

INDIAN SCHOOL, AL GHUBRA

**SULTANATE OF OMAN**

PRACTICAL RECORD

**COMPUTER SCIENCE (NEW) - 083**

**2020 - 21**

Name: **Syed Ayaan Jilani** CBSE Roll No: **41**

Class: **12** Section: **A**

**CERTIFICATE**

*It is hereby certified that* ***Syed Ayaan Jilani*** *of Class* ***12*** *Section* ***A*** *has carried out the necessary practical work as per the syllabus prescribed by the* ***Central Board of Secondary Education****, New Delhi, for the academic year* ***2020-2021****.*

Principal School Seal Teacher – in – Charge

Date: 2020-11-20

**INDEX**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Ex. No.** | **Date** | **Problem Description** | **Page** | **Remarks** |
| 1 | 13/3 | Pig Latin is a common secret language (also known as an argot). Words are created in Pig Latin by taking the first letter of the word, moving it to the end, and adding 'ay'. Write an user defined function that receives a Pig Latin word and translate back into English and return it. | 8 |  |
| 2 | 13/3 | To redact means to edit or obscure text, sometimes for security or legal reasons.  Write an UDF which redacts Top Secret documents. Your function should read in a string to be redacted, and some text. Your function should print the text, replacing the redacted string with  REDACTED. | 9 |  |
| 3 | 13/3 | The Caesar cipher is one of the earliest known and simplest ciphers. It is a type of substitution cipher in which each letter in the plaintext is 'shifted' a certain number of places down the alphabet. For example, with a shift of 1, A would be replaced by B, B would become C, and so on.  The method is named after Julius Caesar, who apparently used it to communicate with his generals. Write an user defined function in Python that takes a Text and a Shift as arguments and returns  the encrypted text. | 10 |  |
| 4 | 13/3 | In the rental shop system, the BIKEID and time OUT are stored in parallel lists.  The end of the BIKEID list is marked with the rogue value ZZZ. ID is a string list and OUT is a list of real numbers representing time. When a bike is returned, the following operations are  needed:  � the ID is looked up in the ID array  � if the ID does not exist an error message is output  � the OUT time is found  � the current time is input and the difference between that and time OUT is output.  Write a user defined function in python that receives the ID, OUT and the BIKEID if the returned bike as arguments and perform the above mentioned operations. | 11-12 |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Ex. No.** | **Date** | **Problem Description** | **Page** | **Remarks** |
| 5 | 13/3 | When an airplane is ready to take off, the pilot calls air traffic control to notify them. The  number of the airplane, the time it was due to take off and the time called in are added to a list.  When there is a free runway, the first plane on the list is called to take off.  The airplane identifier, call time and due time are stored in three lists, PLANE, DUE and CALL. An entry of ZZZ in the PLANE array indicates that there are no further airplanes waiting. (Due time and call time  are both stored as minutes since midnight.)  For example, when there are three planes waiting, the lists could be as follows.  Write a user defined procedure that returns the identifier of the next airplane to take off and the  number of airplanes left in the list. In addition, the procedure moves the remaining airplanes up  the list. | 13-14 |  |
| 6 | 27/3 | Write a Program to copy one String to another using Recursion. | 15 |  |
| 7 | 27/3 | Write a recursive program to implement binary search in a list of integers. Assume that the elements in the list are in ascending order. | 16 |  |
| 8 | 27/3 | Bubble Sort is the simplest sorting algorithm that works by repeatedly swapping the adjacent elements if they are in wrong order. Write a recursive program to arrange the list of elements in  ascending order. | 17 |  |
| 9 | 27/3 | Write program to print a pattern without using recursion. Given a number n, print following a pattern in which we basically first reduce 5 one by one until we reach a negative or 0. After we reach 0 or  negative, we add 5 until we reach n. | 18 |  |
| 10 | 27/3 | Write a Recursive Program to print multiplication table of a number. Given a number N, the task is to print its multiplication table using recursion. | 19 |  |
| 11 | 13/4 | A text file class9.txt is stored with name and aggregate marks of students who have passed in class 9 (descending order of aggregate marks). Write a python program to create 3 files names class10A.txt, class10B.txt, and class10C.txt from class9.txt  by equally distributing the students. | 20-21 |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Ex. No.** | **Date** | **Problem Description** | **Page** | **Remarks** |
| 12 | 13/4 | A text file coordinate.txt contain the following text in it:  Do less Thinking and pay more attention to your heart.  Do less Acquiring and pay more Attention to what you already have.  Do less complaining and more Attention to giving.  Do less criticizing and pay more Attention to Complementing.  Do Less talking and Pay more attention to SILIENCE.  Write function in python to count the number of words having first character  capital. | 22 |  |
| 13 | 13/4 | Write a python program to read sentence from a text file Sentence.txt, a sentence  which may be terminated by either “ . ” , “ ? ” or “ ! ” only. The words of sentence  are separated by single blank space and are in UPPER CASE.  Decode the words according to their potential and arrange them in ascending order of their potential  strength and display it. | 23-24 |  |
| 14 | 13/4 | Write a menu driven program to perfom following file operations using a binary  file product.pdf   1. Adding a new product (productid,productname, 2. unitprice,quantity) - tuple 3. searching a product using productid 4. 3.displaying the product details whose unitprice <10 5. delete the products whose quantity is <10 6. 5. decrease the unitprice of the products by 3 whose quantity is >=10 | 25-27 |  |
| 15 | 13/4 | Aditi has used a text editing software to type some text. After saving the article as  WORDS.TXT, she realised that she has wrongly typed alphabet J in place of  alphabet I everywhere in the article.  Write a function definition for JTOI() in python that would display the corrected  version of entire content of the file WORDS.TXT with all the alphabets “J” to be displayed as an alphabet “I” on screen. | 28 |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Ex. No.** | **Date** | **Problem Description** | **Page** | **Remarks** |
| 16 | 4/6 | Create a database Employee and establish database connectivity. Create a table Emp in the database with fields Empno as primary key, Empname, Salary, Department and Designation with appropriate datatypes.  Write a user defined function Insert\_Initial() to insert 6 records in the table. (Use a single parameterized query)  Write a menu driven program in Python which does the following tasks:   1. Insert details of an employee in the table. 2. Search the details of employees (The search should have a combination of 2 conditions ie it 3. has to be based on multiple criteria(Use any of the SQL operators - IN, BETWEEN, 4. Relational Operators, Like etc) 5. Update details of the employees with atleast 2 fields involved. Eg: the salary of all employees 6. in the department Sales with Designation Clerk, by 35% . 7. Remove the details of an employee. Eg: Remove all the employees of HR department who 8. earn a salary less than 15000. | 29-33 |  |
| 17 | 12/7 | Write a program to implement a stack for the given book details (Bookno, Bookname and Cost).ie each item node of the stack contains three types of information. Implement Push, Pop and  Display Operations. Top should be a global variable which will display the size of the stack atany point of time. The program should be Menu driven which should be terminated based on the  user’s choice. | 34-35 |  |
| 18 | 12/7 | Write a program in Python to reverse the contents of a file using stack. | 36-37 |  |
| 19 | 12/7 | Write a program to delete all even elements from a stack  Given a stack with n elements, the task is to remove all the elements of the stack without affecting the order of elements. | 38 |  |
| 20 | 12/7 | Write a Program to enter the numbers and perform Linear Search, Binary Search, Lowest Number and Selection Sort using list/array code. | 39-40 |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Ex. No.** | **Date** | **Problem Description** | **Page** | **Remarks** |
| 21 | 12/7 | Write a program to perform insert and delete operations on a Queue containing Member details as given on the following definition of itemnode :   * MemberNo (integer) * MemberName (String) * Age (integer) | 41-42 |  |
| 22 | 12/7 | Write a program that implements three queues namely VIP, Balcony and Regular. The program accepts the tokenID with it priority from the user eg:   * Enter tokenID: ABC123 * Priority(Highest/Normal/lowest(H/N/L):H   As per the priority entered, the element is added in the corresponding queue.  A menu offers the following options   1. Insert tokenId 2. Search for an Id   3. Change Priority | 43-44 |  |
| 23 | 25/8 | Write a menu driven program to perform read and write operations using a text file called Student.txt containing Roll\_No,Name and Address using separate functions given below   * Add\_Stud() – Entering student details. While adding data to the file, the Roll\_No field should be separated from the remaining fields with a comma separator. * Disp\_Stud() – To display the student details. * Search\_Stud() – To search a student based on Roll\_No | 45-46 |  |
| 24 | 25/8 | 1. Write a python program that allowing you to create a directory in the desktop called myDir. 2. Write a Python program allowing you to create a file in the desktop named myFile.txt and write it the following lines:   Here is an example of a text file This file was created with python We can write on this file   1. Write  a Python program allowing you to moving myFile.txt in the  directory  myDir. | 47-48 |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Ex. No.** | **Date** | **Problem Description** | **Page** | **Remarks** |
| 25 | 25/8 | Write a program in Python to create a binary file with numbers as key and the equivalent Roman numeral as the value (eg: { 1:’I’ ,6: ‘VI’….and so on). The dictionary should have a minimum of 10 numbers. Write a menu driven program to accept a number from the user and display its equivalent Roman Numeral. The program stops when the user enters -1. | 49 |  |
| 26 | 15/11 | Write MySQL statements for the following queries based on the given table. | 50-54 |  |
| 27 | 15/11 | Write MySQL statements for the following queries based on the given table. | 55-56 |  |

***PRACTICAL FILE***

***CLASS 12 – COMPUTER SCIENCE WITH PYTHON(083)***

**Ex.1**

Pig Latin is a common secret language (also known as an argot). Words are created in Pig Latin by taking the first letter of the word, moving it to the end, and adding 'ay'.

