

# **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:  <ul style="list-style-type: none"> <li>◆ the ID is looked up in the ID array</li> <li>◆ if the ID does not exist an error message is output</li> <li>◆ the OUT time is found</li> <li>◆ the current time is input and the difference between that and time OUT is output.</li> </ul> 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	<p>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.</p> <p>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.</p> <p>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.</p>	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	<p>A text file coordinate.txt contain the following text in it:</p> <p>Do less Thinking and pay more attention to your heart.</p> <p>Do less Acquiring and pay more Attention to what you already have.</p> <p>Do less complaining and more Attention to giving.</p> <p>Do less criticizing and pay more Attention to Complementing.</p> <p>Do Less talking and Pay more attention to SILENCE.</p> <p>Write function in python to count the number of words having first character capital.</p>	22	
13	13/4	<p>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.</p> <p>Decode the words according to their potential and arrange them in ascending order of their potential strength and display it.</p>	23-24	
14	13/4	<p>Write a menu driven program to perform following file operations using a binary file product.pdf</p> <ol style="list-style-type: none"> <li>1. Adding a new product (productid,productname, unitprice,quantity) - tuple</li> <li>2. searching a product using productid</li> <li>3. displaying the product details whose unitprice &lt;10</li> <li>4. delete the products whose quantity is &lt;10</li> <li>5. decrease the unitprice of the products by 3 whose quantity is &gt;=10</li> </ol>	25-27	
15	13/4	<p>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.</p> <p>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.</p>	28	

Ex. No.	Date	Problem Description	Page	Remarks
16	4/6	<p>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.</p> <p>Write a user defined function Insert_Initial() to insert 6 records in the table. (Use a single parameterized query)</p> <p>Write a menu driven program in Python which does the following tasks:</p> <ol style="list-style-type: none"> <li>1. Insert details of an employee in the table.</li> <li>2. Search the details of employees (The search should have a combination of 2 conditions ie it               <ol style="list-style-type: none"> <li>1. has to be based on multiple criteria(Use any of the SQL operators - IN, BETWEEN,</li> <li>2. Relational Operators, Like etc)</li> </ol> </li> <li>3. Update details of the employees with atleast 2 fields involved. Eg: the salary of all employees</li> <li>4. in the department Sales with Designation Clerk, by 35% .</li> <li>5. Remove the details of an employee. Eg: Remove all the employees of HR department who</li> <li>6. earn a salary less than 15000.</li> </ol>	29-33	
17	12/7	<p>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.</p>	34-35	
18	12/7	<p>Write a program in Python to reverse the contents of a file using stack.</p>	36-37	
19	12/7	<p>Write a program to delete all even elements from a stack</p> <p>Given a stack with n elements, the task is to remove all the elements of the stack without affecting the order of elements.</p>	38	
20	12/7	<p>Write a Program to enter the numbers and perform Linear Search, Binary Search, Lowest Number and Selection Sort using list/array code.</p>	39-40	

Ex. No.	Date	Problem Description	Page	Remarks
21	12/7	<p>Write a program to perform insert and delete operations on a Queue containing Member details as given on the following definition of itemnode :</p> <ul style="list-style-type: none"> <li>• MemberNo (integer)</li> <li>• MemberName (String)</li> <li>• Age (integer)</li> </ul>	41-42	
22	12/7	<p>Write a program that implements three queues namely VIP, Balcony and Regular. The program accepts the tokenID with it priority from the user eg:</p> <ul style="list-style-type: none"> <li>• Enter tokenID: ABC123</li> <li>• Priority(Highest/Normal/lowest(H/N/L):H</li> </ul> <p>As per the priority entered, the element is added in the corresponding queue.</p> <p>A menu offers the following options</p> <ol style="list-style-type: none"> <li>1. Insert tokenId</li> <li>2. Search for an Id</li> <li>3. Change Priority</li> </ol>	43-44	
23	25/8	<p>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</p> <ul style="list-style-type: none"> <li>• 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.</li> <li>• Disp_Stud() – To display the student details.</li> <li>• Search_Stud() – To search a student based on Roll_No</li> </ul>	45-46	
24	25/8	<ol style="list-style-type: none"> <li>1. Write a python program that allowing you to create a directory in the desktop called myDir.</li> <li>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</li> <li>3. Write a Python program allowing you to moving myFile.txt in the directory myDir.</li> </ol>	47-48	

<b>Ex. No.</b>	<b>Date</b>	<b>Problem Description</b>	<b>Page</b>	<b>Remarks</b>
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

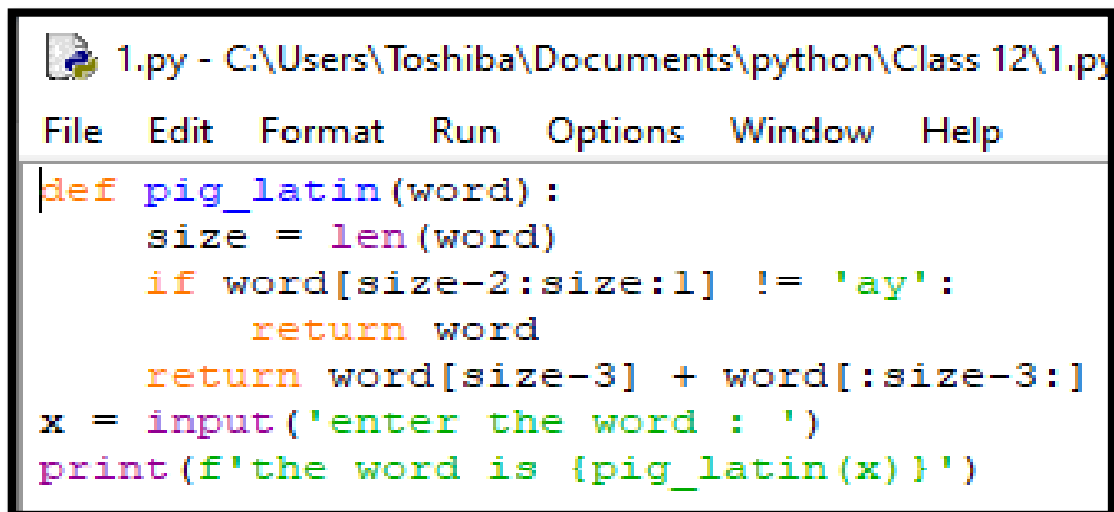
maths

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

Word: science

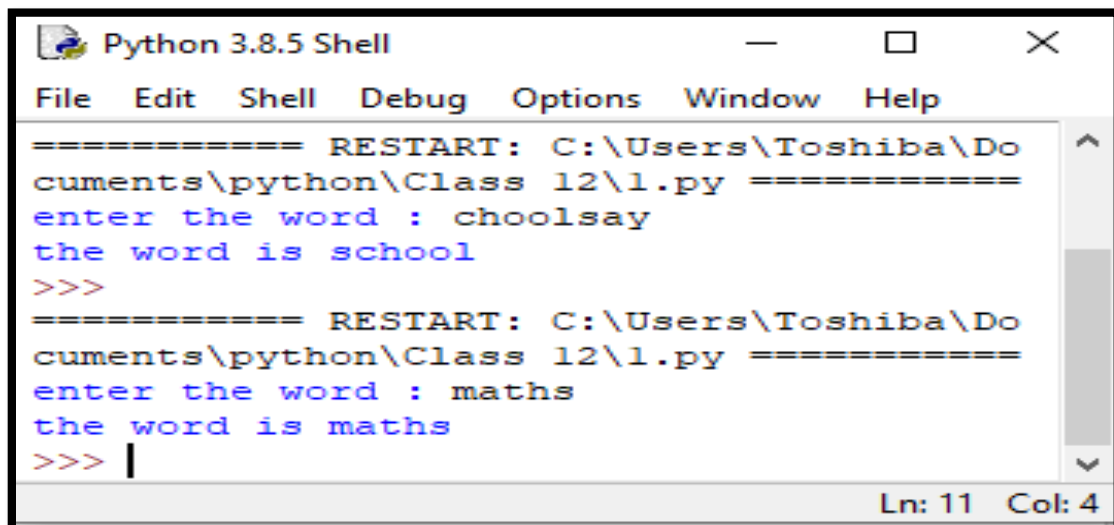
Science

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



```
1.py - C:\Users\Toshiba\Documents\python\Class 12\1.py
File Edit Format Run Options Window Help
def pig_latin(word):
    size = len(word)
    if word[size-2:size:1] != 'ay':
        return word
    return word[size-3] + word[:size-3:]
x = input('enter the word : ')
print(f'the word is {pig_latin(x)}')
```

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



```
Python 3.8.5 Shell
File Edit Shell Debug Options Window Help
===== RESTART: C:\Users\Toshiba\Documents\python\Class 12\1.py =====
enter the word : choolsay
the word is school
>>>
===== RESTART: C:\Users\Toshiba\Documents\python\Class 12\1.py =====
enter the word : maths
the word is maths
>>> |
Ln: 11 Col: 4
```

## 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.

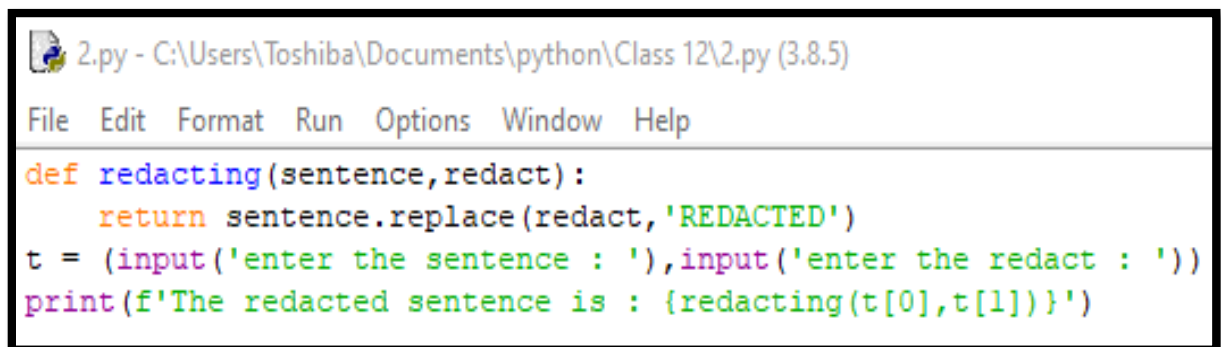
Here is an example with multiple replacements. Sometimes the replacements might be a bit strange!

Redact: Fred

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

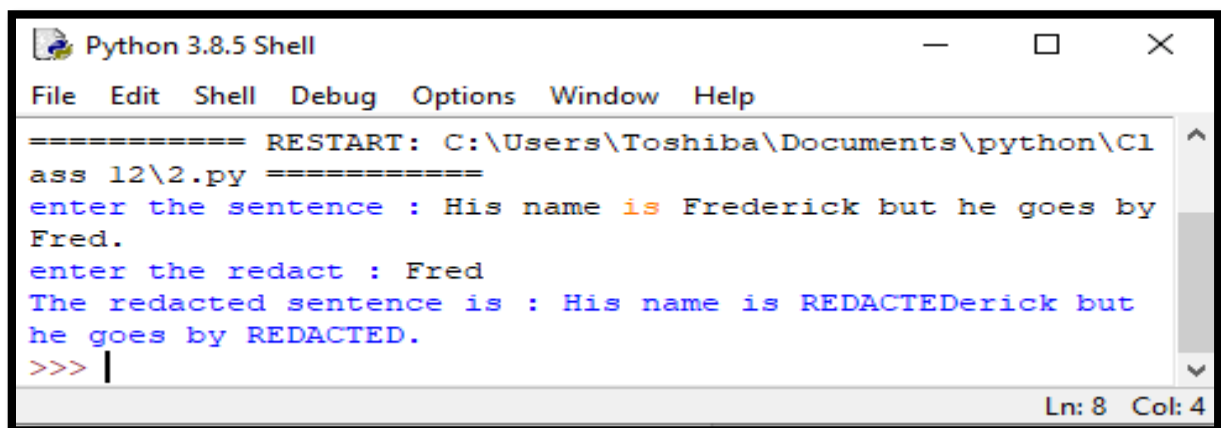
His name is REDACTEDerick but he goes by REDACTED.

