Python and PostgreSQL Exercises

DESCRIPTION: Homework assessments students’ knowledge in creating a Python program to create and manipulate for and while loop.

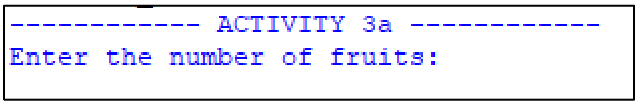
GRADING: 10 points each activity

STEPS:

1. Open a new file and save it as: *python_postgresql.py*
2. In the file, type the first two lines with comments as:
   ```
   """
   Python and Postgresql Exercises
   Student's Name:
   """
   ```
3. Complete activity from **a** to **e**

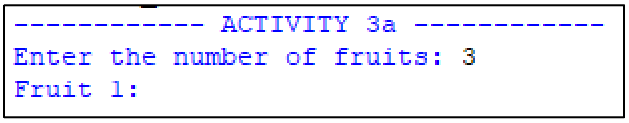
a) for loop practice: write a python program that will collect items from the user and save them into a list. The program starts asking the user for the number of items, uses a **for** loop to collect each item, and print the results of the for loop, list. *Hint: use method **`append()`** to add the items to the list*

<u>*Template*</u>



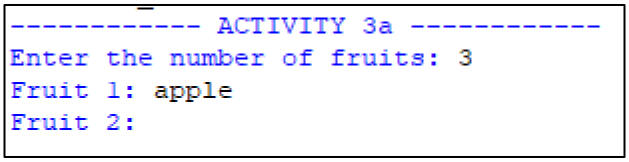
```
------------ ACTIVITY 3a ------------
Enter the number of fruits:
```

When runs the python file, it prompts a message asking the user to enter the number of elements. For example, if user types 3 and presses Enter…



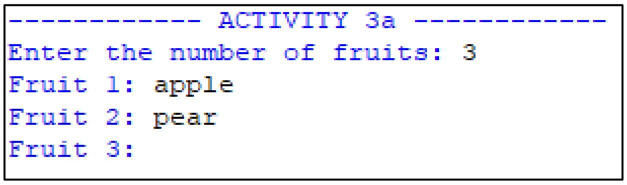
```
------------ ACTIVITY 3a ------------
Enter the number of fruits: 3
Fruit 1:
```

the program asks the user to enter a value of the first item…

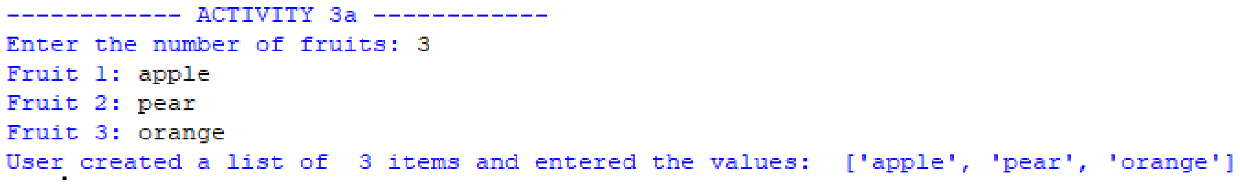


```
------------ ACTIVITY 3a ------------
Enter the number of fruits: 3
Fruit 1: apple
Fruit 2:
```

for example, if the user types and enters *apple*, the program asks the user to enter a value of the second item…



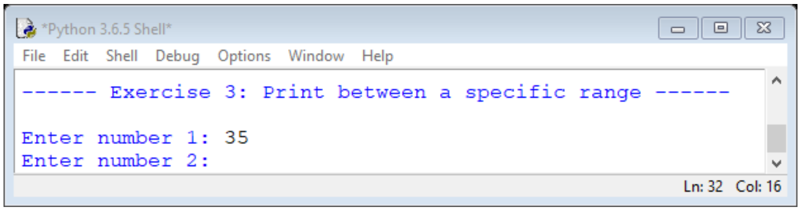
```
------------ ACTIVITY 3a ------------
Enter the number of fruits: 3
Fruit 1: apple
Fruit 2: pear
Fruit 3:
```



```
------------ ACTIVITY 3a ------------
Enter the number of fruits: 3
Fruit 1: apple
Fruit 2: pear
Fruit 3: orange
User created a list of  3 items and entered the values:  ['apple', 'pear', 'orange']
```

Activity b) Nesting **while** loop and **if-else** statement: write a Python program that uses while loop to print all the numbers (between two numbers exclusive). The two numbers are entered by the user. The program should print from the smaller number of the two entered numbers, and with an increment of 1 up to the higher number.