Write an user defined function that receives a Pig Latin word and translate back into English and return it. Like this:

Word: athsmay

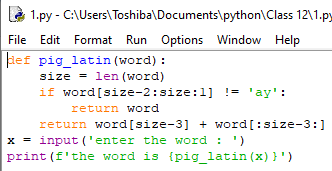
maths

If the word does not end in ay then do not change anything:

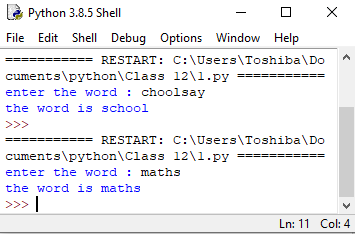
Word: science

Science

\*\*\*\*\*\*\*\*\*\* CODE\*\*\*\*\*\*\*\*\*\*\*\*



\*\*\*\*\*\*\*\*\*\* OUTPUT\*\*\*\*\*\*\*\*\*\*\*\*



**Ex.2**

To redact means to edit or obscure text, sometimes for security or legal reasons.

Write an UDF which redacts Top Secret documents. Your function should read in a string to be redacted, and some text. Your function should print the text, replacing the redacted string with REDACTED.

Here is an example:

Redact: Hercules Mulligan

Text: The spy's name is Hercules Mulligan.

The spy's name is REDACTED.

Your function should only work when the string matches exactly, including case:

Redact: rochambeau

Text: Rochambeau is the code word. When you hear rochambeau it is time to charge the fort.

Rochambeau is the code word. When you hear REDACTED it is time to charge the fort.

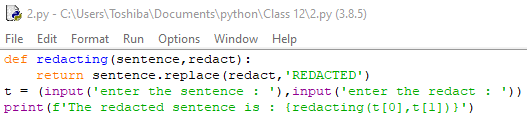
Here is an example with multiple replacements. Sometimes the replacements might be a bit strange!

Redact: Fred

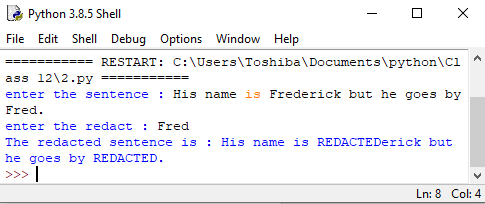
Text: His name is Frederick but he goes by Fred.

His name is REDACTEDerick but he goes by REDACTED.

\*\*\*\*\*\*\*\*\*\* CODE\*\*\*\*\*\*\*\*\*\*\*\*



\*\*\*\*\*\*\*\*\*\* OUTPUT\*\*\*\*\*\*\*\*\*\*\*\*



**Ex.3**

The Caesar cipher is one of the earliest known and simplest ciphers. It is a type of substitution cipher in which each letter in the plaintext is 'shifted' a certain number of places down the alphabet. For example, with a shift of 1, A would be replaced by B, B would become C, and so on. The method is named after Julius Caesar, who apparently used it to communicate with his

generals.

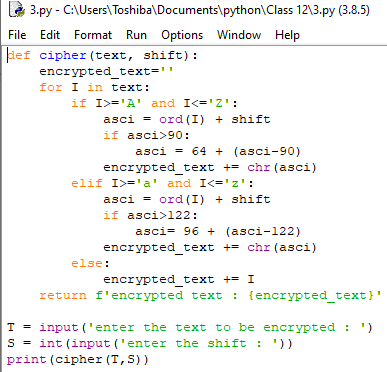
Write an user defined function in Python that takes a Text and a Shift as arguments and returns the encrypted text.

Here is a quick example of the encryption with the Caesar cipher. The text we will encrypt is 'defend the east wall of the castle', with a shift of 1.

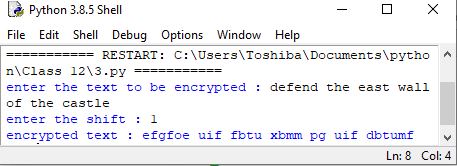
plaintext: defend the east wall of the castle

ciphertext: efgfoe uif fbtu xbmm pg uif dbtumf

\*\*\*\*\*\*\*\*\*\* CODE\*\*\*\*\*\*\*\*\*\*\*\*

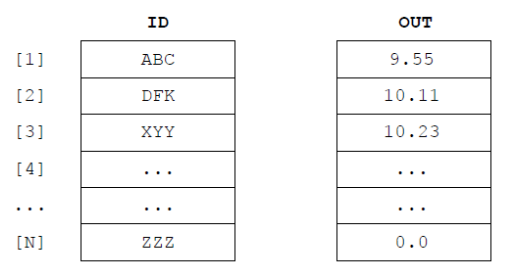


\*\*\*\*\*\*\*\*\*\* OUTPUT\*\*\*\*\*\*\*\*\*\*\*\*



**Ex.4**

In the rental shop system, the BIKEID and time OUT are stored in parallel lists as follows:



The end of the BIKEID list is marked with the rogue value ZZZ. ID is a string list and OUT is a list of real numbers representing times. When a bike is returned, the following operations are needed:

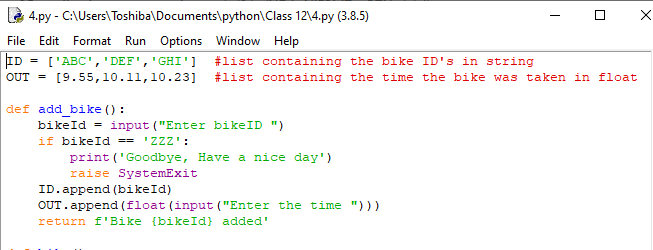
� the ID is looked up in the ID array

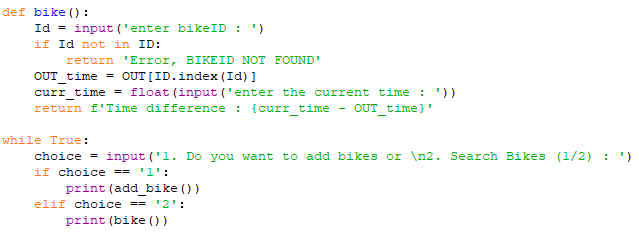
� if the ID does not exist an error message is output

� the OUT time is found

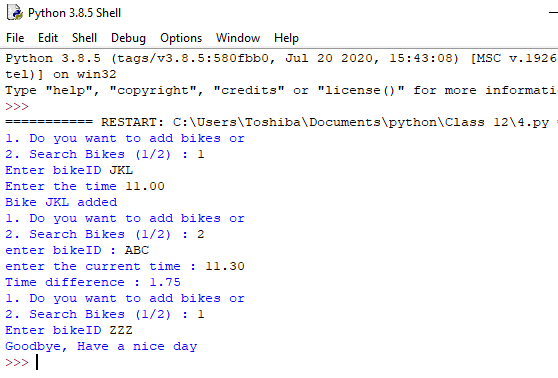
� the current time is input & the difference between that and time OUT is output

Write a user defined function in python that receives the ID, OUT and the BIKEID if the returned bike as arguments and perform the above mentioned operations.

\*\*\*\*\*\*\*\*\*\* CODE\*\*\*\*\*\*\*\*\*\*\*\*



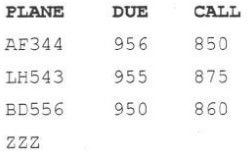
\*\*\*\*\*\*\*\*\*\* OUTPUT\*\*\*\*\*\*\*\*\*\*\*\*



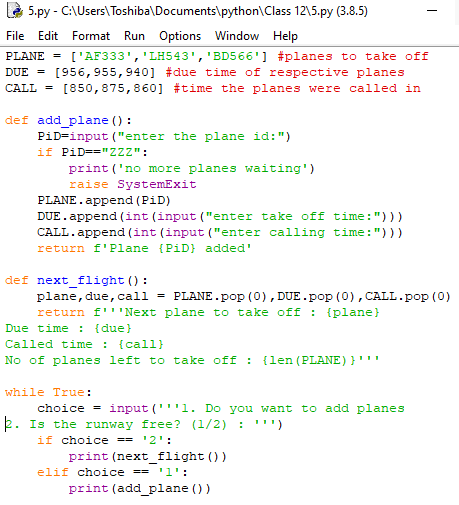
**Ex.5**

When an airplane is ready to take off, the pilot calls air traffic control to notify them. The number of the airplane, the time it was due to take off and the time called in are added to a list. When there is a free runway, the first plane on the list is called to take off. The airplane identifier, call time and due time are stored in three lists- PLANE, DUE and CALL. An entry of ZZZ in the PLANE array indicates that there are no further airplanes waiting. (Due time and call time are both stored as minutes since midnight.)

