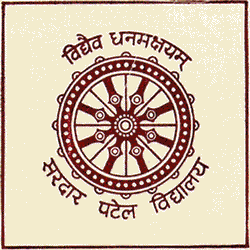
COMPUTER SCIENCE PRACTICAL FILE



Submitted by: Vinayak Bector

Class: XII D

Roll No: 33

Session:

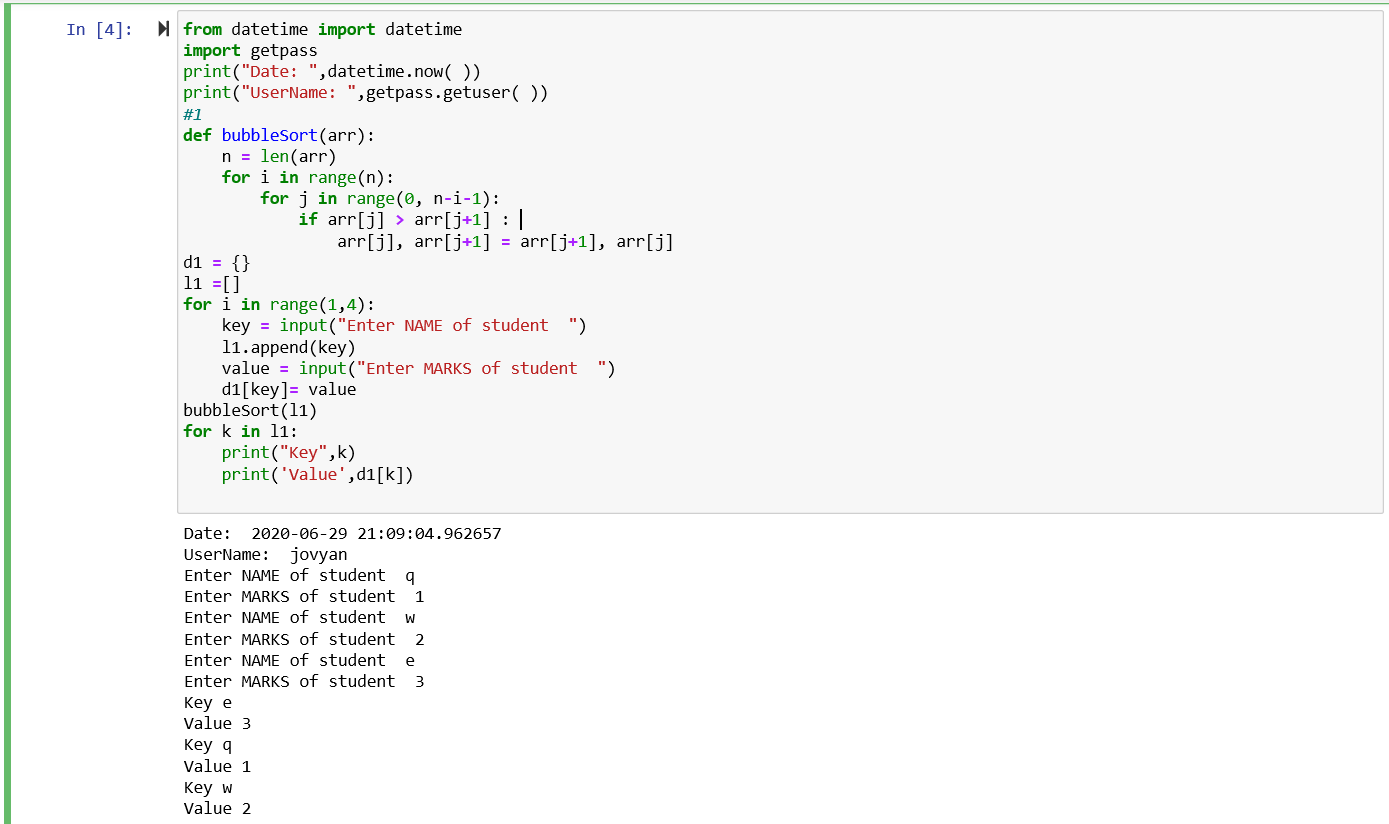
School:

**INDEX**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S. No.** | **Name of Program** | **Page No.** | **Date** | **Teacher’s Sign** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
| **S. No.** | **Name of Program** | **Page No.** | **Date** | **Teacher’s Sign** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**TERM 1**

**P1. Take names and marks of 5 students and save as key: value pairs in a dictionary RESULT. WAP that prints the dictionary contents in ascending order of marks. (Do not use built in methods)**

****

**P2. WAP to print the following pattern:**

****

**P3. WAP to generate a 3 X 4 X 6 3D array whose each element is \*.**

****

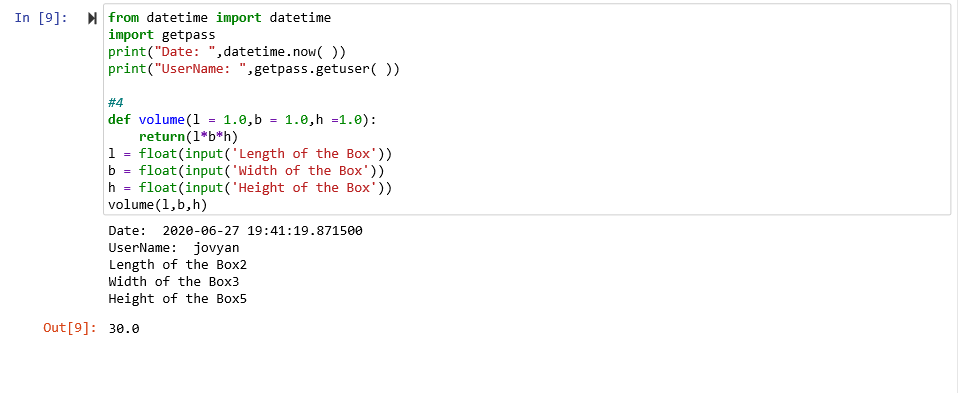
**Functions**

**P4. Write a function to calculate volume of a box with appropriate default values for its parameters. Your function should have the following input parameters:**