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



```
2.py - C:\Users\Toshiba\Documents\python\Class 12\2.py (3.8.5)
File Edit Format Run Options Window Help
def redacting(sentence,redact):
    return sentence.replace(redact,'REDACTED')
t = (input('enter the sentence : '),input('enter the redact : '))
print(f'The redacted sentence is : {redacting(t[0],t[1])}')
```

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



```
Python 3.8.5 Shell
File Edit Shell Debug Options Window Help
===== RESTART: C:\Users\Toshiba\Documents\python\Cl
ass 12\2.py =====
enter the sentence : His name is Frederick but he goes by
Fred.
enter the redact : Fred
The redacted sentence is : His name is REDACTEDerick but
he goes by REDACTED.
>>> |
```

### 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.

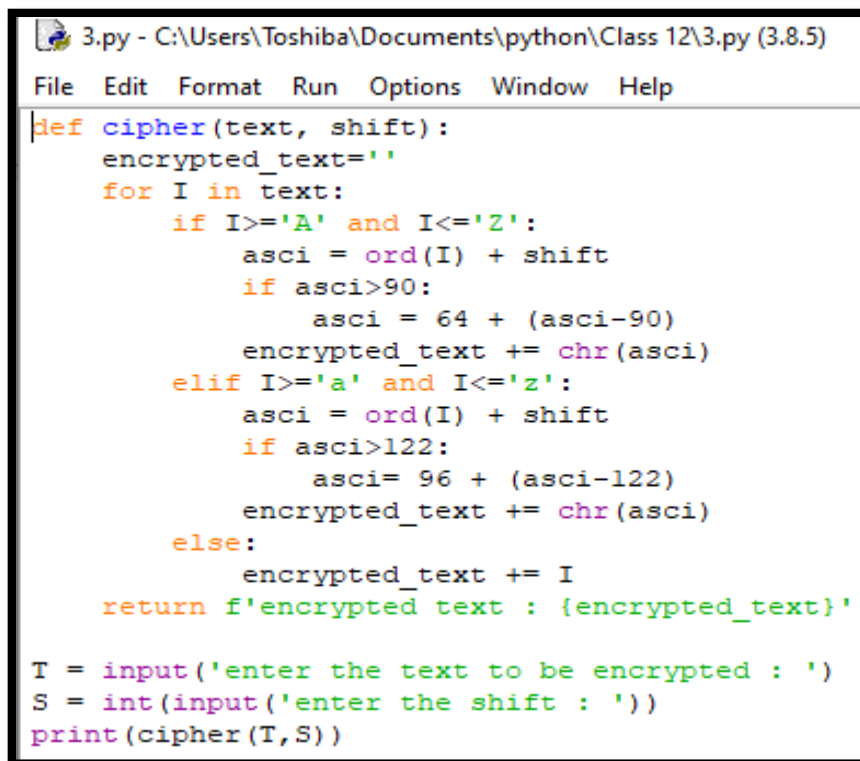
Write an user defined function in Python that takes a Text and a Shift as arguments and returns the encrypted text.

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.

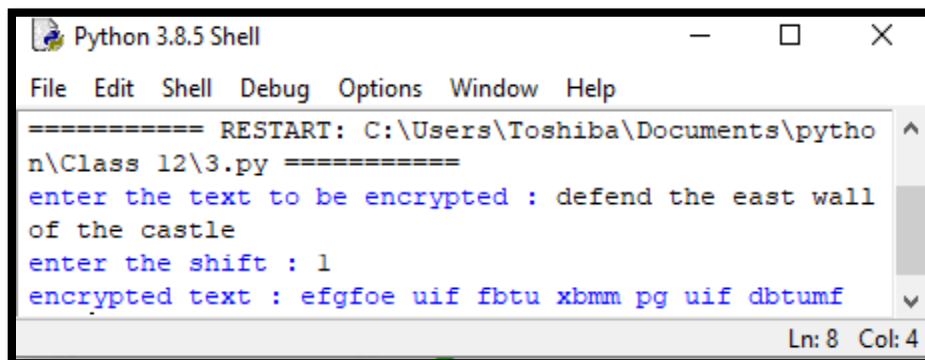
plaintext: defend the east wall of the castle

ciphertext: efgfoe uif fbtu xbmm pg uif dbtumf

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

A screenshot of a Python IDE window titled '3.py - C:\Users\Toshiba\Documents\python\Class 12\3.py (3.8.5)'. The window contains a Python function 'cipher(text, shift)' that takes a string 'text' and an integer 'shift' as arguments. The function iterates through each character 'I' in 'text'. If 'I' is an uppercase letter (between 'A' and 'Z'), it calculates its ASCII value, adds the shift, and wraps around if it exceeds 90. If 'I' is a lowercase letter (between 'a' and 'z'), it calculates its ASCII value, adds the shift, and wraps around if it exceeds 122. If 'I' is not a letter, it remains unchanged. The function returns a string 'encrypted text : {encrypted\_text}'. Below the function, there are two input prompts: 'T = input('enter the text to be encrypted : ')' and 'S = int(input('enter the shift : '))', followed by 'print(cipher(T,S))'.

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

A screenshot of a Python 3.8.5 Shell window. The window shows the execution of the program. It starts with a 'RESTART' message. Then, it prompts 'enter the text to be encrypted : ' and the user enters 'defend the east wall of the castle'. Next, it prompts 'enter the shift : ' and the user enters '1'. Finally, it outputs 'encrypted text : efgfoe uif fbtu xbmm pg uif dbtumf'. The status bar at the bottom indicates 'Ln: 8 Col: 4'.

#### Ex.4

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

	ID	OUT
[1]	ABC	9.55
[2]	DEF	10.11
[3]	XYZ	10.23
[4]	...	...
...	...	...
[N]	ZZZ	0.0

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\*\*\*\*\*

```
4.py - C:\Users\Toshiba\Documents\python\Class 12\4.py (3.8.5)
File Edit Format Run Options Window Help
ID = ['ABC','DEF','GHI'] #list containing the bike ID's in string
OUT = [9.55,10.11,10.23] #list containing the time the bike was taken in float

def add_bike():
    bikeId = input("Enter bikeID ")
    if bikeId == 'ZZZ':
        print('Goodbye, Have a nice day')
        raise SystemExit
    ID.append(bikeId)
    OUT.append(float(input("Enter the time ")))
    return f'Bike {bikeId} added'
```

```

def bike():
    Id = input('enter bikeID : ')
    if Id not in ID:
        return 'Error, BIKEID NOT FOUND'
    OUT_time = OUT[ID.index(Id)]
    curr_time = float(input('enter the current time : '))
    return f'Time difference : {curr_time - OUT_time}'

while True:
    choice = input('1. Do you want to add bikes or \n2. Search Bikes (1/2) : ')
    if choice == '1':
        print(add_bike())
    elif choice == '2':
        print(bike())

```

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

```

Python 3.8.5 Shell
File Edit Shell Debug Options Window Help
Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020, 15:43:08) [MSC v.1926
tel)] on win32
Type "help", "copyright", "credits" or "license()" for more informati
>>>
===== RESTART: C:\Users\Toshiba\Documents\python\Class 12\4.py
1. Do you want to add bikes or
2. Search Bikes (1/2) : 1
Enter bikeID JKL
Enter the time 11.00
Bike JKL added
1. Do you want to add bikes or
2. Search Bikes (1/2) : 2
enter bikeID : ABC
enter the current time : 11.30
Time difference : 1.75
1. Do you want to add bikes or
2. Search Bikes (1/2) : 1
Enter bikeID ZZZ
Goodbye, Have a nice day
>>> |

```

## 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.

PLANE	DUE	CALL
AF344	956	850
LH543	955	875
BD556	950	860
ZZZ		

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\*\*\*\*\*

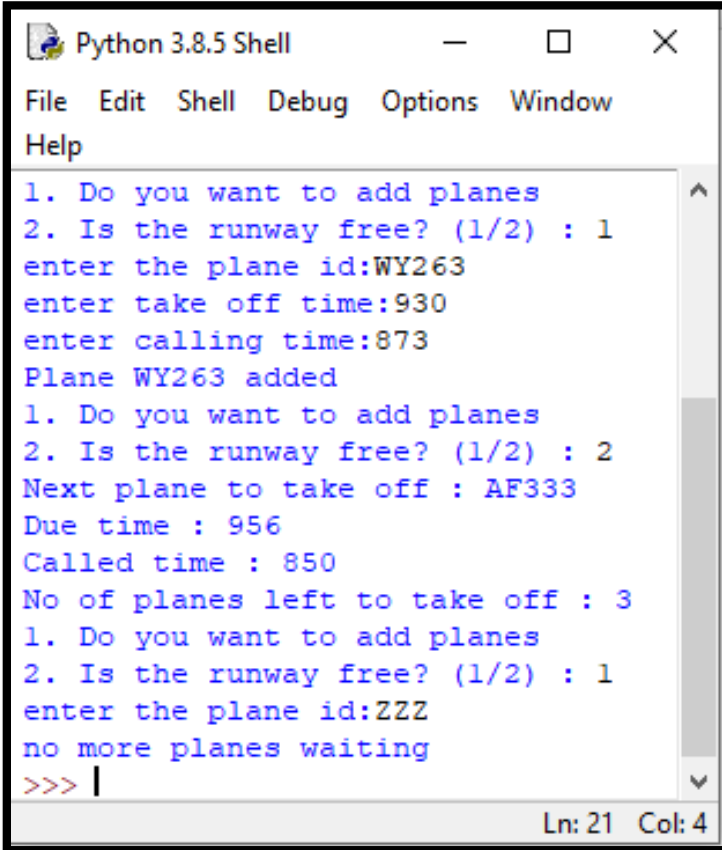
```
5.py - C:\Users\Toshiba\Documents\python\Class 12\5.py (3.8.5)
File Edit Format Run Options Window Help
PLANE = ['AF333', 'LH543', 'BD566'] #planes to take off
DUE = [956, 955, 940] #due time of respective planes
CALL = [850, 875, 860] #time the planes were called in

def add_plane():
    PiD=input("enter the plane id:")
    if PiD=="ZZZ":
        print('no more planes waiting')
        raise SystemExit
    PLANE.append(PiD)
    DUE.append(int(input("enter take off time:")))
    CALL.append(int(input("enter calling time:")))
    return f'Plane {PiD} added'

def next_flight():
    plane,due,call = PLANE.pop(0),DUE.pop(0),CALL.pop(0)
    return f'''Next plane to take off : {plane}
Due time : {due}
Called time : {call}
No of planes left to take off : {len(PLANE)}'''

while True:
    choice = input('1. Do you want to add planes
2. Is the runway free? (1/2) : ')
    if choice == '2':
        print(next_flight())
    elif choice == '1':
        print(add_plane())
```

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

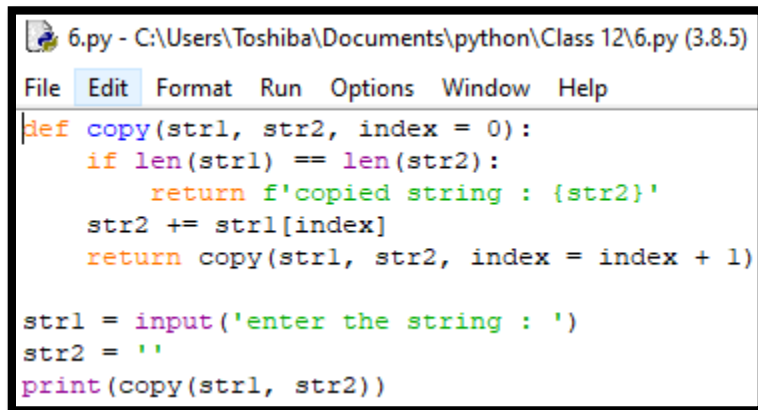
A screenshot of a Python 3.8.5 Shell window. The window has a title bar with the text 'Python 3.8.5 Shell' and standard window controls (minimize, maximize, close). Below the title bar is a menu bar with the following items: 'File', 'Edit', 'Shell', 'Debug', 'Options', 'Window', and 'Help'. The main area of the window displays the output of a program in blue text. The output consists of several lines of prompts and user input. The first part shows a loop where the user is asked if they want to add planes, and if so, if the runway is free. The user enters '1' for both prompts. Then, the user is prompted to enter the plane id, take off time, and calling time. The user enters 'WY263', '930', and '873' respectively. The program then outputs 'Plane WY263 added'. This sequence repeats for a second plane, 'AF333', with take off time '956' and calling time '850'. After this, the program outputs 'No of planes left to take off : 3'. The user is then asked if they want to add more planes, and they enter '1'. They are then prompted for the plane id, and they enter 'ZZZ'. The program then outputs 'no more planes waiting'. The prompt '>>>>' is shown at the end of the output. A status bar at the bottom right of the window shows 'Ln: 21 Col: 4'.

```
Python 3.8.5 Shell
File Edit Shell Debug Options Window Help
1. Do you want to add planes
2. Is the runway free? (1/2) : 1
enter the plane id:WY263
enter take off time:930
enter calling time:873
Plane WY263 added
1. Do you want to add planes
2. Is the runway free? (1/2) : 2
Next plane to take off : AF333
Due time : 956
Called time : 850
No of planes left to take off : 3
1. Do you want to add planes
2. Is the runway free? (1/2) : 1
enter the plane id:ZZZ
no more planes waiting
>>> |
Ln: 21 Col: 4
```