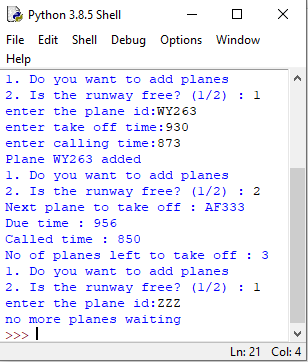
For example, when there are three planes waiting, the lists could be as follows.



Write an user defined procedure that returns the identifier of the next airplane to take off and the number of airplanes left in the list. In addition, the procedure moves the remaining airplanes up the list.

\*\*\*\*\*\*\*\*\*\* CODE\*\*\*\*\*\*\*\*\*\*\*\*

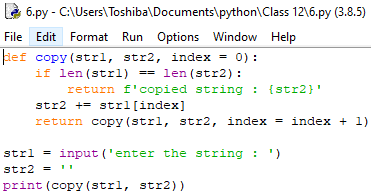
\*\*\*\*\*\*\*\*\*\* OUTPUT\*\*\*\*\*\*\*\*\*\*\*\*



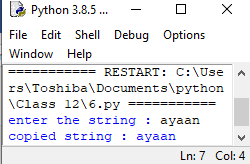
**Ex.6**

Write a Program to copy one String to another using Recursion.

\*\*\*\*\*\*\*\*\*\* CODE\*\*\*\*\*\*\*\*\*\*\*\*



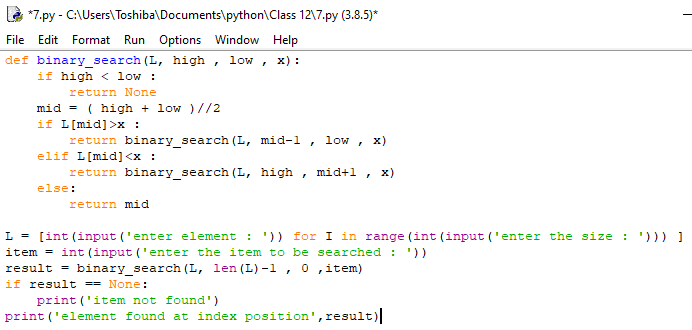
\*\*\*\*\*\*\*\*\*\* OUTPUT\*\*\*\*\*\*\*\*\*\*\*\*

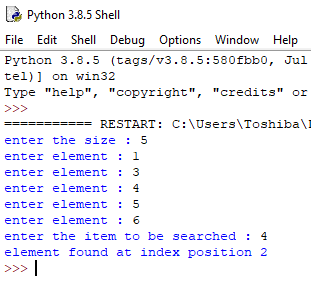


**Ex.7**

Write a recursive program to implement binary search in a list of integers. Assume that the elements in the list are in ascending order.

\*\*\*\*\*\*\*\*\*\* CODE\*\*\*\*\*\*\*\*\*\*\*\*

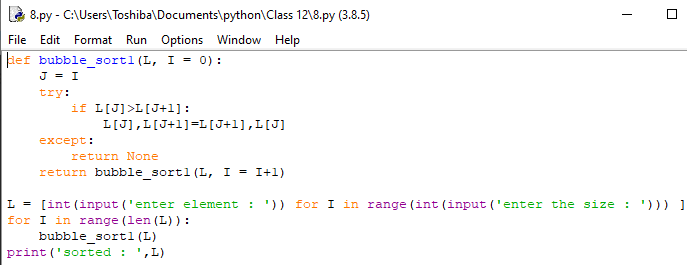


\*\*\*\*\*\*\*\*\*\* OUTPUT\*\*\*\*\*\*\*\*\*\*\*\*

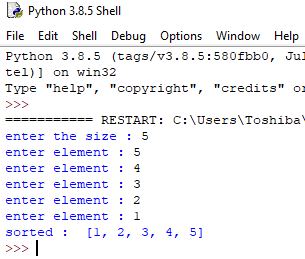
**Ex.8**

Bubble Sort is the simplest sorting algorithm that works by repeatedly swapping the adjacent elements if they are in wrong order. Write a recursive program to arrange the list of elements in ascending order.

\*\*\*\*\*\*\*\*\*\* CODE\*\*\*\*\*\*\*\*\*\*\*\*



\*\*\*\*\*\*\*\*\*\* OUTPUT\*\*\*\*\*\*\*\*\*\*\*\*



**Ex.9**

Write program to print a pattern without using recursion. Given a number n, print following a pattern.

Examples :

Input: n = 16

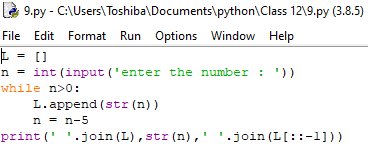
Output: 16, 11, 6, 1, -4, 1, 6, 11, 16

Input: n = 10

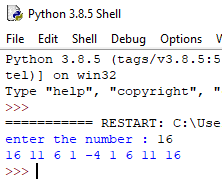
Output: 10, 5, 0, 5, 10

We basically first reduce 5 one by one until we reach a negative or 0. After we reach 0 or negative, we add 5 until we reach n.

\*\*\*\*\*\*\*\*\*\* CODE\*\*\*\*\*\*\*\*\*\*\*\*



\*\*\*\*\*\*\*\*\*\* OUTPUT\*\*\*\*\*\*\*\*\*\*\*\*



**Ex.10**

Write a Recursive Program to print multiplication table of a number. Given a number N, the task

is to print its multiplication table using recursion.

Input: N=8

Output:

8\*1=8

8\*2=16

8\*3=24

8\*4=32

8\*5=40

8\*6=48

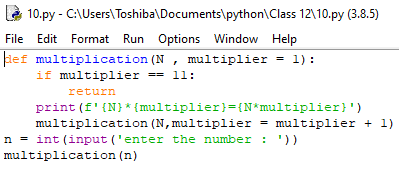
8\*7=56

8\*8=64

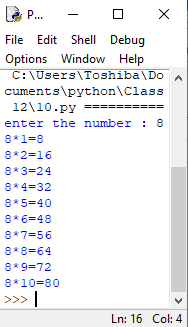
8\*9=72

8\*10 = 80

\*\*\*\*\*\*\*\*\*\* CODE\*\*\*\*\*\*\*\*\*\*\*\*



\*\*\*\*\*\*\*\*\*\* OUTPUT\*\*\*\*\*\*\*\*\*\*\*\*

****

**Ex.11**

A text file class9.txt is stored with name and aggregate marks of students who have passed in class 9 (descending order of aggregate marks). Write a python program to create 3 files names class10A.txt, class10B.txt, and class10C.txt from class9.txt by equally distributing the students.

Example:

class9.txt

Rohit 98.4

Akash Bandyopadhyay 98.4

Anurag Joshi 98.3

Pranav Nair 97.0

Kartik Pathak 97.0

Dipayan Nag 96.5

Mayurnath 96.2

Gayathri 96.1

Shifa Majeed 96

Samya Kazmi 95

Rizwan Mohammed Numan 95

class10A.txt

Rohit 98.4

Dipayan Nag 96.5

Mayurnath 96.2

class10B.txt

Akash Bandyopadhyay 98.4

Kartik Pathak 97.0

Gayathri 96.1

Rizwan Mohammed Numan 95

class10C.txt

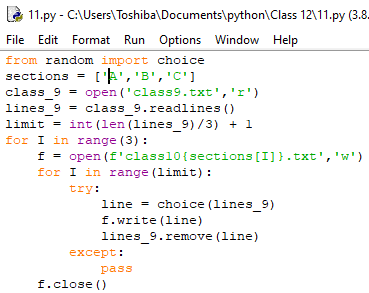
Anurag Joshi 98.3

Pranav Nair 97.0

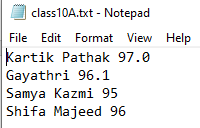
Shifa Majeed 96

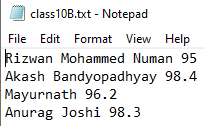
Samya Kazmi 95

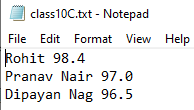
\*\*\*\*\*\*\*\*CODE\*\*\*\*\*\*\*\*\*



\*\*\*\*\*\*\*\*OUTPUT\*\*\*\*\*\*\*\*\*





****

**Ex.12**

A text file coordinate.txt contain the following text in it:

Do less Thinking and pay more attention to your heart.

Do less Acquiring and pay more Attention to what you already have.

Do less complaining and more Attention to giving.

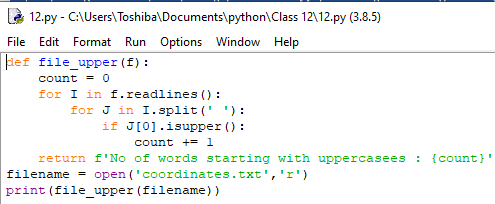
Do less criticizing and pay more Attention to Complementing.

Do Less talking and Pay more attention to SILIENCE.