Template

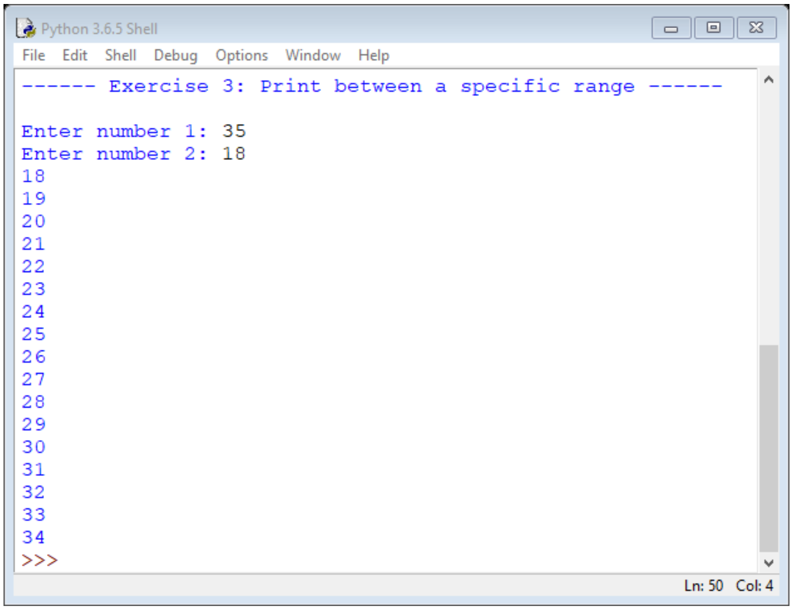


```
------ Exercise 3: Print between a specific range ------

Enter number 1: 35
Enter number 2:
```

When run the program, it asks the user to enter a number.

For example, if user types 35 and presses Enter…



```
------ Exercise 3: Print between a specific range ------

Enter number 1: 35
Enter number 2: 18
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
>>>
```

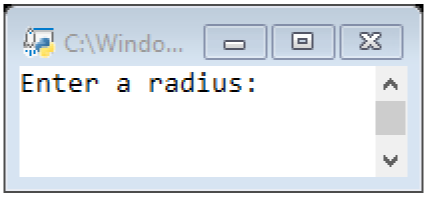
… the program will ask for the second number. For example, if user types 18 and presses Enter…

The program will print from the smaller number and print up to the greater number with an increment.

Checkpoint: If the user enters the same number, the program should ask the user to enter another number for the last number, which is the second number. Once the two numbers are different, the program will print from the smaller to the higher number with an increment of 1.

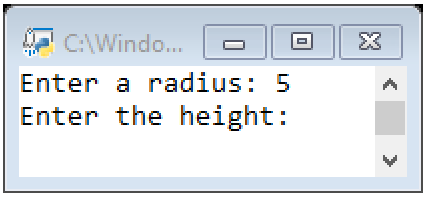
Activity c) Defined and math function: write a function called **`volumeCylinder(h,r)`** that takes two arguments for height and radius, calculates the volume of a cylinder, and returns the volume rounded off to two decimal places ($volume\ of\ cylinder: radius^2 * \pi * height$). The radius and height are entered by the user.

Template



```
Enter a radius:
```

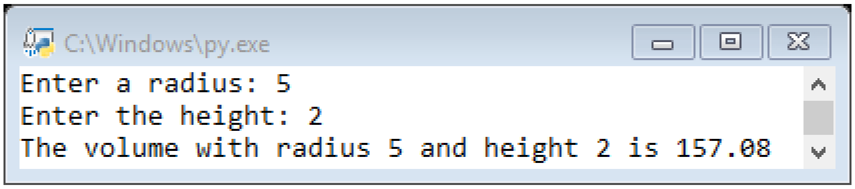
When the program is ran, it asks the user to enter a radius…



```
Enter a radius: 5
Enter the height:
```

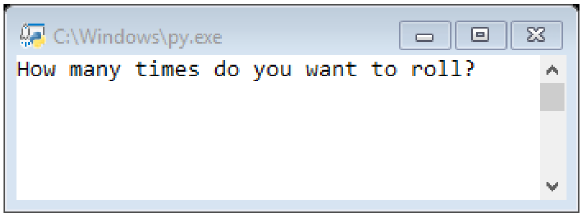
if the user types and enters 5, the program asks for the height…

If the user types and enters 2, the program displays a message showing the radius, height, and volume as:



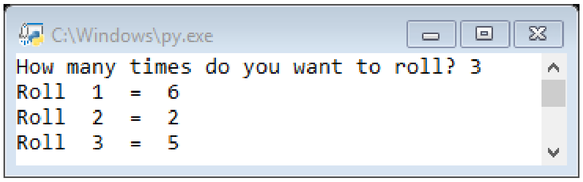
```
Enter a radius: 5
Enter the height: 2
The volume with radius 5 and height 2 is 157.08
```

Activity d) Write a function that simulate roll the dice. The program takes a number of roll from the user as an argument. Within the function, for each round should display the random number between 1 and 6. *Hint: use random number to randomly select a number between 1 and 6*



```
How many times do you want to roll?
```

When the program is ran, it asks the user to enter the number of rolls…



```
How many times do you want to roll? 3
Roll  1  =  6
Roll  2  =  2
Roll  3  =  5
```

If the user enters 3, the program shows three rolls with a random number between 1 and 6.

Note: submit the Python code with activity a,b,c, and d.

Activity e) Log-in to your **postgresql** account, go to **demoDB** database and create a table called **customers** with the following columns:

- **id** set as SERIAL PRIMARY KEY
- **firstName** set as VARCHAR(120)
- **lastName** set as VARCHAR(120)
- **email** set as VARCHAR(120) UNIQUE
- **total** set as NUMERIC

Note: submit a code of the SQL code and the screenshot of table **customers.**