## Ex.6

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

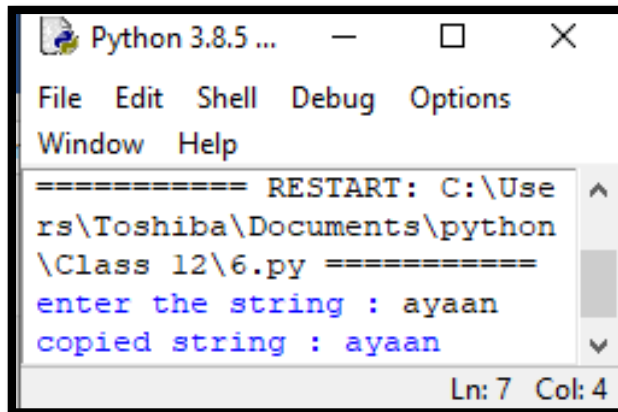
\*\*\*\*\* CODE\*\*\*\*\*



```
6.py - C:\Users\Toshiba\Documents\python\Class 12\6.py (3.8.5)
File Edit Format Run Options Window Help
def copy(str1, str2, index = 0):
    if len(str1) == len(str2):
        return f'copied string : {str2}'
    str2 += str1[index]
    return copy(str1, str2, index = index + 1)

str1 = input('enter the string : ')
str2 = ''
print(copy(str1, str2))
```

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



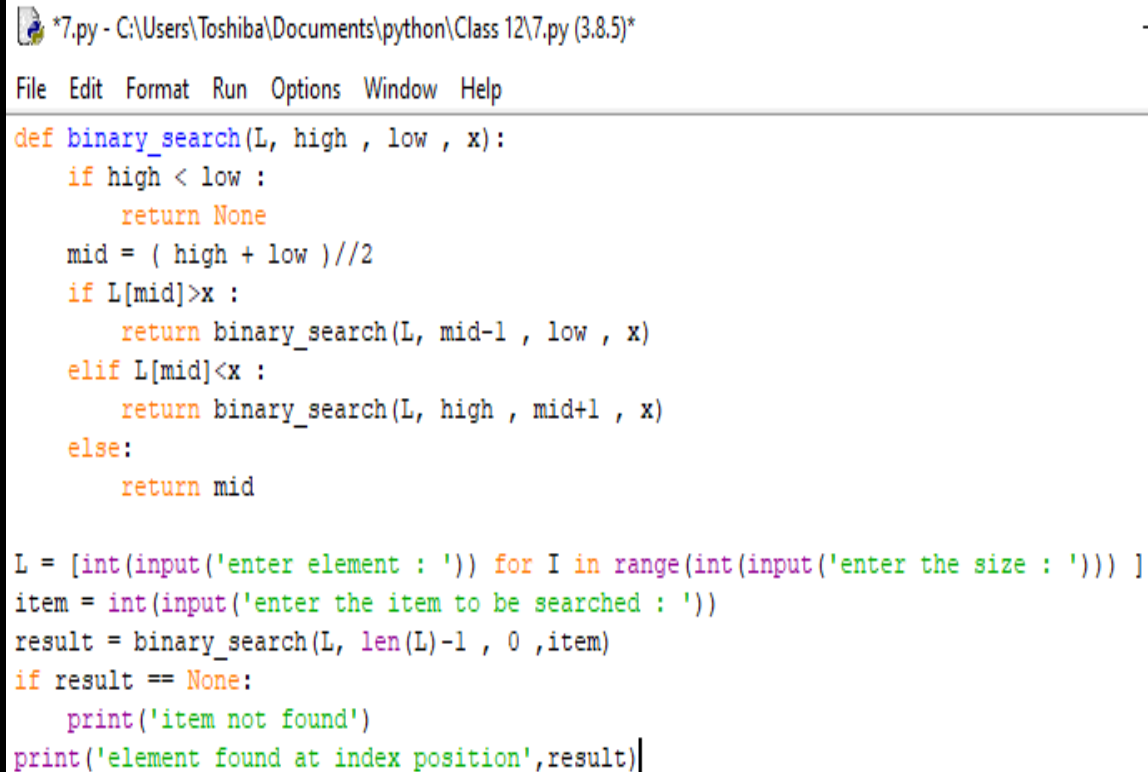
```
Python 3.8.5 ...
File Edit Shell Debug Options
Window Help
===== RESTART: C:\Use
rs\Toshiba\Documents\python
\Class 12\6.py =====
enter the string : ayaan
copied string : ayaan
Ln: 7 Col: 4
```



## 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.

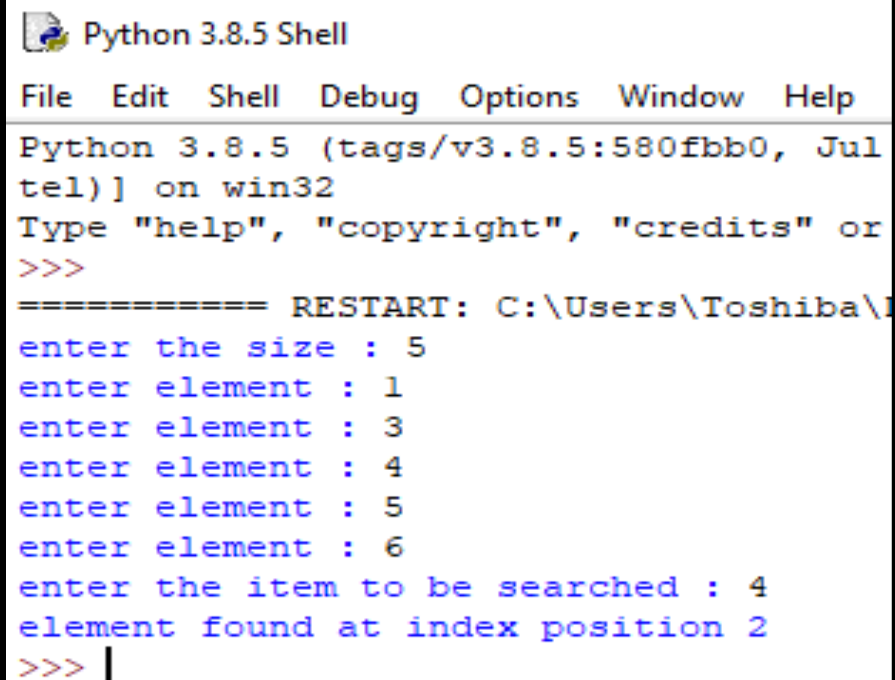
\*\*\*\*\* CODE\*\*\*\*\*



```
*7.py - C:\Users\Toshiba\Documents\python\Class 12\7.py (3.8.5)*
File Edit Format Run Options Window Help
def binary_search(L, high , low , x):
    if high < low :
        return None
    mid = ( high + low )//2
    if L[mid]>x :
        return binary_search(L, mid-1 , low , x)
    elif L[mid]<x :
        return binary_search(L, high , mid+1 , x)
    else:
        return mid

L = [int(input('enter element : ')) for I in range(int(input('enter the size : '))) ]
item = int(input('enter the item to be searched : '))
result = binary_search(L, len(L)-1 , 0 ,item)
if result == None:
    print('item not found')
print('element found at index position',result)
```

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

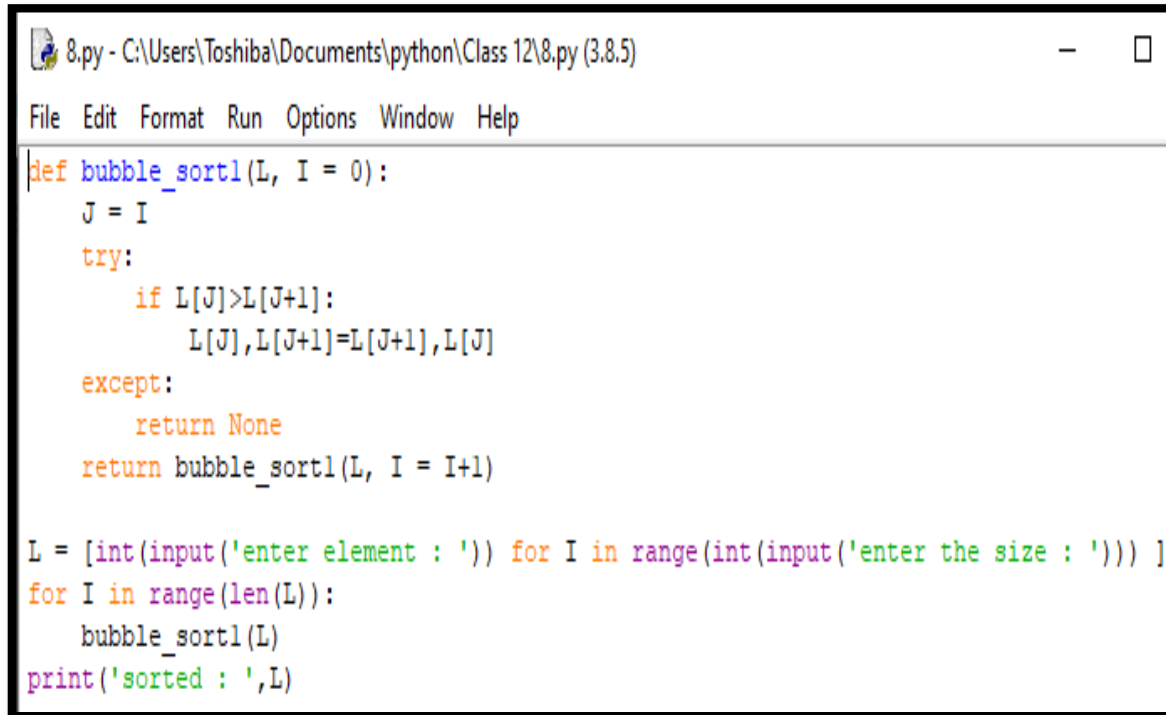


```
Python 3.8.5 Shell
File Edit Shell Debug Options Window Help
Python 3.8.5 (tags/v3.8.5:580fbb0, Jul
tel)] on win32
Type "help", "copyright", "credits" or
>>>
===== RESTART: C:\Users\Toshiba\
enter the size : 5
enter element : 1
enter element : 3
enter element : 4
enter element : 5
enter element : 6
enter the item to be searched : 4
element found at index position 2
>>> |
```

## 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.

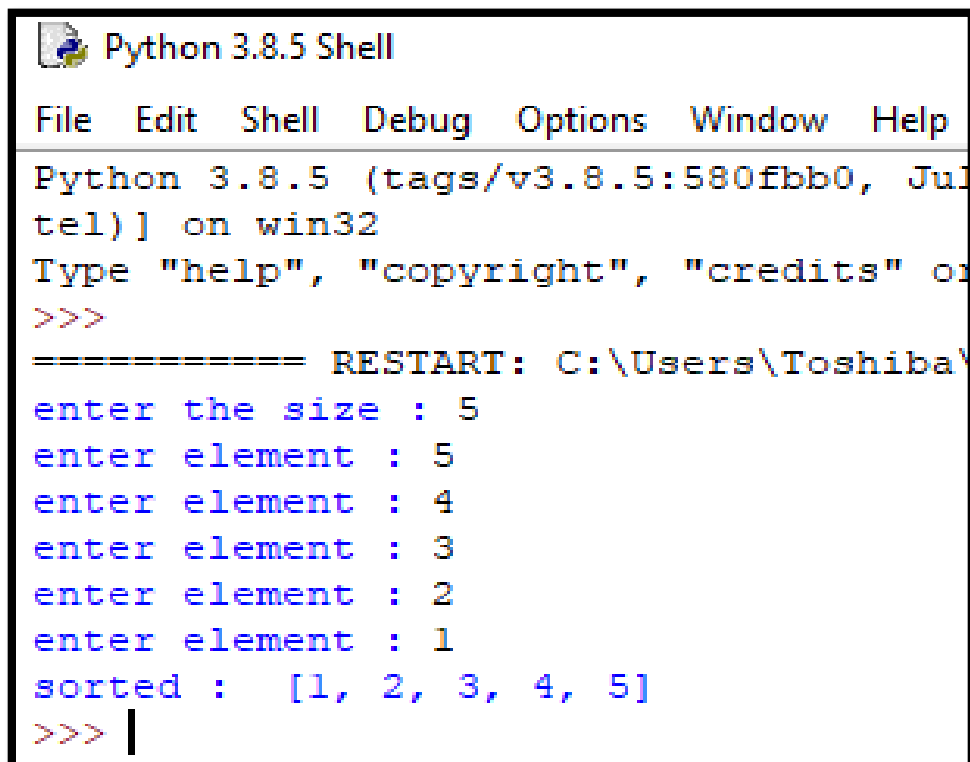
\*\*\*\*\* CODE\*\*\*\*\*



```
8.py - C:\Users\Toshiba\Documents\python\Class 12\8.py (3.8.5)
File Edit Format Run Options Window Help
def bubble_sort1(L, I = 0):
    J = I
    try:
        if L[J]>L[J+1]:
            L[J],L[J+1]=L[J+1],L[J]
    except:
        return None
    return bubble_sort1(L, I = I+1)

L = [int(input('enter element : ')) for I in range(int(input('enter the size : ')))]
for I in range(len(L)):
    bubble_sort1(L)
print('sorted : ',L)
```