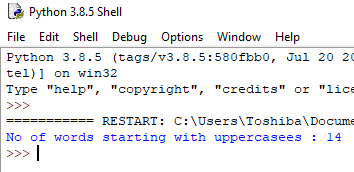
Write function in python to count the number of words having first character

capital.

\*\*\*\*\*\*\*\*\*\* CODE\*\*\*\*\*\*\*\*\*\*\*\*



\*\*\*\*\*\*\*\*\*\*OUTPUT\*\*\*\*\*\*\*\*\*



**Ex.13**

Write a python program to read sentence from a text file Sentence.txt, a sentence which may be terminated by either “ . ” , “ ? ” or “ ! ” only. The words of sentence are separated by single blank space and are in UPPER CASE. Decode the words according to their potential and arrange them in ascending order of their potential strength and display it. Test your program with the following data and some random data:

Example:

INPUT: HOW DO YOU DO?

OUTPUT: HOW = 238

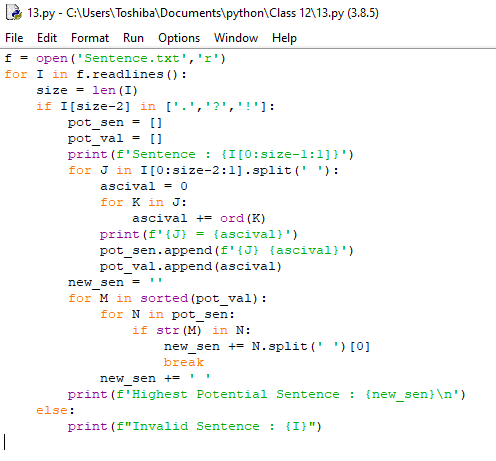
DO = 147

YOU = 253

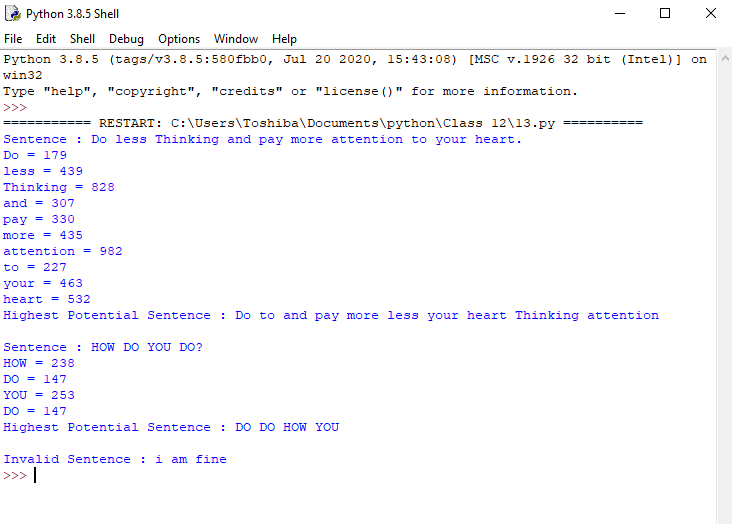
DO = 147

DO DO HOW YOU

\*\*\*\*\*\*\*\*\*\* CODE\*\*\*\*\*\*\*\*\*\*\*\*



\*\*\*\*\*\*\*\*\*OUTPUT\*\*\*\*\*\*\*\*\*



**Ex.14**

Write a menu driven program to perfom following file operations using a binary file product.pdf

1. Adding a new product (productid,productname, unitprice,quantity) - tuple

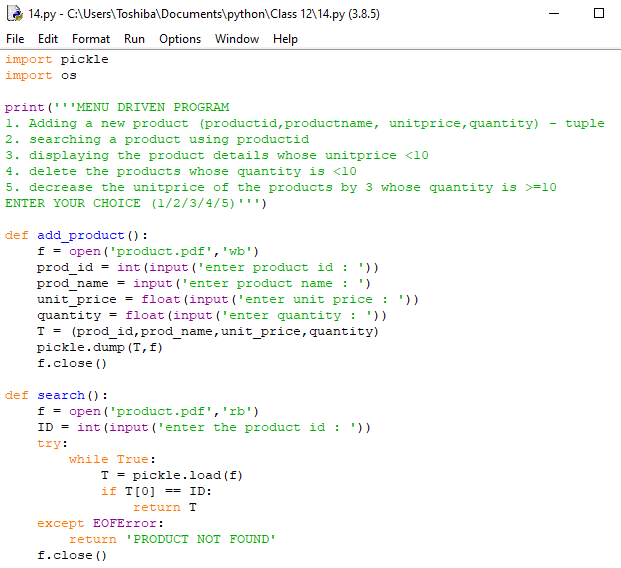
2. searching a product using productid

3.displaying the product details whose unitprice <10

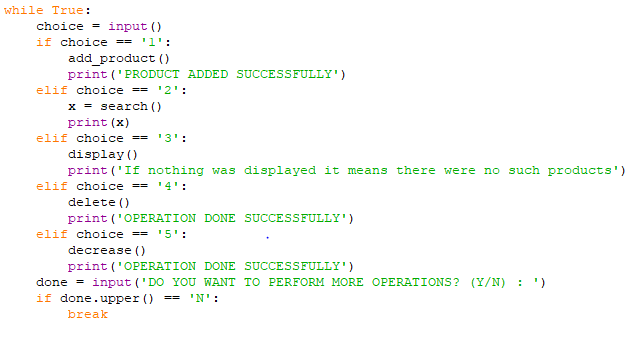
4. delete the products whose quantity is <10

5. decrease the unitprice of the products by 3 whose quantity is >=10

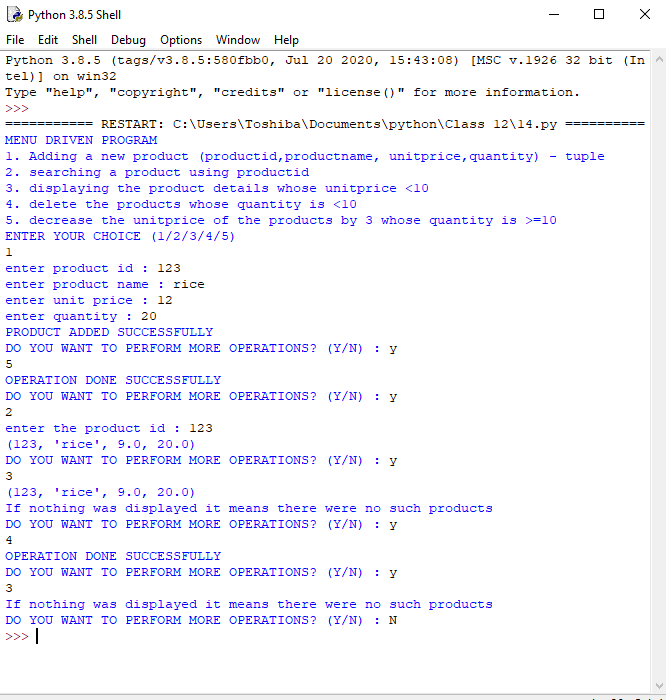
\*\*\*\*\*\*\*\*\*\* CODE\*\*\*\*\*\*\*\*\*\*\*\*







\*\*\*\*\*\*\*OUTPUT\*\*\*\*\*\*\*\*\*



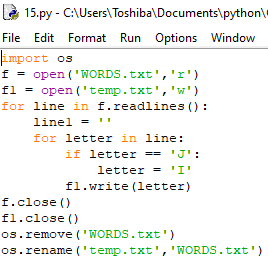
**Ex.15**

Aditi has used a text editing software to type some text. After saving the article as WORDS.TXT, she realised that she has wrongly typed alphabet J in place of alphabet I everywhere in the article. Write a function definition for JTOI() in python that would display the corrected version of entire content of the file WORDS.TXT with all the alphabets “J” to be

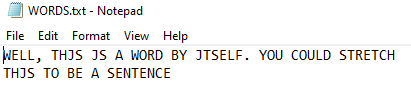
displayed as an alphabet “I” on screen.

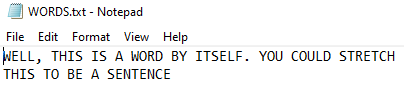
Note : Assuming that WORD.TXT does not contain any J alphabet otherwise.