**a) Length of box**

**b) Width of box**

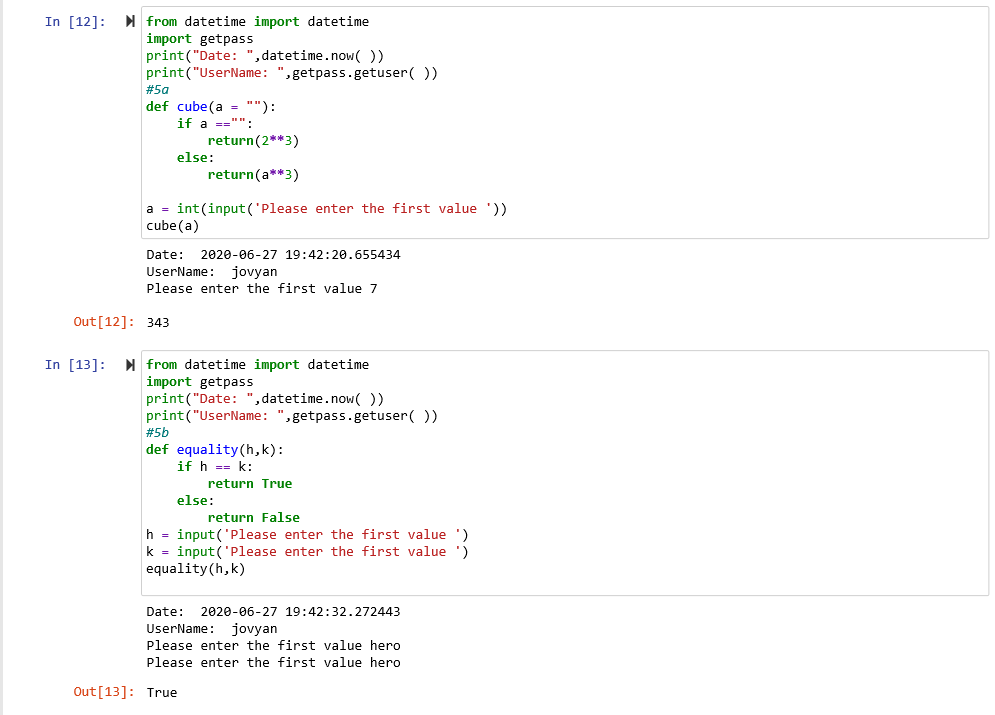
**c) Height of box Test it by writing complete program to invoke it.**

****

**P5. Write a program to have the following functions: a) A function that takes a number as argument and calculates cube for it. The function does not return a value. If there is no value passed to the function in function call, the function should calculate cube of 2**

**b) A function that takes two char arguments and returns True if both the arguments are equal otherwise false.**

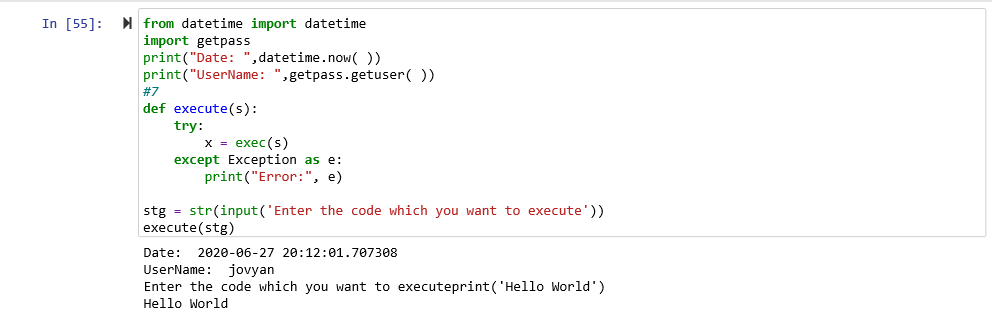
**Test both these functions by giving appropriate function call statements.**

****

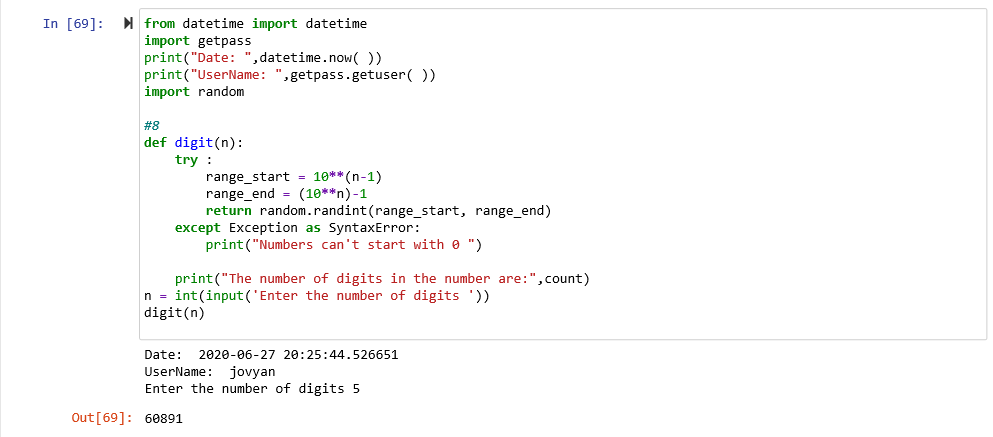
**P6. Write a Python function to check whether a number is perfect or not (number to be tested as parameter). Test the function by giving appropriate function call statements.**

****

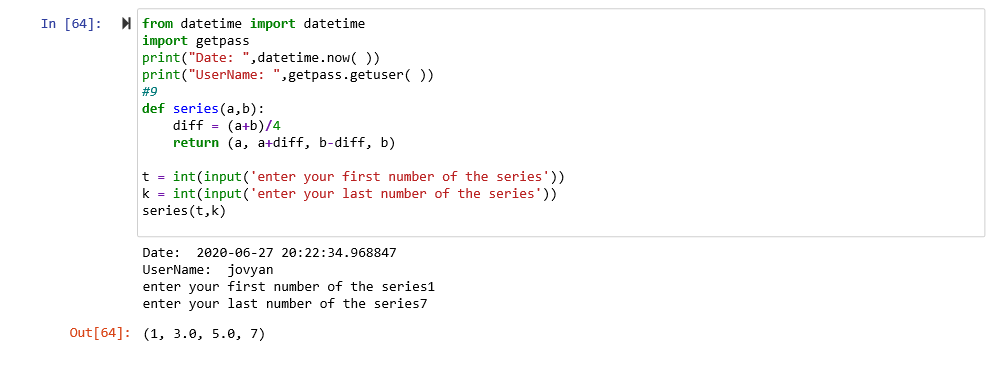
**P7. Write a Python program to execute a string containing Python code.**