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



```
Python 3.8.5 Shell
File Edit Shell Debug Options Window Help
Python 3.8.5 (tags/v3.8.5:580fbb0, Jul
tel)] on win32
Type "help", "copyright", "credits" or
>>>
===== RESTART: C:\Users\Toshiba\
enter the size : 5
enter element : 5
enter element : 4
enter element : 3
enter element : 2
enter element : 1
sorted : [1, 2, 3, 4, 5]
>>> |
```

## Ex.9

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

Examples :

Input: n = 16

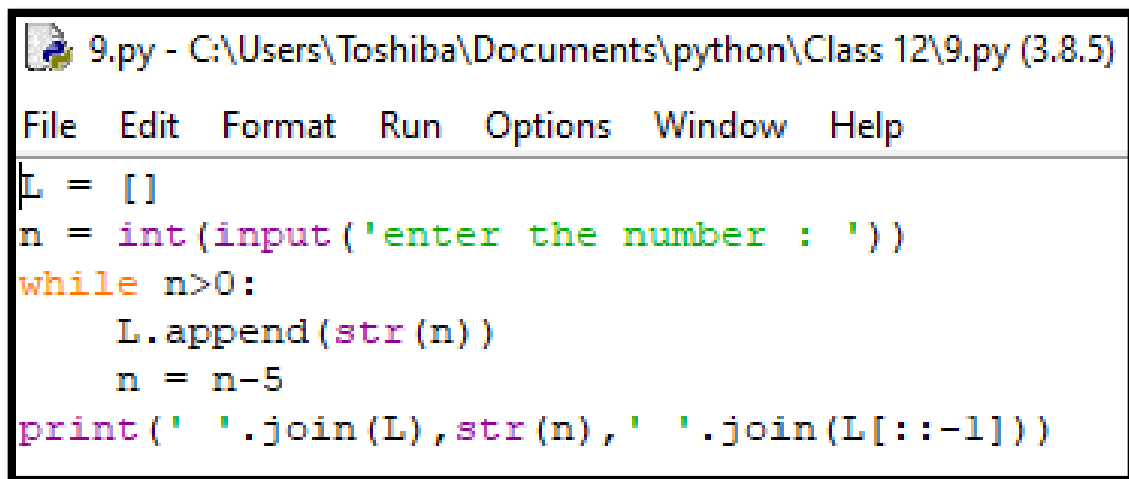
Output: 16, 11, 6, 1, -4, 1, 6, 11, 16

Input: n = 10

Output: 10, 5, 0, 5, 10

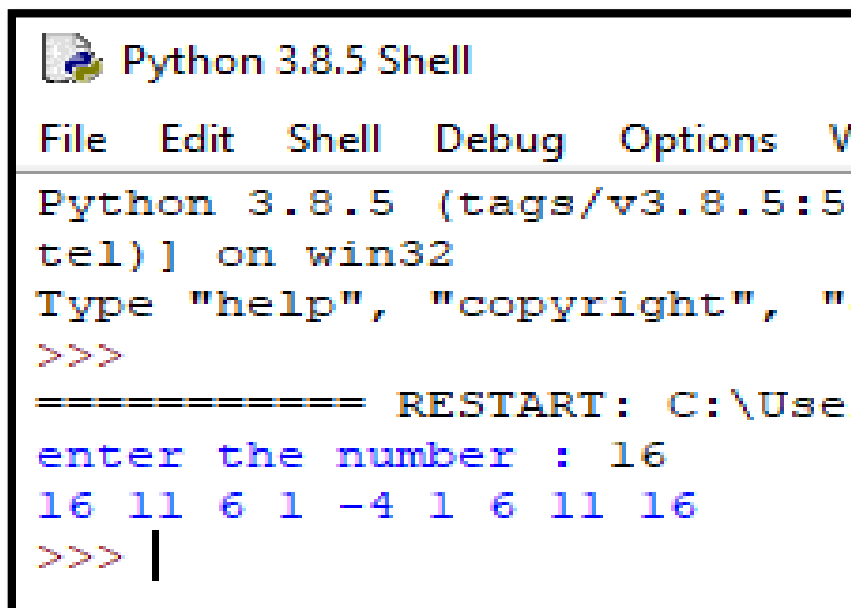
We basically first reduce 5 one by one until we reach a negative or 0. After we reach 0 or negative, we add 5 until we reach n.

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



```
9.py - C:\Users\Toshiba\Documents\python\Class 12\9.py (3.8.5)
File Edit Format Run Options Window Help
L = []
n = int(input('enter the number : '))
while n>0:
    L.append(str(n))
    n = n-5
print(' '.join(L),str(n), ' '.join(L[::-1]))
```

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



```
Python 3.8.5 Shell
File Edit Shell Debug Options V
Python 3.8.5 (tags/v3.8.5:5
tel)] on win32
Type "help", "copyright", "
>>>
===== RESTART: C:\Use
enter the number : 16
16 11 6 1 -4 1 6 11 16
>>> |
```

## 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

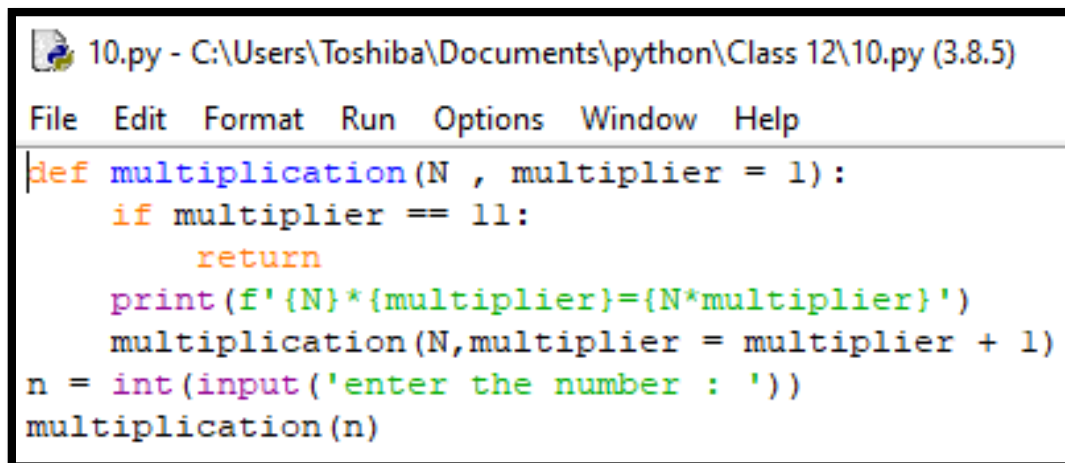
8\*7=56

8\*8=64

8\*9=72

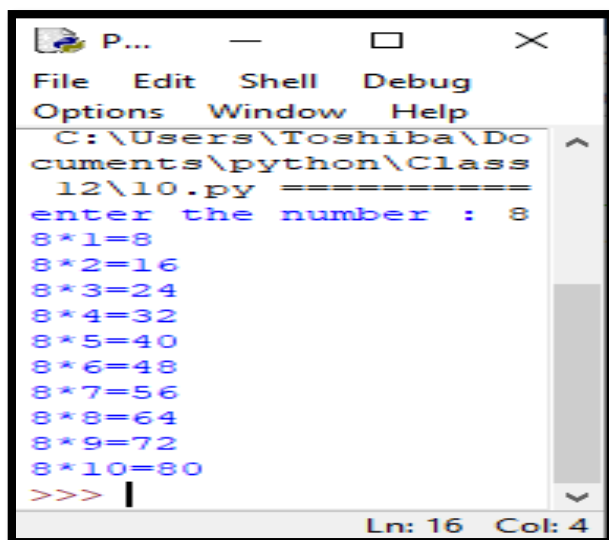
8\*10 = 80

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



```
10.py - C:\Users\Toshiba\Documents\python\Class 12\10.py (3.8.5)
File Edit Format Run Options Window Help
def multiplication(N , multiplier = 1):
    if multiplier == 11:
        return
    print(f'{N}*{multiplier}={N*multiplier}')
    multiplication(N,multiplier = multiplier + 1)
n = int(input('enter the number : '))
multiplication(n)
```

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



```
P...
File Edit Shell Debug
Options Window Help
C:\Users\Toshiba\Documents\python\Class 12\10.py =====
enter the number : 8
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
>>> |
Ln: 16 Col: 4
```

### **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\*\*\*\*\*

```
11.py - C:\Users\Toshiba\Documents\python\Class 12\11.py (3.8
File Edit Format Run Options Window Help
from random import choice
sections = ['A', 'B', 'C']
class_9 = open('class9.txt', 'r')
lines_9 = class_9.readlines()
limit = int(len(lines_9)/3) + 1
for I in range(3):
    f = open(f'class10{sections[I]}.txt', 'w')
    for I in range(limit):
        try:
            line = choice(lines_9)
            f.write(line)
            lines_9.remove(line)
        except:
            pass
    f.close()
```

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

```
class10A.txt - Notepad
File Edit Format View Help
Kartik Pathak 97.0
Gayathri 96.1
Samya Kazmi 95
Shifa Majeed 96
```

```
class10B.txt - Notepad
File Edit Format View Help
Rizwan Mohammed Numan 95
Akash Bandyopadhyay 98.4
Mayurnath 96.2
Anurag Joshi 98.3
```

```
class10C.txt - Notepad
File Edit Format View Help
Rohit 98.4
Pranav Nair 97.0
Dipayan Nag 96.5
```

## 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.

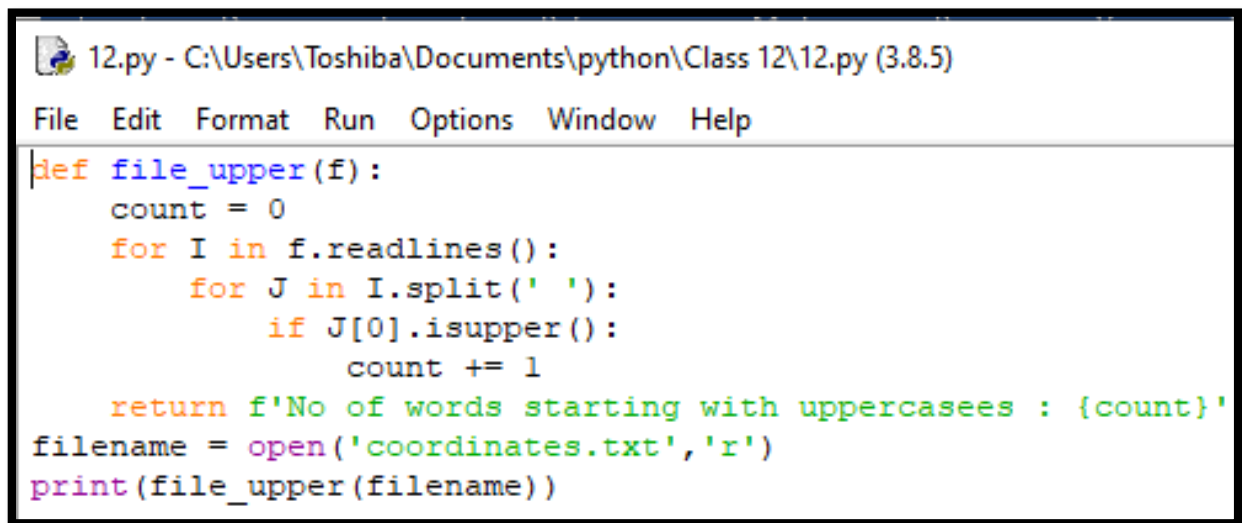
Do less complaining and more Attention to giving.