\*\*\*\*\*\*\*\*\*\* CODE\*\*\*\*\*\*\*\*\*\*\*\*



\*\*\*\*\*\*\*\*\*\*OUTPUT\*\*\*\*\*\*\*\*\*





**Ex.16**

MySQL-Python Connectivity

* Create a database Employee and establish database connectivity.
* Create a table Emp in the database with fields Empno as primary key, Empname, Salary, Department and Designation with appropriate datatypes.
* Write a user defined function Insert\_Initial() to insert 6 records in the table. (Use a single parameterized query)

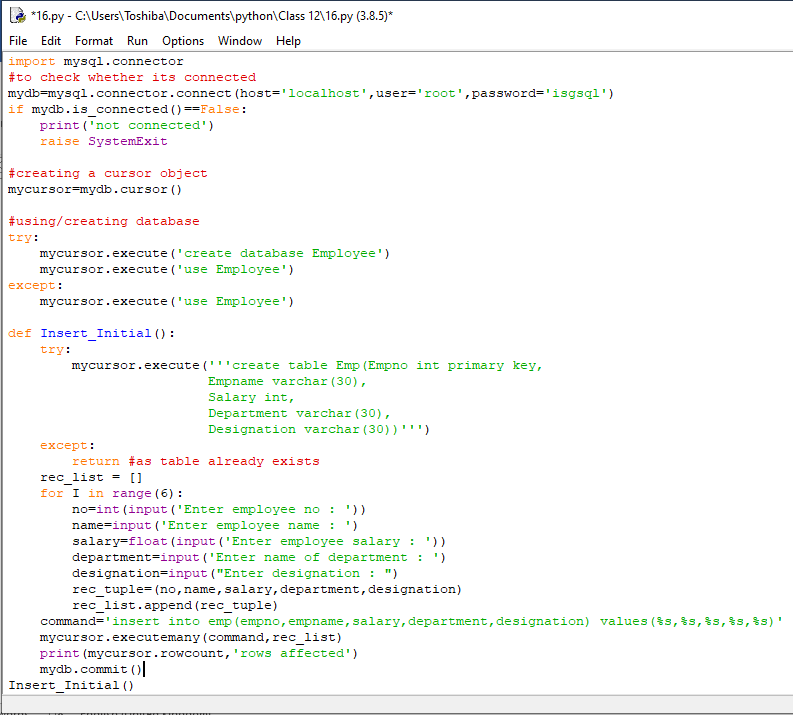
Write a menu driven program in Python which does the following tasks:

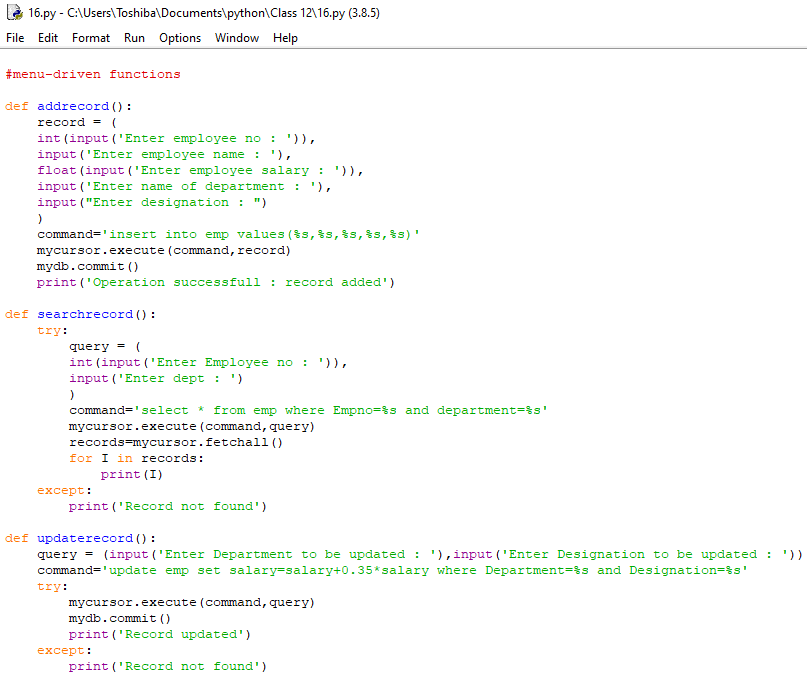
1. Insert details of an employee in the table.
2. Search the details of employees (The search should have a combination of 2 conditions ie it has to be based on multiple criteria(Use any of the SQL operators - IN, BETWEEN, Relational Operators, Like etc)
3. Update details of the employees with atleast 2 fields involved. Eg: the salary of all employees in the department Sales with Designation Clerk, by 35% .
4. Remove the details of an employee. Eg: Remove all the employees of HR department who earn a salary less than 15000.

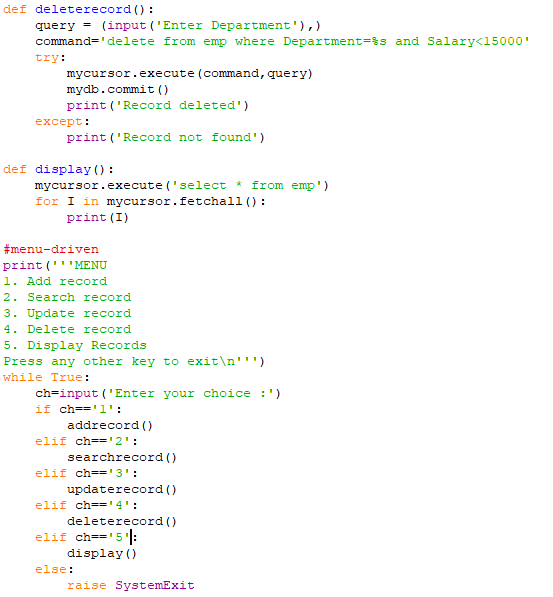
Points to be noted

* Insert and Delete should be associated with suitable confirmation messages (Hint:rowcount)
* Error in connection needs to be tackled with appropriate function (Hint:built in)
* Write separate user defined functions for Insert, Search, Delete, Update and Display
* Exceptions should be handled.

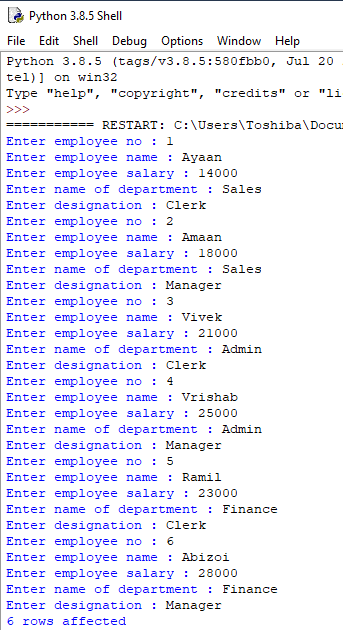
\*\*\*\*\*\*\*\*\*\*CODE\*\*\*\*\*\*\*\*\*\*\*

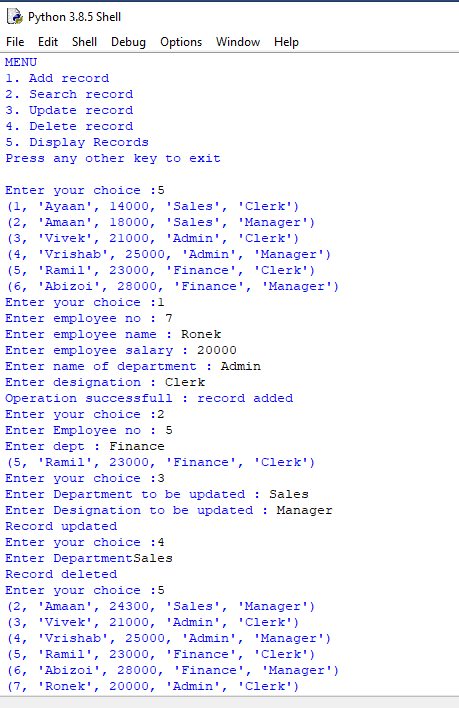






\*\*\*\*\*\*\*OUTPUT\*\*\*\*\*\*\*\*



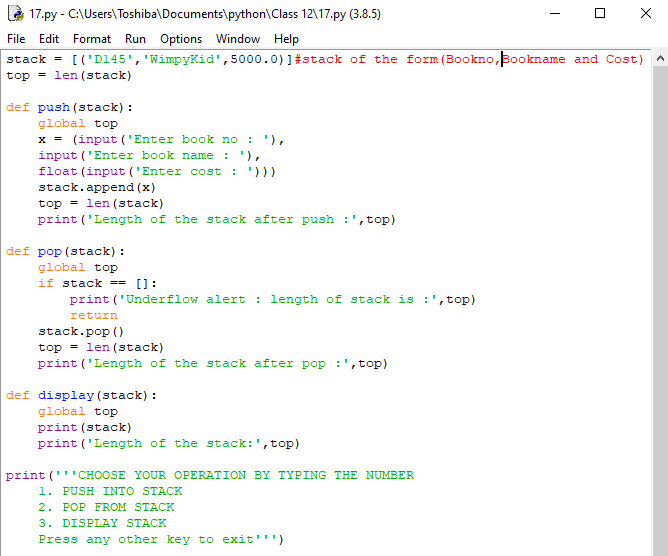


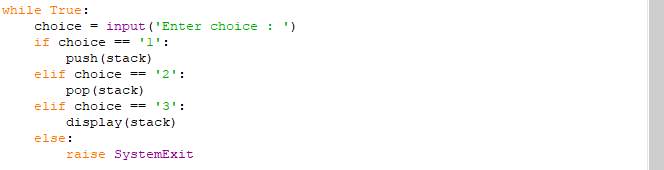


**Ex.17**

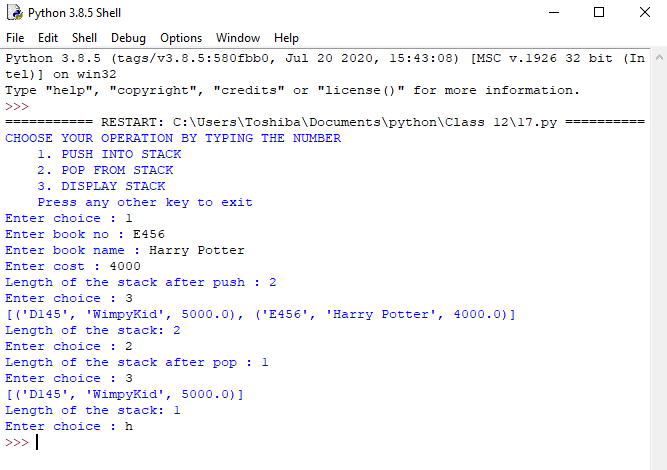
Write a program to implement a stack for the given book details (Bookno, Bookname and Cost) ie each item node of the stack contains three types of information. Implement Push, Pop and Display Operations. Top should be a global variable which will display the size of the stack at any point of time. The program should be Menu driven which should be terminated based on the user’s choice.