****

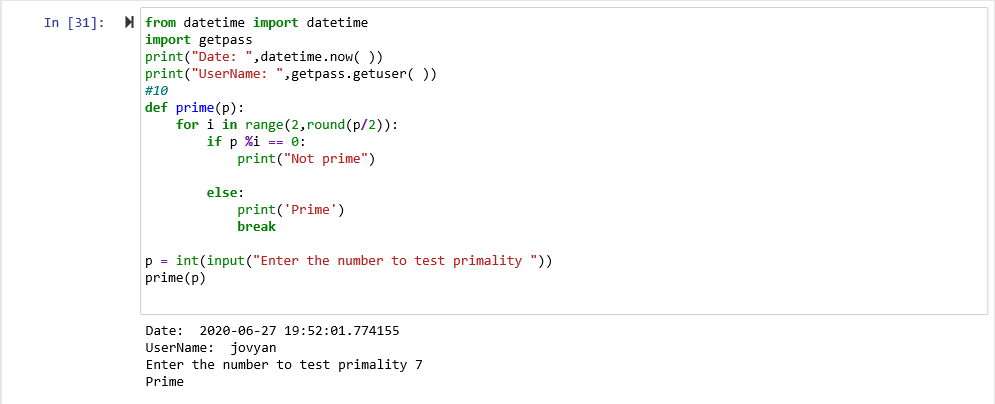
**P8. Write a function that takes a number n and then returns a randomly generated number having exactly n digits(not starting with zero) e.g., if n is 2 then function can randomly return a number 10 – 99 but 07, 02 are not valid two digit numbers.**

****

**P9. Write a program that generates a series using a function which takes first and last values of the series and then generates four terms that are equidistant .g., if two numbers passed are 1 and 7 then function returns 1 3 5 7.**

****

**P10. Write a function that takes one argument (a positive integer) and reports if the argument is prime or not. Write a program that invokes this function.**

****

**Recursion**

**P11. Do program 10 using recursive technique.**

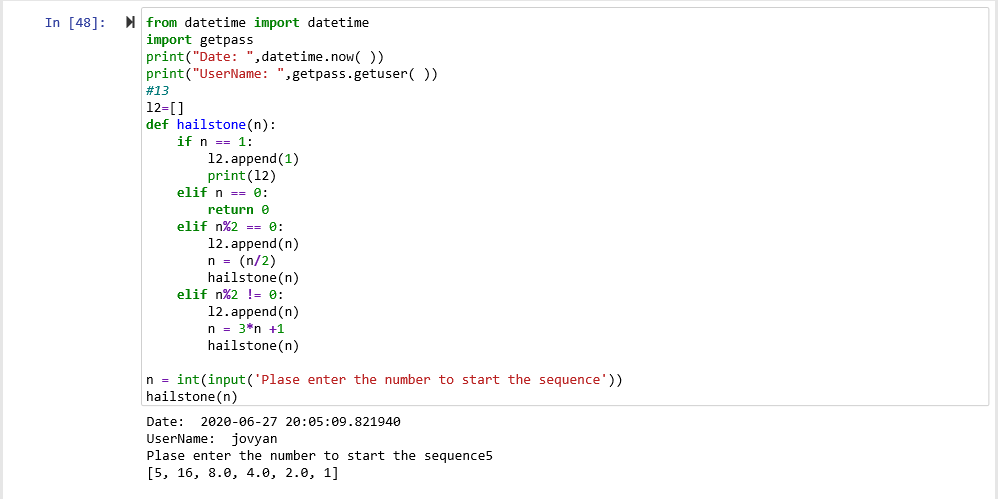
****

**P12. Implement a function product( ) to multiply 2 positive numbers recursively using addition. **

**P13. Write recursive code to compute GCD of two numbers.**

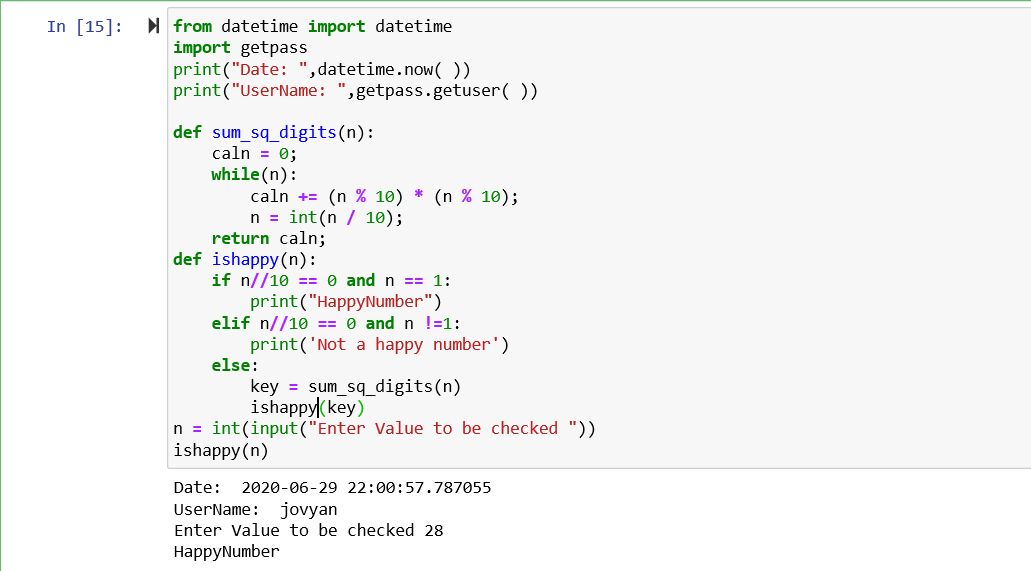
****

**P14. The hailstone sequence starting at a positive integer n is generated by following 2 simple rules. If n is even, the next number in the sequence is n/2. If n is odd, the next number in the sequence is 3\*n + 1. Repeating this process, the hailstone sequence gets generated. Write a recursive function hailstone(n) which prints the hailstone sequence beginning at n. Stop when the sequence reaches number 1**.