Do less criticizing and pay more Attention to Complementing.

Do Less talking and Pay more attention to SILIENCE.

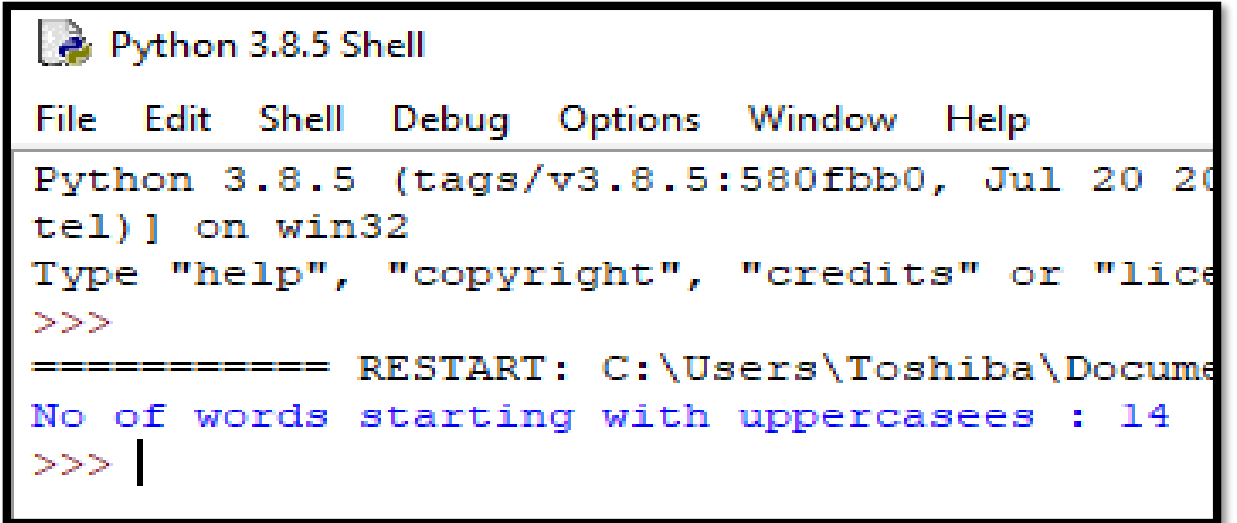
Write function in python to count the number of words having first character capital.

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



```
12.py - C:\Users\Toshiba\Documents\python\Class 12\12.py (3.8.5)
File Edit Format Run Options Window Help
def file_upper(f):
    count = 0
    for I in f.readlines():
        for J in I.split(' '):
            if J[0].isupper():
                count += 1
    return f'No of words starting with uppercasees : {count}'
filename = open('coordinates.txt','r')
print(file_upper(filename))
```

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



```
Python 3.8.5 Shell
File Edit Shell Debug Options Window Help
Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2019) [AMD64] on win32
Type "help", "copyright", "credits" or "license()"
>>>
===== RESTART: C:\Users\Toshiba\Documents\python\Class 12\12.py
No of words starting with uppercasees : 14
>>> |
```

### 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

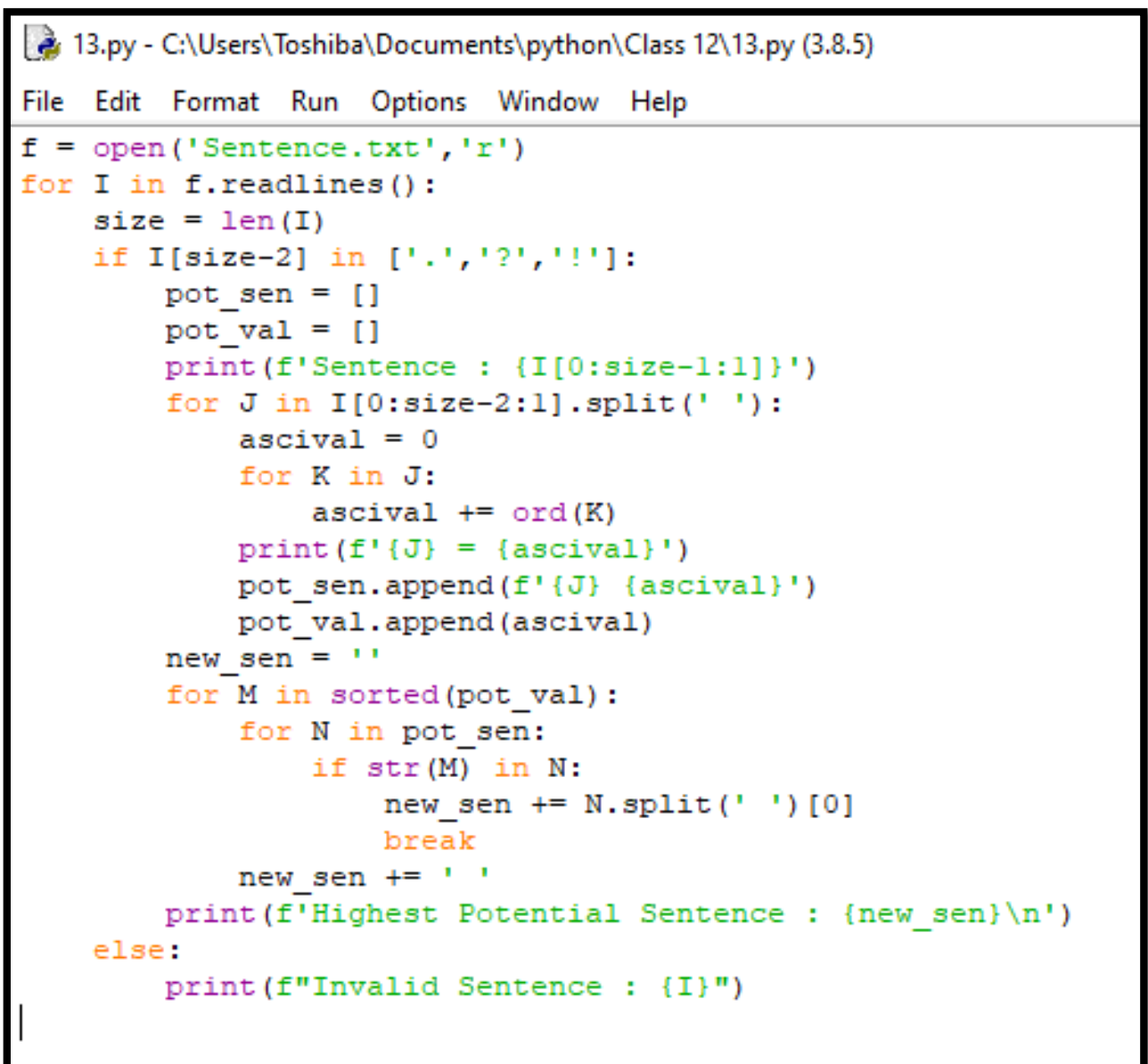
DO = 147

YOU = 253

DO = 147

DO DO HOW YOU

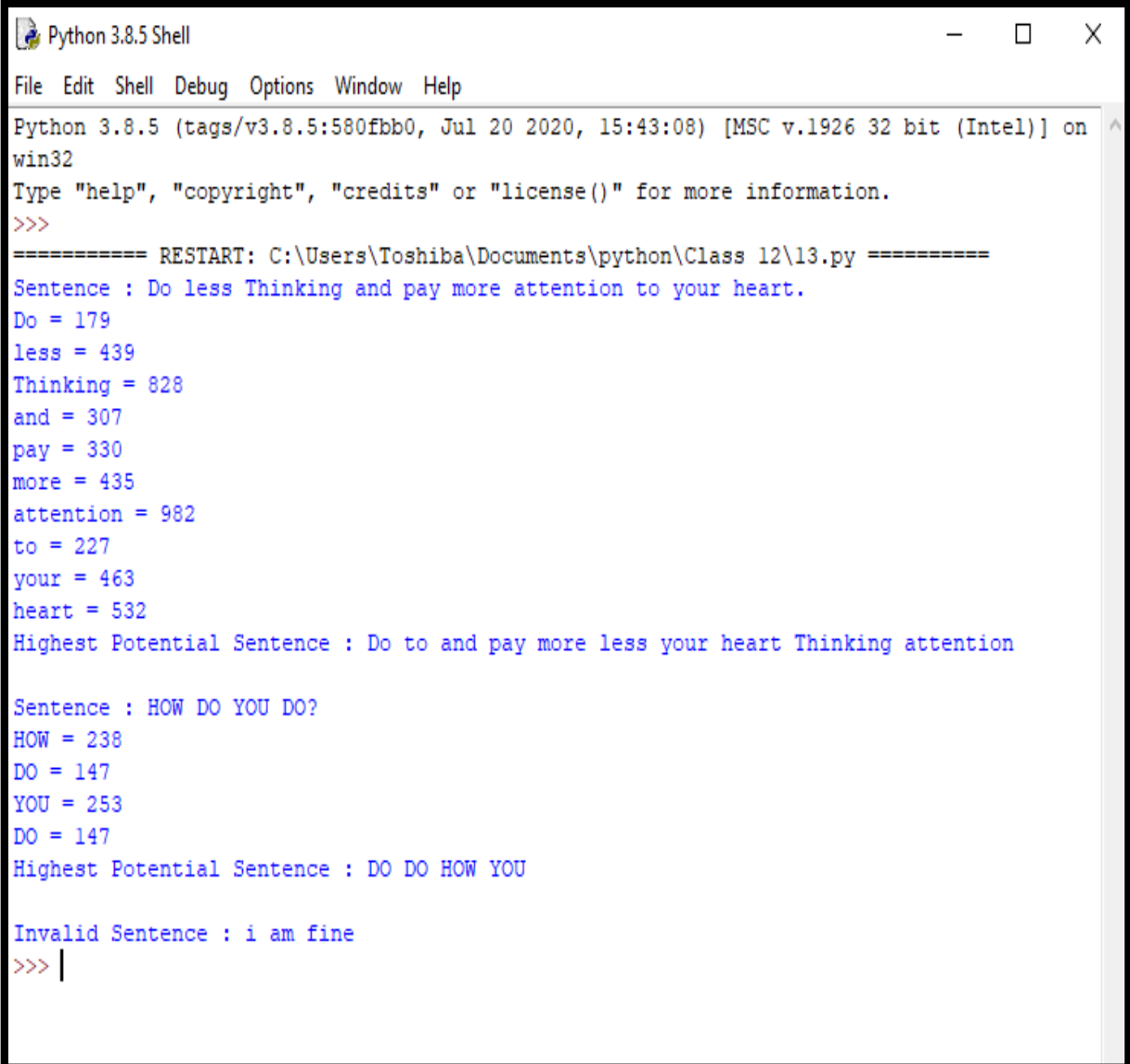
\*\*\*\*\* CODE\*\*\*\*\*



```
13.py - C:\Users\Toshiba\Documents\python\Class 12\13.py (3.8.5)
File Edit Format Run Options Window Help
f = open('Sentence.txt','r')
for I in f.readlines():
    size = len(I)
    if I[size-2] in ['.','?','!']:
        pot_sen = []
        pot_val = []
        print(f'Sentence : {I[0:size-1:1]}')
        for J in I[0:size-2:1].split(' '):
            ascival = 0
            for K in J:
                ascival += ord(K)
            print(f'{J} = {ascival}')
            pot_sen.append(f'{J} {ascival}')
            pot_val.append(ascival)
        new_sen = ''
        for M in sorted(pot_val):
            for N in pot_sen:
                if str(M) in N:
                    new_sen += N.split(' ')[0]
                    break
            new_sen += ' '
        print(f'Highest Potential Sentence : {new_sen}\n')
    else:
        print(f'Invalid Sentence : {I}')
```