\*\*\*\*\*\*\*\*\*\* CODE\*\*\*\*\*\*\*\*\*\*\*\*





\*\*\*\*\*\*\*\*OUTPUT\*\*\*\*\*\*\*\*



**Ex.18**

Write a program in Python to reverse the contents of a file using stack.

Examples:

Input :

This is the program

to reverse the

content of a file

using stack

Output :

using stack

content of a file

to reverse the

This is the program

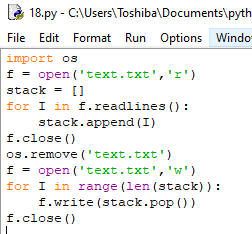
Hint:

 Create an empty stack.

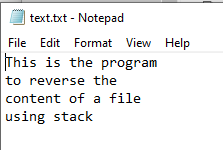
 One by one push every line of the file to the stack.

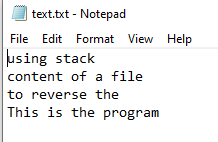
 One by one pop each line from the stack and put them back to the file.

\*\*\*\*\*\*\*\*\*\* CODE\*\*\*\*\*\*\*\*\*\*\*\*



\*\*\*\*\*\*\*\*OUTPUT\*\*\*\*\*\*\*\*





**Ex.19**

Write a program to delete all even elements from a stack

Given a stack with n elements, the task is to remove all the elements of the stack without affecting the order of elements.

Example:

Input : s = 16 &lt;- 15 &lt;- 29 &lt;- 24 &lt;- 19 (TOP)

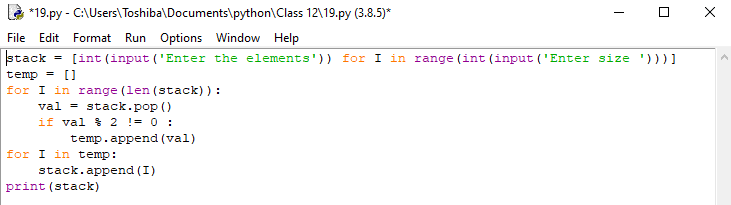
Output: 19 29 15

19 29 15 is the order of odd elements in which they will be popped from the given stack.

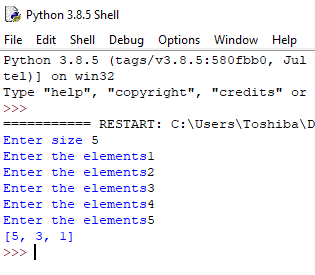
Approach:

* Create a temporary stack temp and start popping the elements of the given stack s.
* For every popped element say val, if val % 2 == 1 then push it to temp.
* At the end of step 2, temp will contain all the odd elements from s but in reverse order.
* Now, to get the original order, pop every element from temp & push it to s.

\*\*\*\*\*\*\*\*\*\* CODE\*\*\*\*\*\*\*\*\*\*\*\*



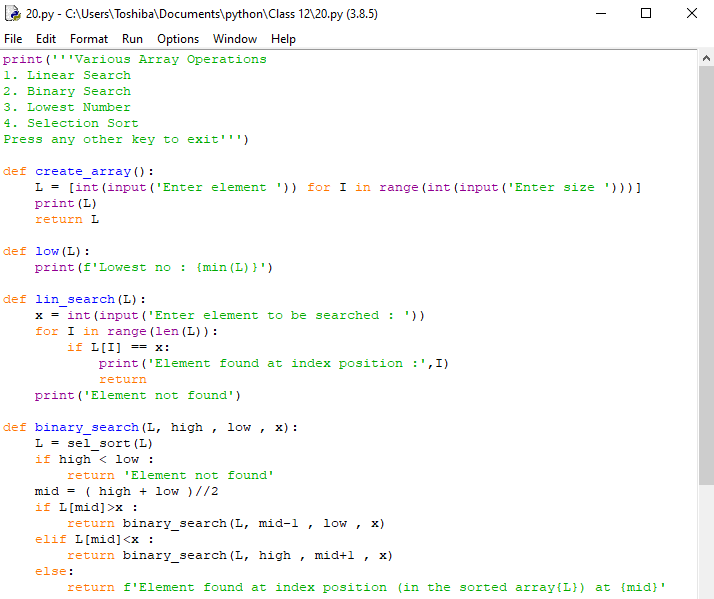
\*\*\*\*\*\*\*OUTPUT\*\*\*\*\*\*\*

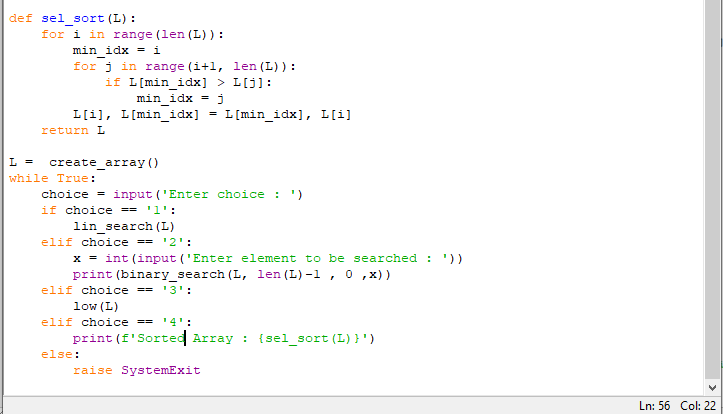


**Ex.20**

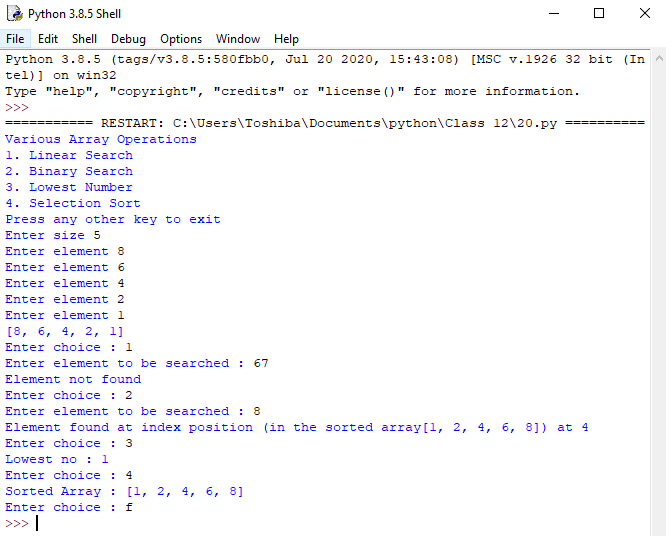
Write a Program to enter the numbers and perform Linear Search, Binary Search, Lowest Number and Selection Sort using list/array code.

\*\*\*\*\*\*\*\*\*\* CODE\*\*\*\*\*\*\*\*\*\*\*\*





\*\*\*\*\*\*\*\*\*OUTPUT\*\*\*\*\*\*\*\*\*

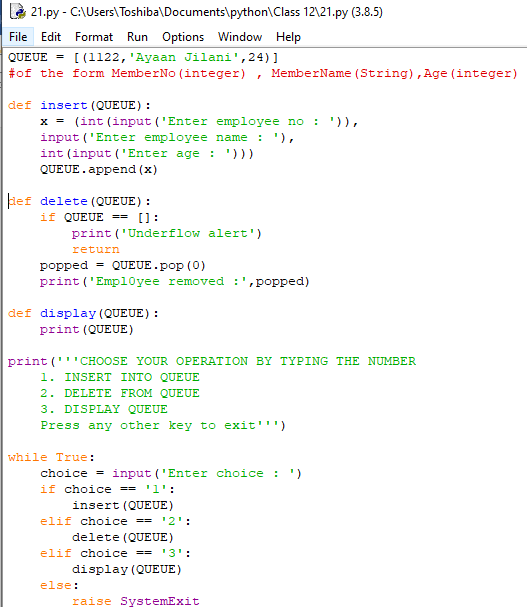


**Ex.21**

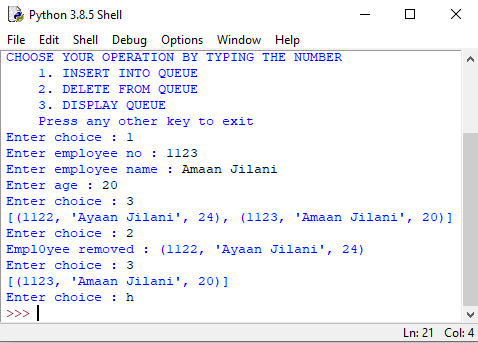
Write a program to perform insert and delete operations on a Queue containing Member details as given on the following definition of itemnode:

* MemberNo (integer)
* MemberName (String)
* Age (integer)

\*\*\*\*\*\*\*\*\*\* CODE\*\*\*\*\*\*\*\*\*\*\*\*



\*\*\*\*\*\*\*\*\*\*\*OUTPUT\*\*\*\*\*\*\*\*\*\*\*



**Ex.22**

Write a program that implements three queues namely VIP, Balcony and Regular. The program accepts the tokenID with it priority from the user eg:

Enter tokenID: ABC123

Priority(Highest/Normal/lowest(H/N/L): H

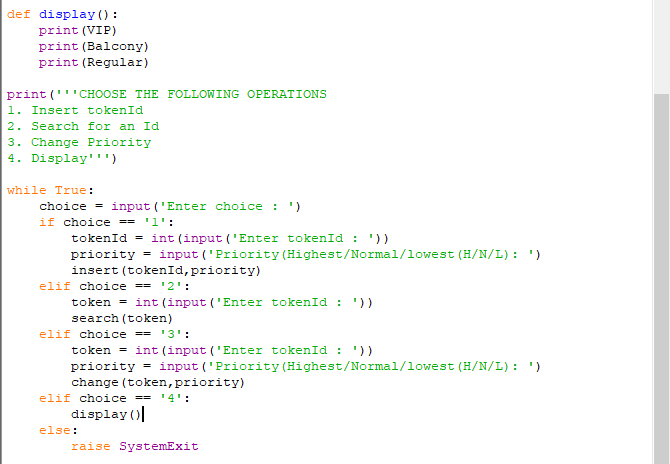
As per the priority entered, the element is added in the corresponding queue.

A menu offers the following options

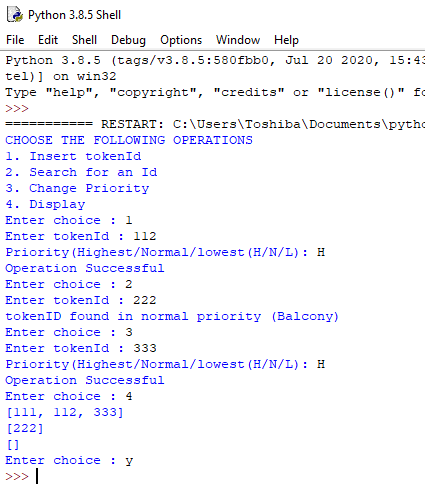
1. Insert tokenId
2. Search for an Id
3. Change Priority

\*\*\*\*\*\*\*\*\*\* CODE\*\*\*\*\*\*\*\*\*\*\*\*





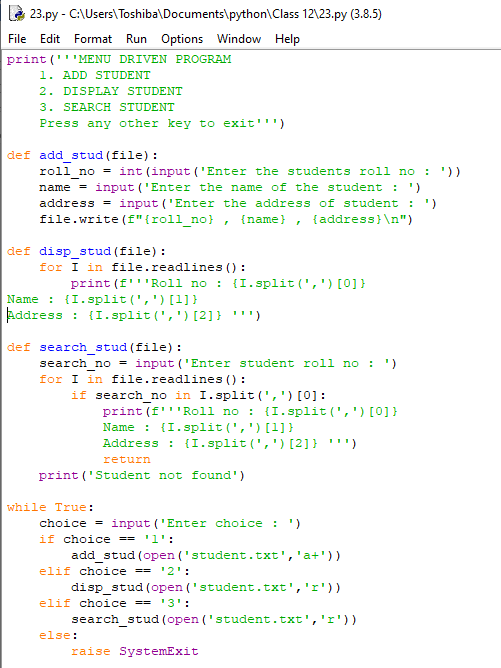
\*\*\*\*\*\*\*\*\*OUTPUT\*\*\*\*\*\*\*\*\*



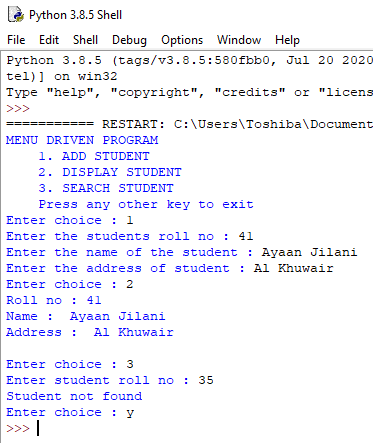
**Ex.23**

Write a menu driven program to perform read and write operations using a text file called Student.txt containing Roll\_No,Name and Address using separate functions given below

* Add\_Stud() – Entering student details. While adding data to the file, the Roll\_No field should be separated from the remaining fields with a comma separator.
* Disp\_Stud() – To display the student details.
* Search\_Stud() – To search a student based on Roll\_No

\*\*\*\*\*\*\*\*\*\* CODE\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*OUTPUT\*\*\*\*\*\*\*\*\*\*



**Ex.24**

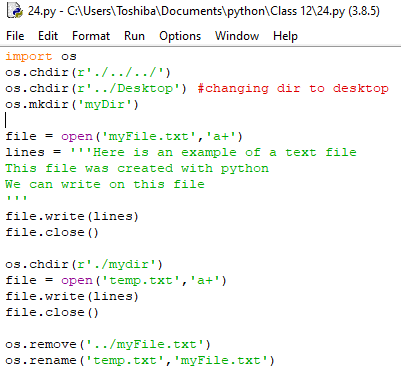
Write a python program that allowing you to create a directory in the desktop called myDir.

Write  a Python program allowing you to create a file in the desktop named myFile.txt and write it the following lines:

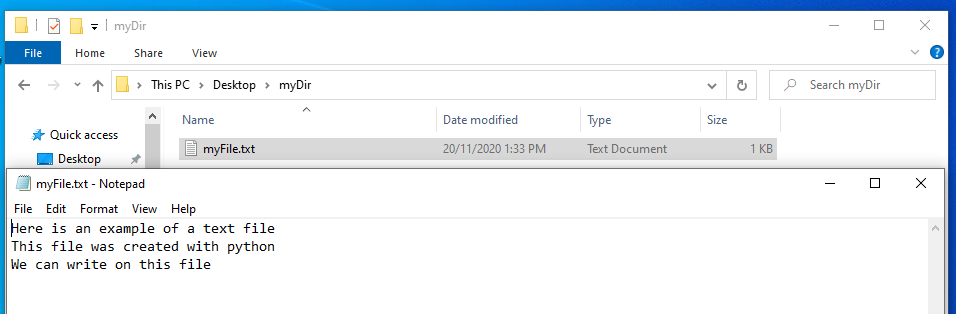
Here is an example of a text file  
This file was created with python  
We can write on this file

Write  a Python program allowing you to moving myFile.txt in the  directory  myDir.

\*\*\*\*\*\*\*\*\*\* CODE\*\*\*\*\*\*\*\*\*\*\*\*



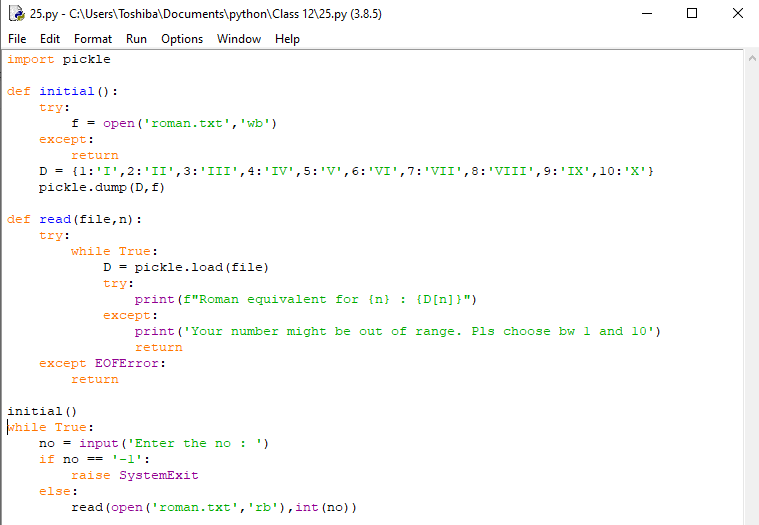
\*\*\*\*\*\*\*\*\*OUTPUT\*\*\*\*\*\*\*\*\*\*



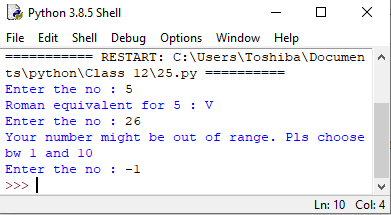
**Ex.25**

Write a program in Python to create a binary file with numbers as key and the equivalent Roman numeral as the value (eg: { 1:’I’ ,6: ‘VI’….and so on). The dictionary should have a minimum of 10 numbers. Write a menu driven program to accept a number from the user and display its equivalent Roman Numeral. The program stops when the user enters -1.