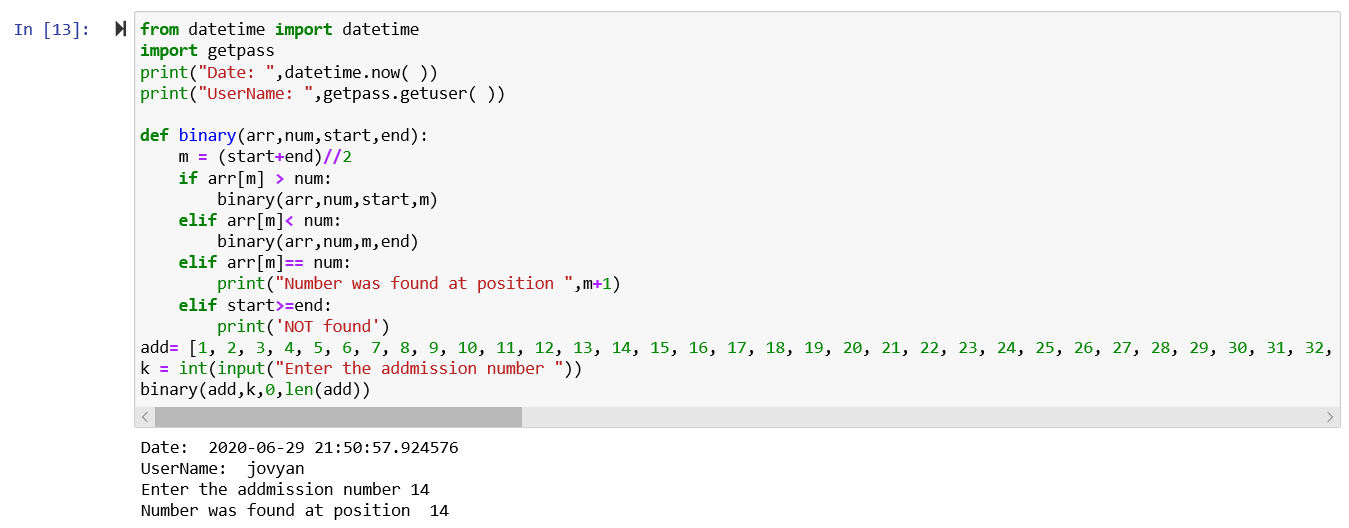


**P15. WAP that takes a number and checks if it is a happy number by using the following two functions in it:**

**sum\_sq\_digits(x) : returns the sum of the square of the digits of number x using recursion ishappy( ) : checks if the number is happy number by calling the function sum\_sq\_digits and displays an appropriate message.**

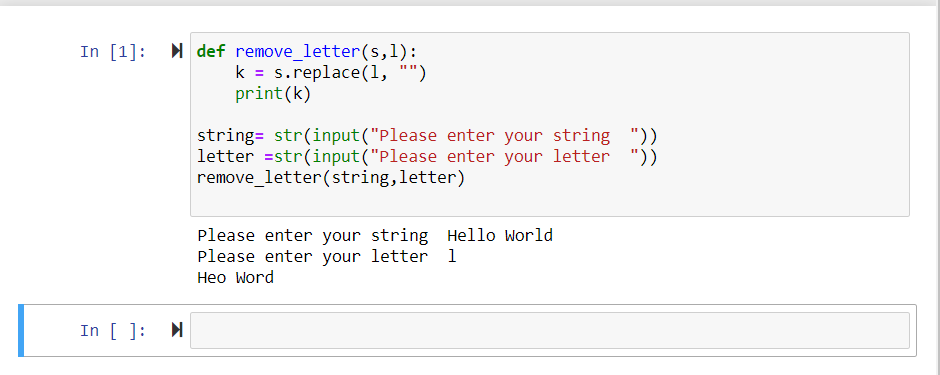
****

**P16. A list namely Adm stores admission numbers of 100 students in it, sorted in ascending order of admission numbers (integers). WAP that takes an admission number and looks for it in list Adm using recursive binary search technique.**

****

**Using Python Libraries**

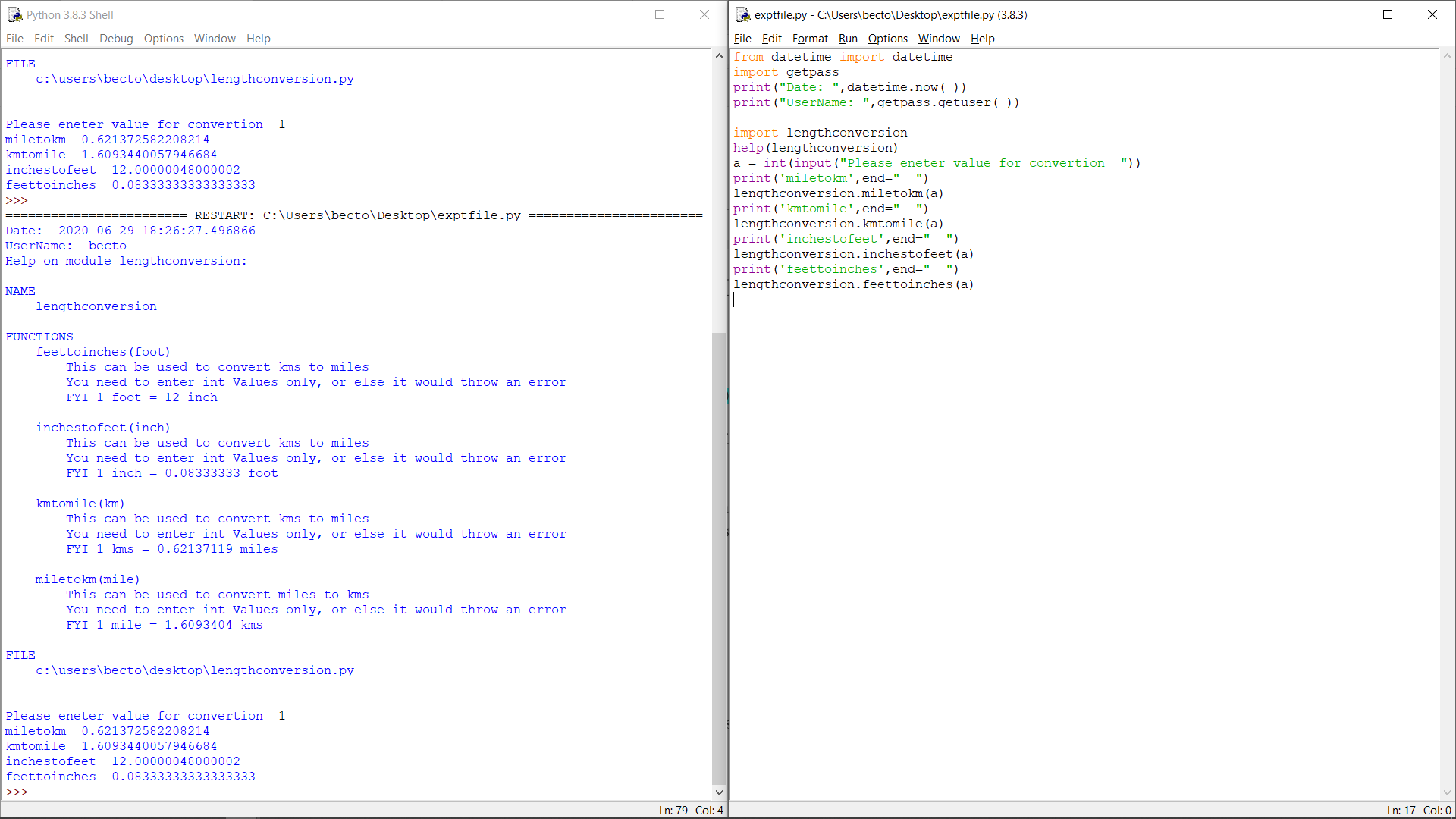
**P17. Write a program with following function: remove\_letter(sentence, letter) This function returns a copy of the above string with every instance of the indicated letter removed. E.g. , remove\_letter(“Welcome Everyone”, “e”) should return the string “Wlcom vryon”.**