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



```
Python 3.8.5 Shell
File Edit Shell Debug Options Window Help
Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020, 15:43:08) [MSC v.1926 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\Toshiba\Documents\python\Class 12\13.py =====
Sentence : Do less Thinking and pay more attention to your heart.
Do = 179
less = 439
Thinking = 828
and = 307
pay = 330
more = 435
attention = 982
to = 227
your = 463
heart = 532
Highest Potential Sentence : Do to and pay more less your heart Thinking attention

Sentence : HOW DO YOU DO?
HOW = 238
DO = 147
YOU = 253
DO = 147
Highest Potential Sentence : DO DO HOW YOU

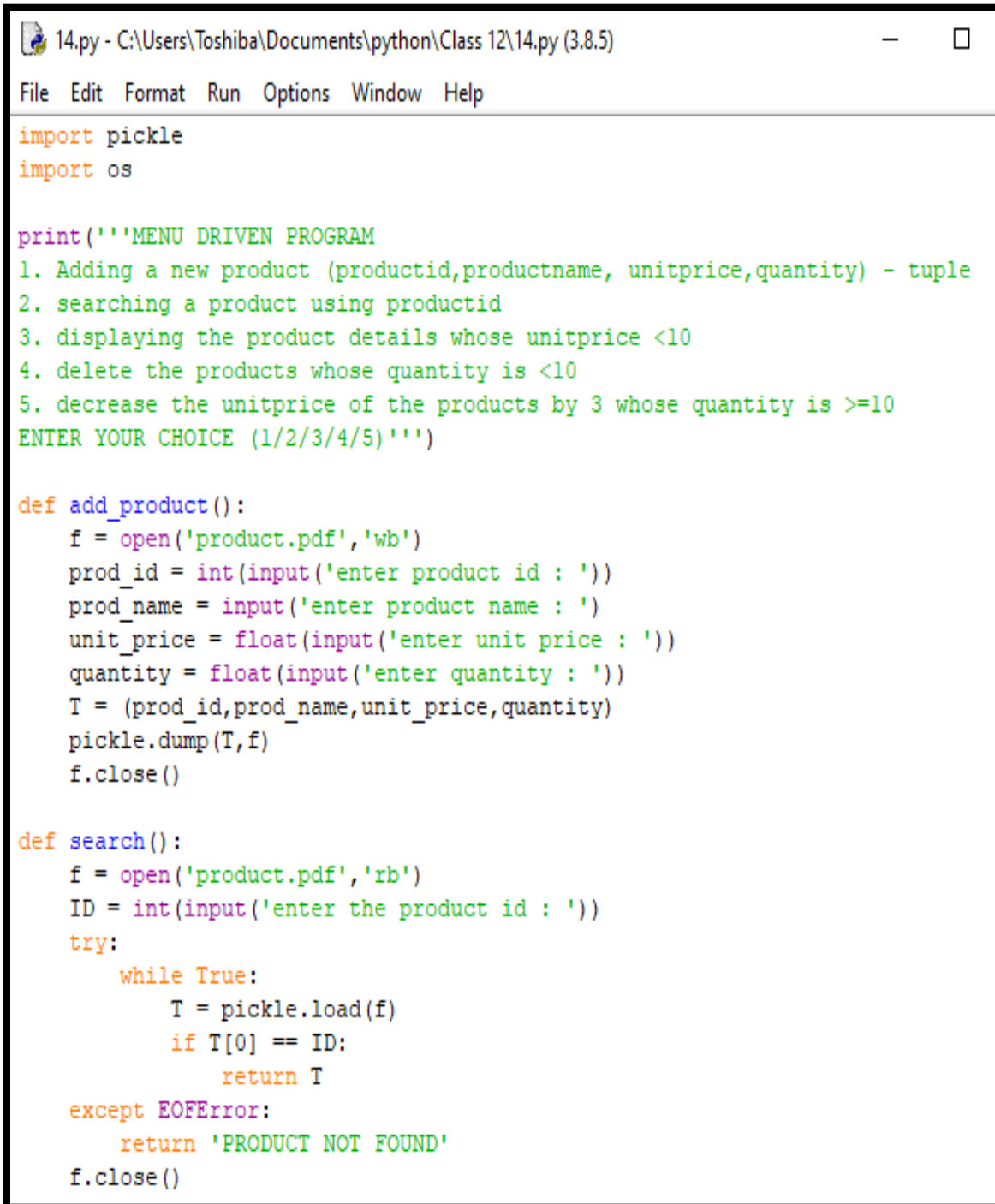
Invalid Sentence : i am fine
>>> |
```

## Ex.14

Write a menu driven program to perform 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\*\*\*\*\*



```
14.py - C:\Users\Toshiba\Documents\python\Class 12\14.py (3.8.5)
File Edit Format Run Options Window Help

import pickle
import os

print('''MENU DRIVEN PROGRAM
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
ENTER YOUR CHOICE (1/2/3/4/5)''')

def add_product():
    f = open('product.pdf','wb')
    prod_id = int(input('enter product id : '))
    prod_name = input('enter product name : ')
    unit_price = float(input('enter unit price : '))
    quantity = float(input('enter quantity : '))
    T = (prod_id,prod_name,unit_price,quantity)
    pickle.dump(T,f)
    f.close()

def search():
    f = open('product.pdf','rb')
    ID = int(input('enter the product id : '))
    try:
        while True:
            T = pickle.load(f)
            if T[0] == ID:
                return T
    except EOFError:
        return 'PRODUCT NOT FOUND'
    f.close()
```

```

def display():
    f = open('product.pdf','rb')
    try:
        while True:
            T = pickle.load(f)
            if T[2]<10:
                print(T)
    except EOFError:
        f.close()

def delete():
    f = open('product.pdf','rb')
    fl = open('temp.pdf','wb')
    try:
        while True:
            T = pickle.load(f)
            if T[3]<10:
                pickle.dump(T,fl)
    except EOFError:
        f.close()
        fl.close()
    os.remove('product.pdf')
    os.rename('temp.pdf','product.pdf')

def decrease():
    f = open('product.pdf','rb')
    fl = open('temp.pdf','wb')
    try:
        while True:
            T = pickle.load(f)
            if T[3]>10 or T[3]==10:
                T = (T[0],T[1],T[2]-3,T[3])
                pickle.dump(T,fl)
    except EOFError:
        f.close()
        fl.close()
    os.remove('product.pdf')
    os.rename('temp.pdf','product.pdf')

```

```

while True:
    choice = input()
    if choice == '1':
        add_product()
        print('PRODUCT ADDED SUCCESSFULLY')
    elif choice == '2':
        x = search()
        print(x)
    elif choice == '3':
        display()
        print('If nothing was displayed it means there were no such products')
    elif choice == '4':
        delete()
        print('OPERATION DONE SUCCESSFULLY')
    elif choice == '5':
        decrease()
        print('OPERATION DONE SUCCESSFULLY')
    done = input('DO YOU WANT TO PERFORM MORE OPERATIONS? (Y/N) : ')
    if done.upper() == 'N':
        break

```

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

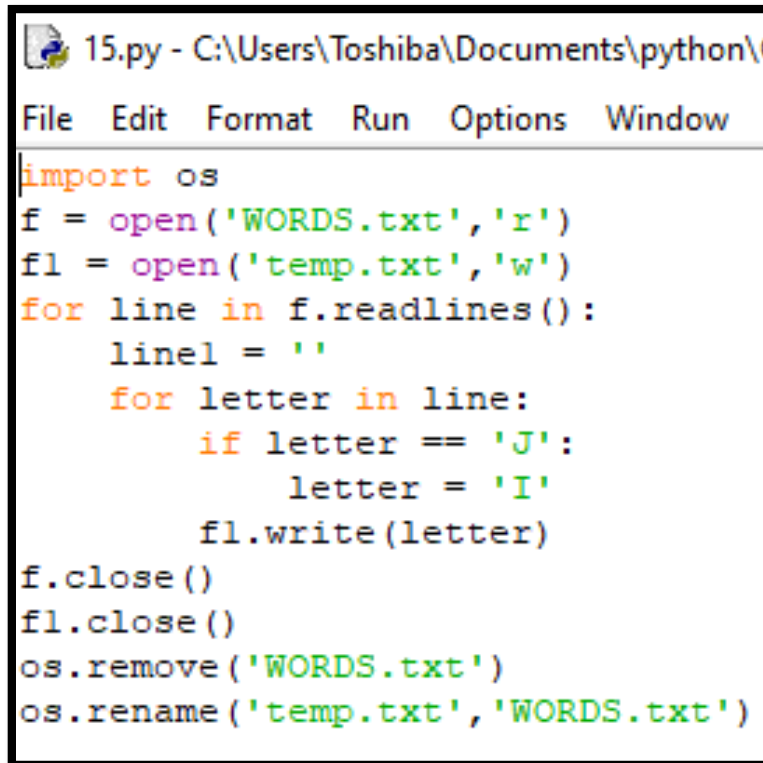
```
Python 3.8.5 Shell
File Edit Shell Debug Options Window Help
Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020, 15:43:08) [MSC v.1926 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\Toshiba\Documents\python\Class 12\14.py =====
MENU DRIVEN PROGRAM
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
ENTER YOUR CHOICE (1/2/3/4/5)
1
enter product id : 123
enter product name : rice
enter unit price : 12
enter quantity : 20
PRODUCT ADDED SUCCESSFULLY
DO YOU WANT TO PERFORM MORE OPERATIONS? (Y/N) : y
5
OPERATION DONE SUCCESSFULLY
DO YOU WANT TO PERFORM MORE OPERATIONS? (Y/N) : y
2
enter the product id : 123
(123, 'rice', 9.0, 20.0)
DO YOU WANT TO PERFORM MORE OPERATIONS? (Y/N) : y
3
(123, 'rice', 9.0, 20.0)
If nothing was displayed it means there were no such products
DO YOU WANT TO PERFORM MORE OPERATIONS? (Y/N) : y
4
OPERATION DONE SUCCESSFULLY
DO YOU WANT TO PERFORM MORE OPERATIONS? (Y/N) : y
3
If nothing was displayed it means there were no such products
DO YOU WANT TO PERFORM MORE OPERATIONS? (Y/N) : N
>>> |
```

## 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.

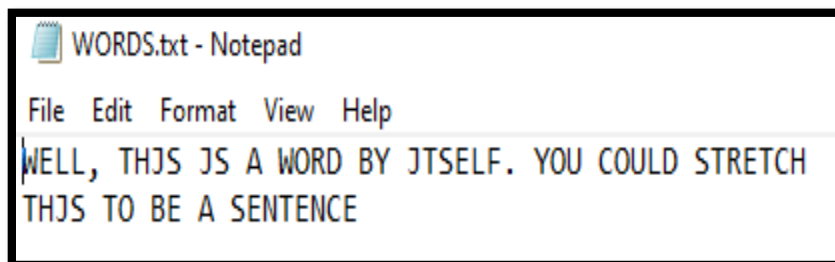
Note : Assuming that WORD.TXT does not contain any J alphabet otherwise.

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

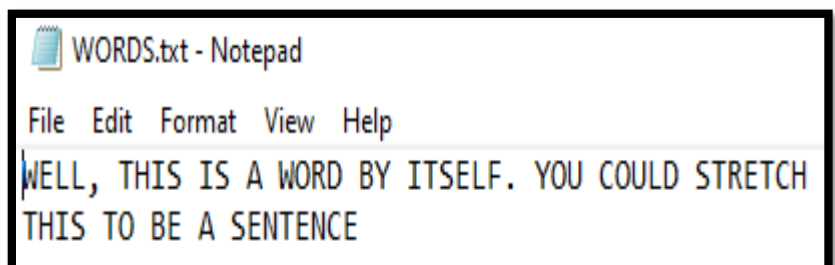


```
15.py - C:\Users\Toshiba\Documents\python\
File Edit Format Run Options Window
import os
f = open('WORDS.txt','r')
fl = open('temp.txt','w')
for line in f.readlines():
    line1 = ''
    for letter in line:
        if letter == 'J':
            letter = 'I'
        fl.write(letter)
f.close()
fl.close()
os.remove('WORDS.txt')
os.rename('temp.txt','WORDS.txt')
```

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



```
WORDS.txt - Notepad
File Edit Format View Help
WELL, THJS JS A WORD BY JTSELF. YOU COULD STRETCH
THJS TO BE A SENTENCE
```



```
WORDS.txt - Notepad
File Edit Format View Help
WELL, THIS IS A WORD BY ITSELF. YOU COULD STRETCH
THIS TO BE A SENTENCE
```