\*\*\*\*\*\*\*\*\*\* CODE\*\*\*\*\*\*\*\*\*\*\*\*



\*\*\*\*\*\*\*\*OUTPUT\*\*\*\*\*\*\*\*

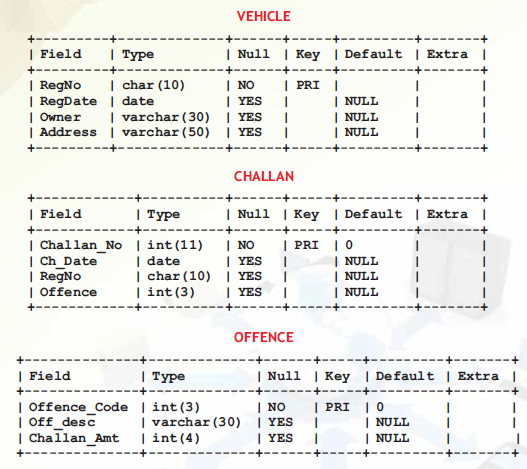


**Ex.26**

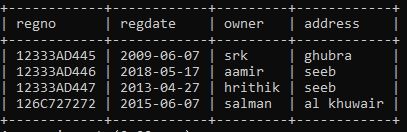
Based on these tables write SQL statements for the following queries:

1. Display the dates of first registration and last registration from the table Vehicle.
2. Display the number of challans issued on each date.
3. Display the total number of challans issued for each offence.
4. Display the total number of vehicles for which the 3rd and 4th characters of RegNo are '6C'.
5. Display the total value of challans issued for which the Off\_Desc is 'Driving without License'.
6. Display details of the challans issued on '2010-04-03' along with Off\_Desc for each challan.
7. Display the RegNo of all vehicles which have been challaned more than once.
8. Display details of each challan alongwith vehicle details, Off\_desc, and Challan\_Amt.

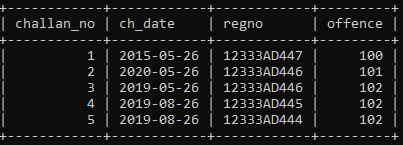
\*\*\*\*\*\*\*\*\*\*\*TABLES\*\*\*\*\*\*\*\*\*\*\*\*\*



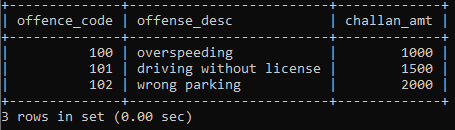
**VEHICLE**



**CHALLAN**

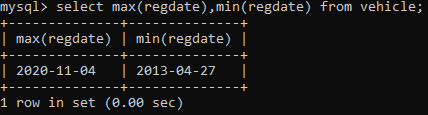


**OFFENCE**

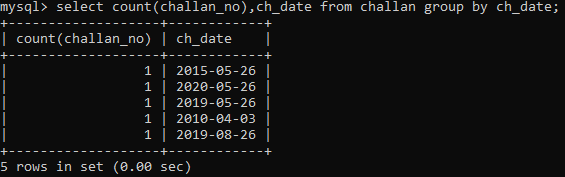


\*\*\*\*\*\*\*\*OUTPUTS\*\*\*\*\*\*\*\*\*

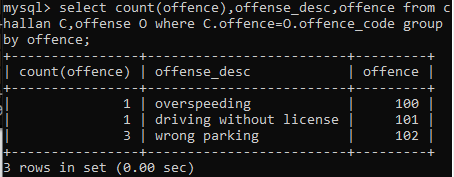
1. Select max(regdate),min(regdate) from vehicle;



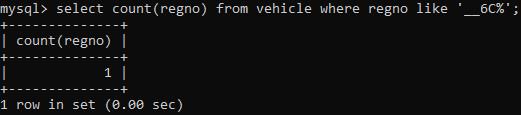
1. Select count(challan\_no),ch\_date from challan group by ch\_date;.



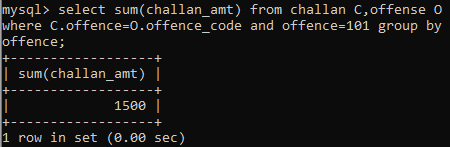
1. Select count(offence),offense\_desc,offence from challan C,offense O where C.offence=O.offence\_code group by offence;.



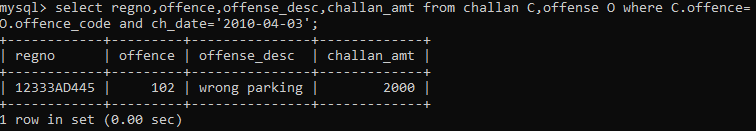
1. Select count(regno) from vehicle where regno like ‘\_\_6C%’;



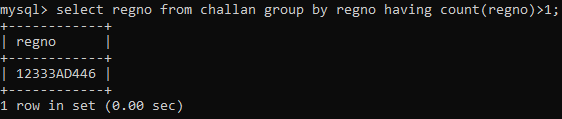
1. Select sum(challan\_amt) from challan C,offense O where C.offence=O.offence\_code and offence=101 group by offence;



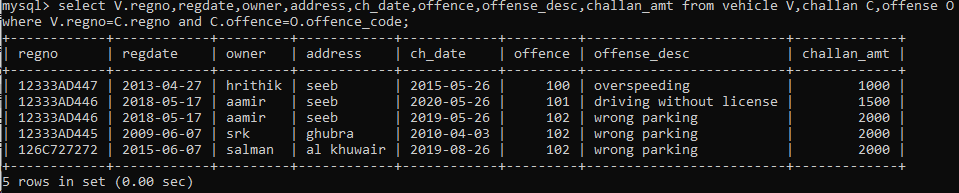
1. Select regno,offence,offense\_desc,challan\_amt from challan C,offense O where C.offence=O.offence\_code and ch\_date=’2010-04-03’;

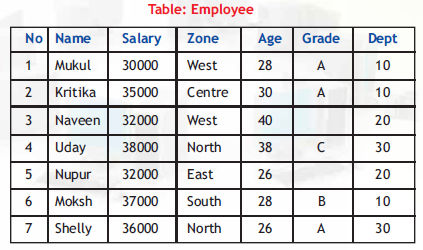


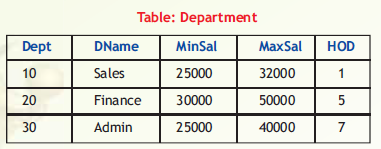
1. Select regno from challan group by regno having count(regno)>1;



1. Select V.regno,regdate,owner,address,ch\_date,offence,offense\_desc,challan\_amt from vehicle V,challan C,offense O where V.regno=C.regno and C.offence=O.offence\_code;



**Ex.27**



Based on these tables write SQL statements for the following queries:

i. Display the details of all the employees who work in Sales department.

ii. Display the Salary, Zone, and Grade of all the employees whose HOD is Nupur.

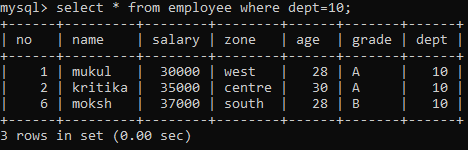
iii. Display the Name and Department Name of all the employees.

iv. Display the names of all the employees whose salary is not within the specified range for the corresponding department.

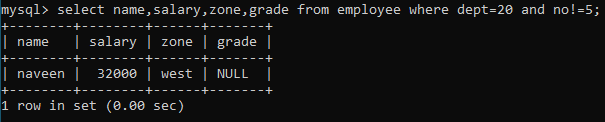
v. Display the name of the department and the name of the corresponding HOD for all the departments.

\*\*\*\*\*\*\*\*OUTPUTS\*\*\*\*\*\*\*\*

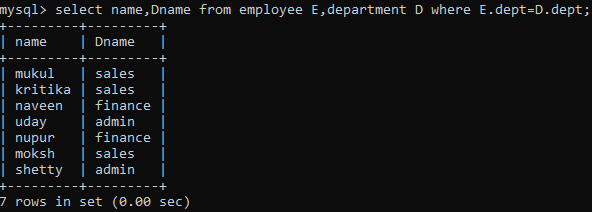
1. Select \* from employee where dept=10;.



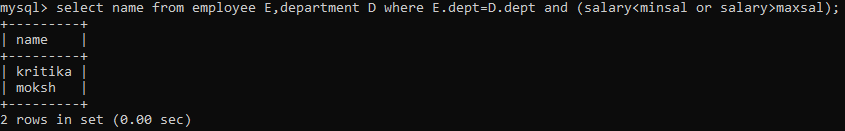
1. Select name,salary,zone,grade from employee where dept=20 and no!=5;



1. Select name.Dname from employee E,department D where E.dept=D.dept;



1. Select name from employee E,department D where E.dept=D.dept and (salary<minsal or salary>maxsal);



1. Select name,Dname from employee E,department D where E.no=D.hod;