****

**P18. Create a module lengthconversion.py that stores functions for various length conversions:**

**miletokm( ) kmtomile( ) feettoinches( ) inchestofeet( )**

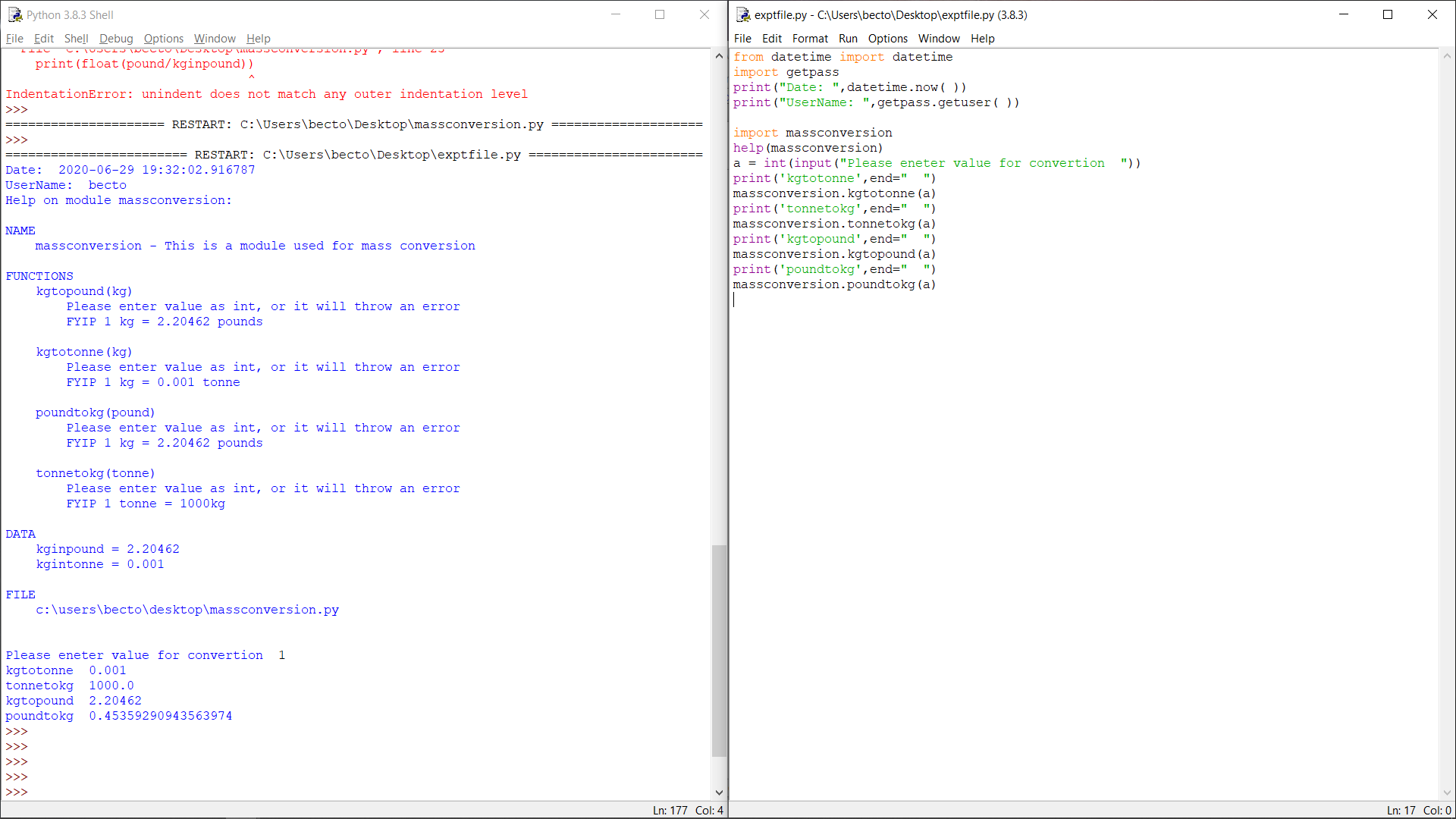
**Also store the constants – mileinkm = 1.609344, feetininches = 12. help( ) method should display proper information (to be shown as output)**