## **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\*\*\*\*\*

```
*16.py - C:\Users\Toshiba\Documents\python\Class 12\16.py (3.8.5)*
File Edit Format Run Options Window Help

import mysql.connector
#to check whether its connected
mydb=mysql.connector.connect(host='localhost',user='root',password='isgsql')
if mydb.is_connected()==False:
    print('not connected')
    raise SystemExit

#creating a cursor object
mycursor=mydb.cursor()

#using/creating database
try:
    mycursor.execute('create database Employee')
    mycursor.execute('use Employee')
except:
    mycursor.execute('use Employee')

def Insert_Initial():
    try:
        mycursor.execute('''create table Emp(Empno int primary key,
                                           Empname varchar(30),
                                           Salary int,
                                           Department varchar(30),
                                           Designation varchar(30))''')
    except:
        return #as table already exists
    rec_list = []
    for i in range(6):
        no=int(input('Enter employee no : '))
        name=input('Enter employee name : ')
        salary=float(input('Enter employee salary : '))
        department=input('Enter name of department : ')
        designation=input("Enter designation : ")
        rec_tuple=(no,name,salary,department,designation)
        rec_list.append(rec_tuple)
    command='insert into emp(empno,empname,salary,department,designation) values(%s,%s,%s,%s,%s)'
    mycursor.executemany(command,rec_list)
    print(mycursor.rowcount,'rows affected')
    mydb.commit()
Insert_Initial()
```

```

#menu-driven functions

def addrecord():
    record = (
        int(input('Enter employee no : ')),
        input('Enter employee name : '),
        float(input('Enter employee salary : ')),
        input('Enter name of department : '),
        input("Enter designation : ")
    )
    command='insert into emp values(%s,%s,%s,%s,%s)'
    mycursor.execute(command,record)
    mydb.commit()
    print('Operation successfull : record added')

def searchrecord():
    try:
        query = (
            int(input('Enter Employee no : ')),
            input('Enter dept : ')
        )
        command='select * from emp where Empno=%s and department=%s'
        mycursor.execute(command,query)
        records=mycursor.fetchall()
        for I in records:
            print(I)
    except:
        print('Record not found')

def updaterecord():
    query = (input('Enter Department to be updated : '),input('Enter Designation to be updated : '))
    command='update emp set salary=salary+0.35*salary where Department=%s and Designation=%s'
    try:
        mycursor.execute(command,query)
        mydb.commit()
        print('Record updated')
    except:
        print('Record not found')

```

```

def deleterecord():
    query = (input('Enter Department'),)
    command='delete from emp where Department=%s and Salary<15000'
    try:
        mycursor.execute(command,query)
        mydb.commit()
        print('Record deleted')
    except:
        print('Record not found')

def display():
    mycursor.execute('select * from emp')
    for I in mycursor.fetchall():
        print(I)

#menu-driven
print('''MENU
1. Add record
2. Search record
3. Update record
4. Delete record
5. Display Records
Press any other key to exit\n''')
while True:
    ch=input('Enter your choice :')
    if ch=='1':
        addrecord()
    elif ch=='2':
        searchrecord()
    elif ch=='3':
        updaterecord()
    elif ch=='4':
        deleterecord()
    elif ch=='5':
        display()
    else:
        raise SystemExit

```



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

```
Python 3.8.5 Shell
File Edit Shell Debug Options Window Help
Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20
tel)] on win32
Type "help", "copyright", "credits" or "li
>>>
===== RESTART: C:\Users\Toshiba\Docu
Enter employee no : 1
Enter employee name : Ayaan
Enter employee salary : 14000
Enter name of department : Sales
Enter designation : Clerk
Enter employee no : 2
Enter employee name : Amaan
Enter employee salary : 18000
Enter name of department : Sales
Enter designation : Manager
Enter employee no : 3
Enter employee name : Vivek
Enter employee salary : 21000
Enter name of department : Admin
Enter designation : Clerk
Enter employee no : 4
Enter employee name : Vrishab
Enter employee salary : 25000
Enter name of department : Admin
Enter designation : Manager
Enter employee no : 5
Enter employee name : Ramil
Enter employee salary : 23000
Enter name of department : Finance
Enter designation : Clerk
Enter employee no : 6
Enter employee name : Abizoi
Enter employee salary : 28000
Enter name of department : Finance
Enter designation : Manager
6 rows affected
```

## MENU

1. Add record
  2. Search record
  3. Update record
  4. Delete record
  5. Display Records
- Press any other key to exit

Enter your choice :5

```
(1, 'Ayaan', 14000, 'Sales', 'Clerk')
(2, 'Amaan', 18000, 'Sales', 'Manager')
(3, 'Vivek', 21000, 'Admin', 'Clerk')
(4, 'Vrishab', 25000, 'Admin', 'Manager')
(5, 'Ramil', 23000, 'Finance', 'Clerk')
(6, 'Abizoi', 28000, 'Finance', 'Manager')
```

Enter your choice :1

Enter employee no : 7

Enter employee name : Ronek

Enter employee salary : 20000

Enter name of department : Admin

Enter designation : Clerk

Operation successfull : record added

Enter your choice :2

Enter Employee no : 5

Enter dept : Finance

```
(5, 'Ramil', 23000, 'Finance', 'Clerk')
```

Enter your choice :3

Enter Department to be updated : Sales

Enter Designation to be updated : Manager

Record updated

Enter your choice :4

Enter DepartmentSales

Record deleted

Enter your choice :5

```
(2, 'Amaan', 24300, 'Sales', 'Manager')
(3, 'Vivek', 21000, 'Admin', 'Clerk')
(4, 'Vrishab', 25000, 'Admin', 'Manager')
(5, 'Ramil', 23000, 'Finance', 'Clerk')
(6, 'Abizoi', 28000, 'Finance', 'Manager')
(7, 'Ronek', 20000, 'Admin', 'Clerk')
```

Enter your choice :g

>>> |

## 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\*\*\*\*\*

```
17.py - C:\Users\Toshiba\Documents\python\Class 12\17.py (3.8.5)
File Edit Format Run Options Window Help
stack = [('D145', 'WimpyKid', 5000.0)] #stack of the form(Bookno, Bookname and Cost)
top = len(stack)

def push(stack):
    global top
    x = (input('Enter book no : '),
        input('Enter book name : '),
        float(input('Enter cost : ')))
    stack.append(x)
    top = len(stack)
    print('Length of the stack after push : ',top)

def pop(stack):
    global top
    if stack == []:
        print('Underflow alert : length of stack is : ',top)
        return
    stack.pop()
    top = len(stack)
    print('Length of the stack after pop : ',top)

def display(stack):
    global top
    print(stack)
    print('Length of the stack:',top)

print('''CHOOSE YOUR OPERATION BY TYPING THE NUMBER
1. PUSH INTO STACK
2. POP FROM STACK
3. DISPLAY STACK
Press any other key to exit''')

while True:
    choice = input('Enter choice : ')
    if choice == '1':
        push(stack)
    elif choice == '2':
        pop(stack)
    elif choice == '3':
        display(stack)
    else:
        raise SystemExit
```

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

```
Python 3.8.5 Shell
File Edit Shell Debug Options Window Help
Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020, 15:43:08) [MSC v.1926 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\Toshiba\Documents\python\Class 12\17.py =====
CHOOSE YOUR OPERATION BY TYPING THE NUMBER
    1. PUSH INTO STACK
    2. POP FROM STACK
    3. DISPLAY STACK
    Press any other key to exit
Enter choice : 1
Enter book no : E456
Enter book name : Harry Potter
Enter cost : 4000
Length of the stack after push : 2
Enter choice : 3
[('D145', 'WimpyKid', 5000.0), ('E456', 'Harry Potter', 4000.0)]
Length of the stack: 2
Enter choice : 2
Length of the stack after pop : 1
Enter choice : 3
[('D145', 'WimpyKid', 5000.0)]
Length of the stack: 1
Enter choice : h
>>> |
```

## **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

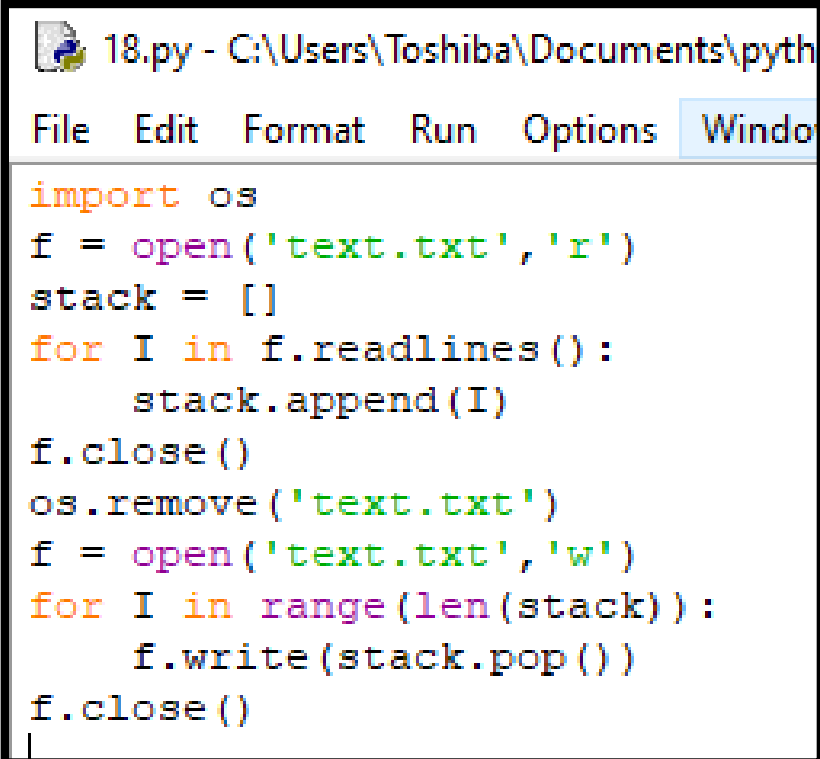
Output :

using stack  
content of a file  
to reverse the  
This is the program

Hint:

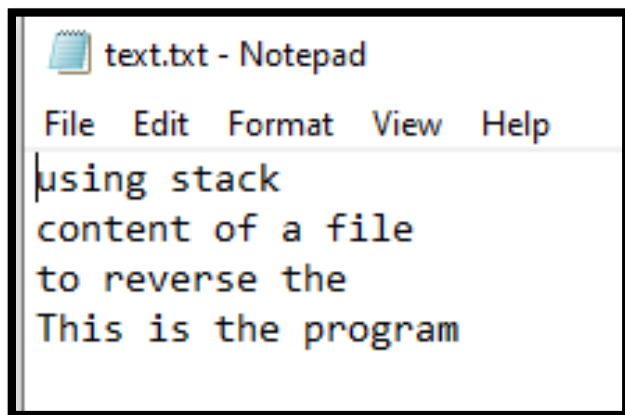
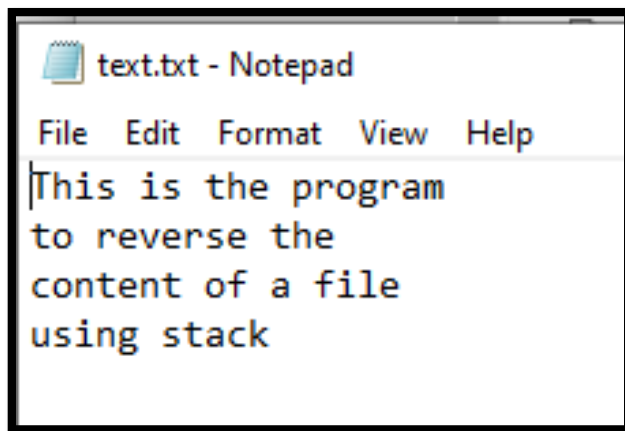
- ☐ Create an empty stack.
- ☐ One by one push every line of the file to the stack.
- ☐ One by one pop each line from the stack and put them back to the file.

