercise1_Week 6	10
2	Exercise2_Week 6 + *finalGrades.txt*	10
3	Exercise3_Week 6 + *loanCustomer.txt*	20
4	Exercise4_Week 6 + studentAnswers.txt + keyAnswers.txt	20
5	Exercise5_Week 6	10