****

**P19. Create a module massconversion.py that stores functions for various mass conversions:**

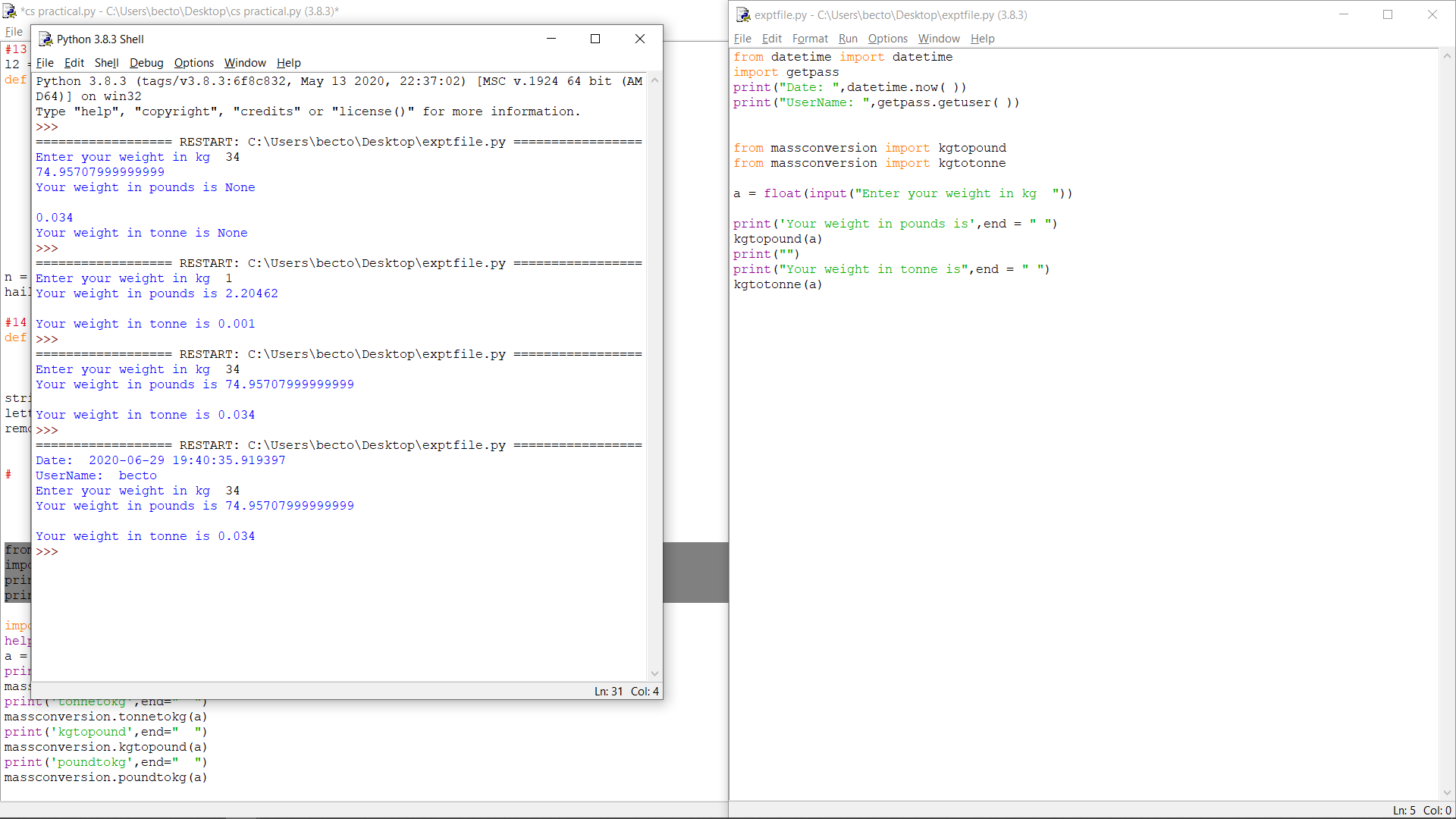
**kgtotonne( ) tonnetokg( ) kgtopound( ) poundtokg( )**

**Also store the constants – kgintonne = 0.001, kginpound = 2.20462. help( ) method should display proper information (to be shown as output)**

****

**P20. Create a package from the above two modules as this:**

**Make sure that above packages meet the requirements of being a Python Package. Access at least two methods of lengthconversion.py using import statement. Access at least two methods of massconversion.py using from-import statement.**

****

**Interface Python with MySQL**

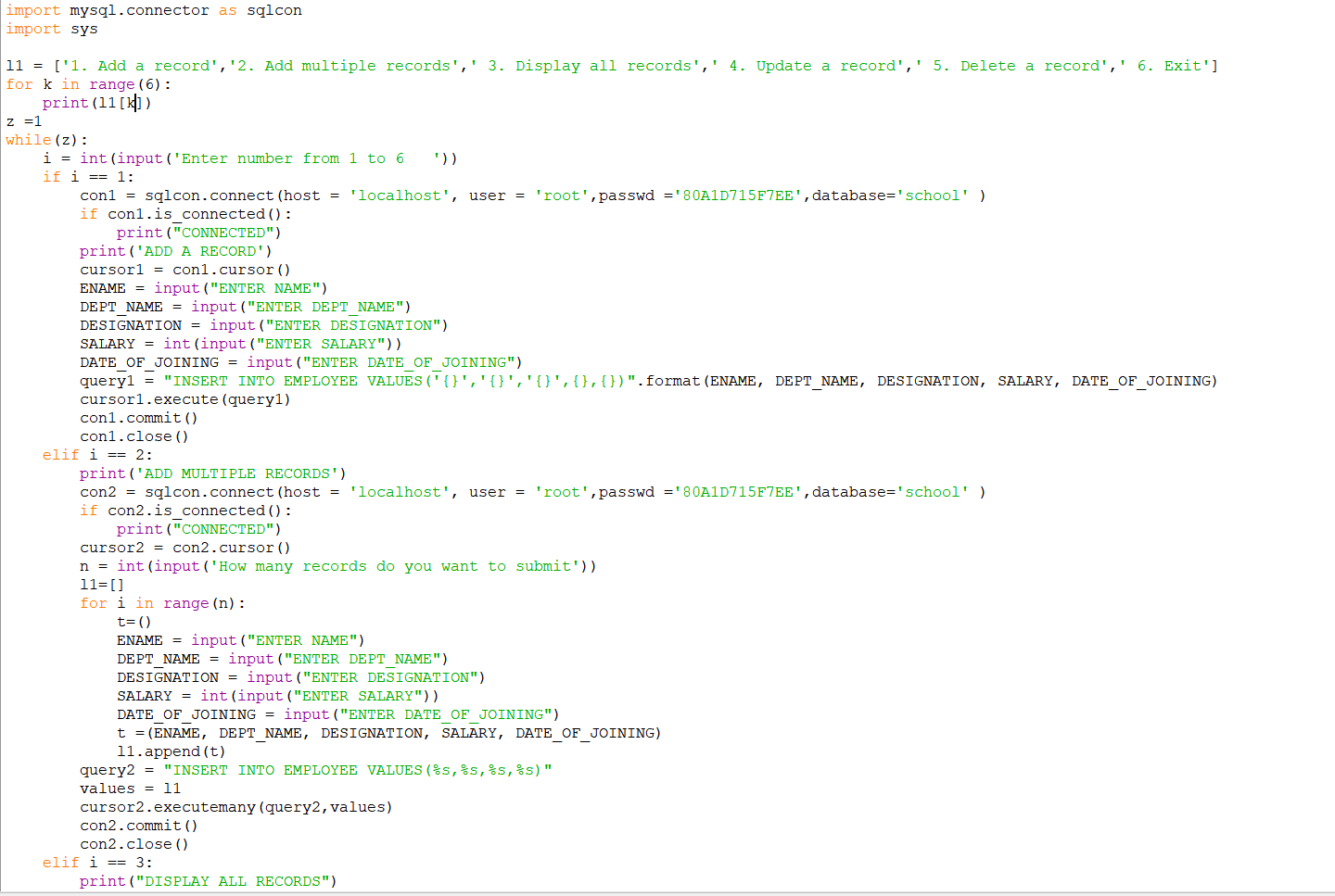
**P21. Write a menu-driven program to do the following tasks: 1. Add a record 2. Add multiple records 3. Display all records 4. Update a record 5. Delete a record 6. Exit**

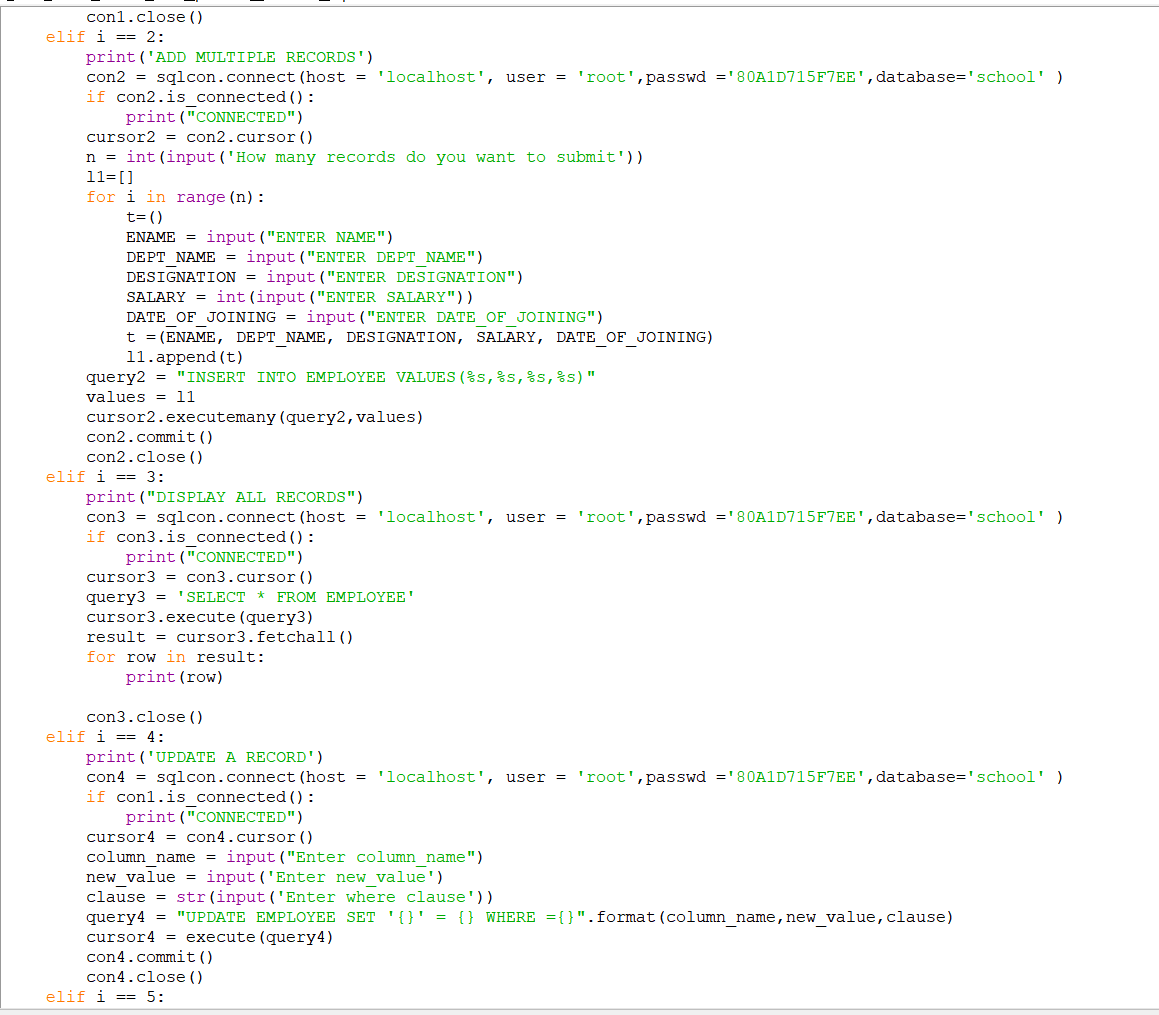
**Instructions for the above program:**

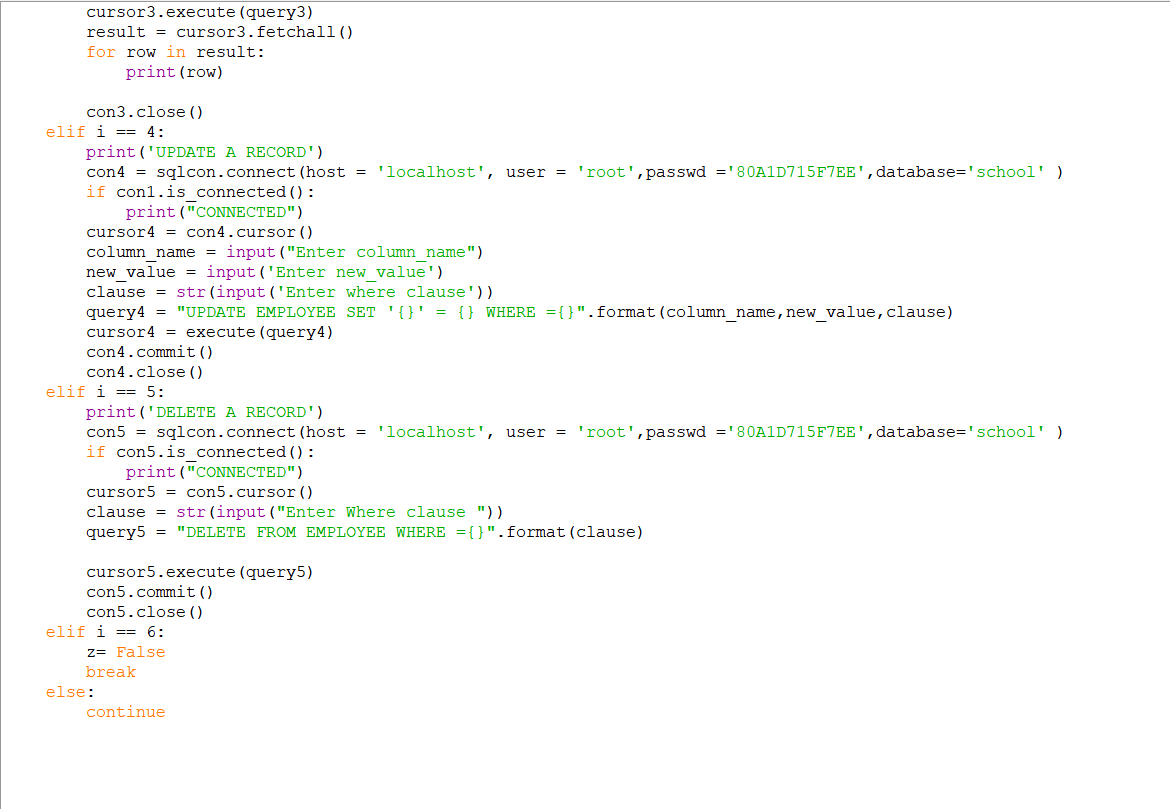
**a) The program should ask the user for the choice number (1 to 6) every performed operation. The program should display the menu continually till the user enters ‘no’ as choice.**

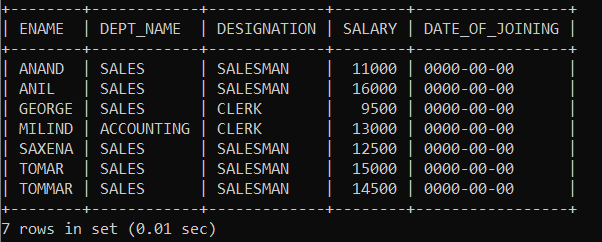
**b) In every operation, the respective function should be called. So there should be 5 functions in total, each function should include creating a connection with database, performing the respective operation and closing the connection. c) For all the operations, input should be taken from the user. d) In update, insert and delete operations, you must use commit( ) and rollback( ).**

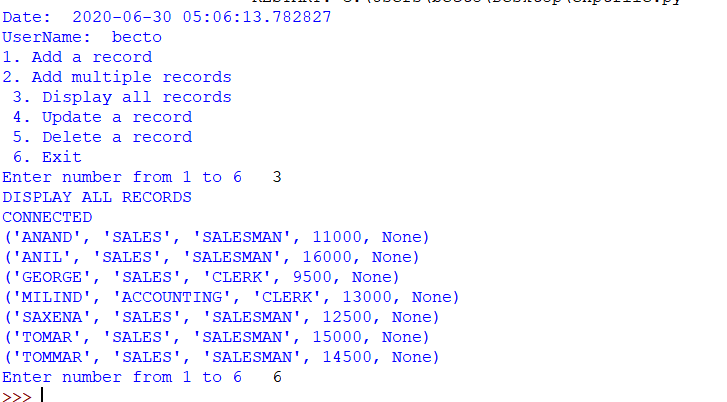
**e) Create the following table EMPLOYEE in your MySQL database: (Columns are: ENAME, DEPT\_NAME, DESIGNATION, SALARY, DATE\_OF\_JOINING)**

****

****

****

****

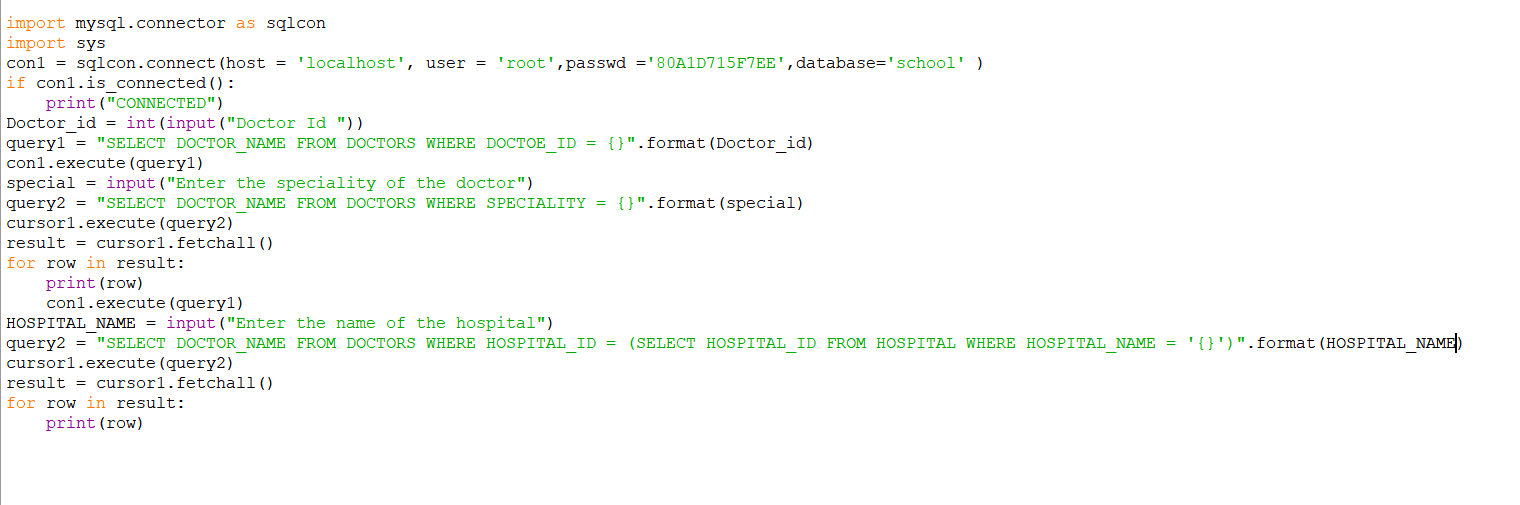
****

**Write a program in Python to do the following:**

**a) Take Doctor\_Id and Hospital\_Id from the user and display all details from the respective tables**

**b) Display the list of doctors with their details for a given speciality (taken as input from user)**

**c) Display list of doctors with their details within a given hospital**

****