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



```
18.py - C:\Users\Toshiba\Documents\pyth
File Edit Format Run Options Window
import os
f = open('text.txt', 'r')
stack = []
for I in f.readlines():
    stack.append(I)
f.close()
os.remove('text.txt')
f = open('text.txt', 'w')
for I in range(len(stack)):
    f.write(stack.pop())
f.close()
|
```

\*\*\*\*\*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)

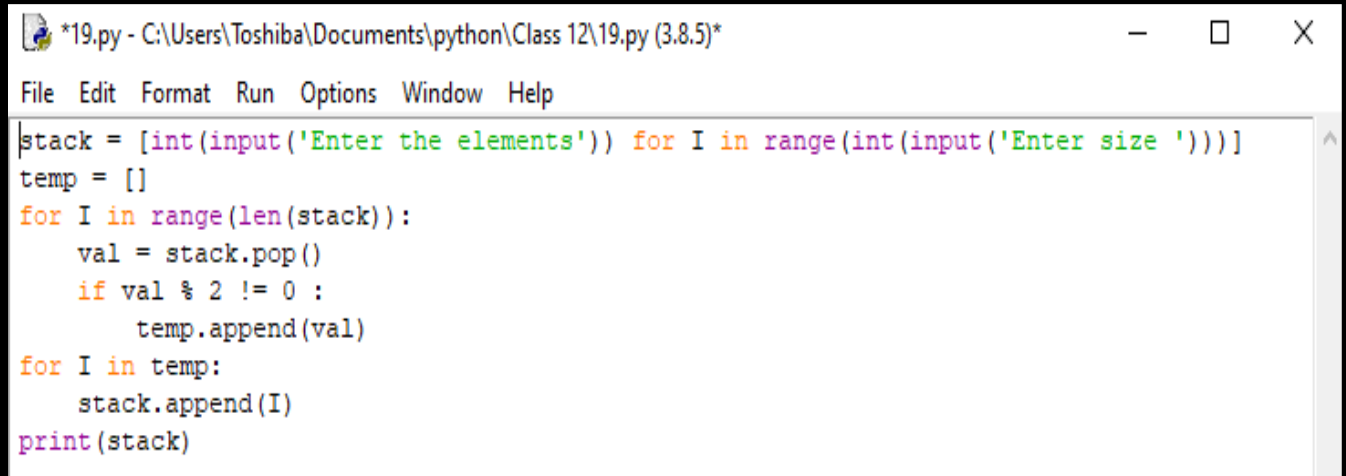
Output: 19 29 15

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

Approach:

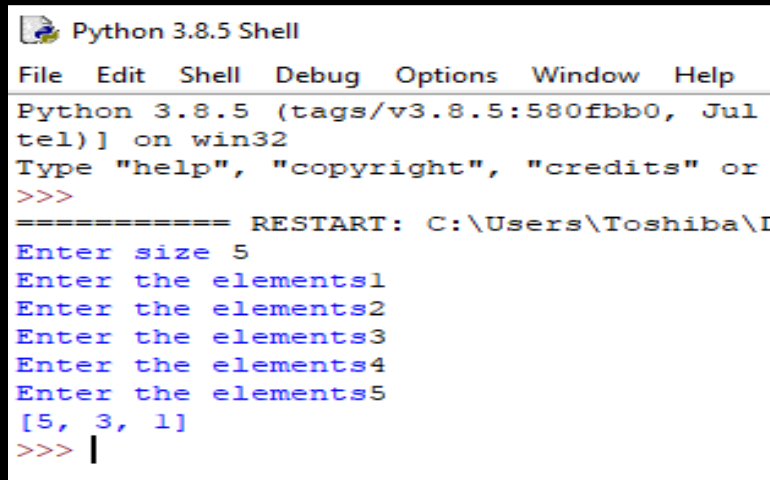
- Create a temporary stack temp and start popping the elements of the given stack s.
- For every popped element say val, if  $val \% 2 == 1$  then push it to temp.
- At the end of step 2, temp will contain all the odd elements from s but in reverse order.
- Now, to get the original order, pop every element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

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



```
*19.py - C:\Users\Toshiba\Documents\python\Class 12\19.py (3.8.5)*
File Edit Format Run Options Window Help
stack = [int(input('Enter the elements')) for I in range(int(input('Enter size ')))]
temp = []
for I in range(len(stack)):
    val = stack.pop()
    if val % 2 != 0 :
        temp.append(val)
for I in temp:
    stack.append(I)
print(stack)
```

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

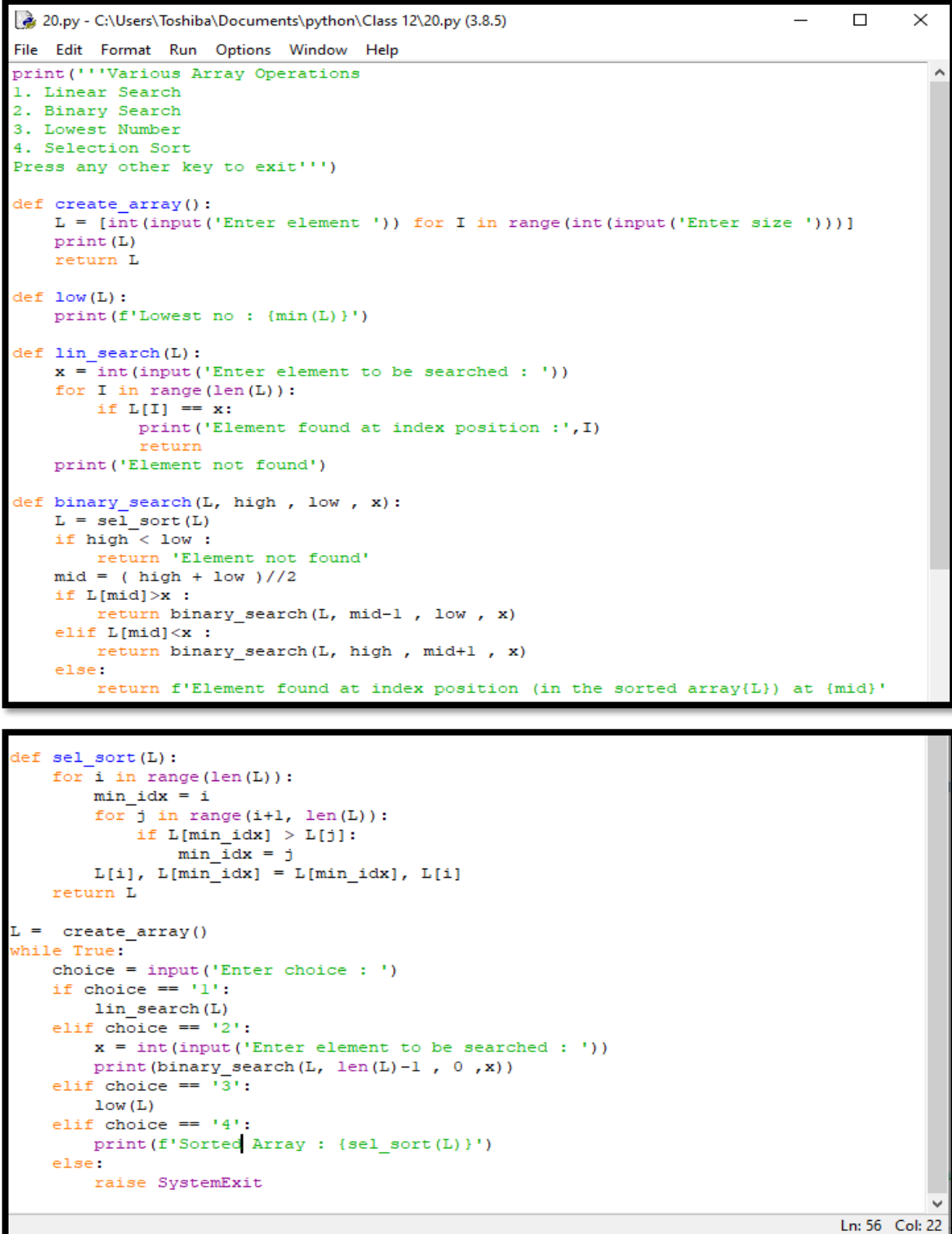


```
Python 3.8.5 Shell
File Edit Shell Debug Options Window Help
Python 3.8.5 (tags/v3.8.5:580fbb0, Jul
tel)] on win32
Type "help", "copyright", "credits" or
>>>
===== RESTART: C:\Users\Toshiba\D
Enter size 5
Enter the elements1
Enter the elements2
Enter the elements3
Enter the elements4
Enter the elements5
[5, 3, 1]
>>> |
```

## 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\*\*\*\*\*



```
20.py - C:\Users\Toshiba\Documents\python\Class 12\20.py (3.8.5)
File Edit Format Run Options Window Help
print('''Various Array Operations
1. Linear Search
2. Binary Search
3. Lowest Number
4. Selection Sort
Press any other key to exit''')

def create_array():
    L = [int(input('Enter element ')) for I in range(int(input('Enter size ')))]
    print(L)
    return L

def low(L):
    print(f'Lowest no : {min(L)}')

def lin_search(L):
    x = int(input('Enter element to be searched : '))
    for I in range(len(L)):
        if L[I] == x:
            print('Element found at index position :',I)
            return
    print('Element not found')

def binary_search(L, high , low , x):
    L = sel_sort(L)
    if high < low :
        return 'Element not found'
    mid = ( high + low )//2
    if L[mid]>x :
        return binary_search(L, mid-1 , low , x)
    elif L[mid]<x :
        return binary_search(L, high , mid+1 , x)
    else:
        return f'Element found at index position (in the sorted array{L}) at {mid}'

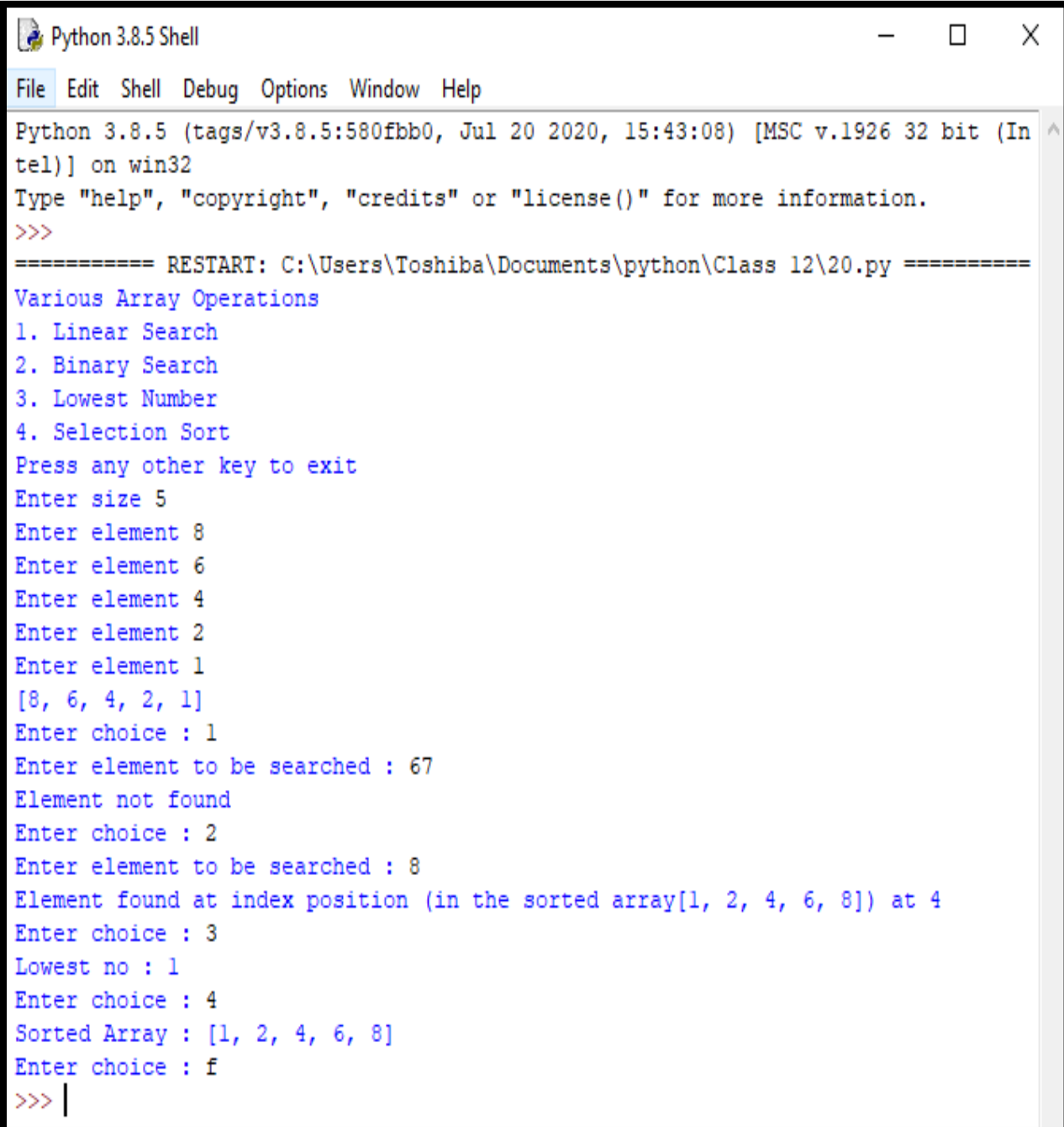
def sel_sort(L):
    for i in range(len(L)):
        min_idx = i
        for j in range(i+1, len(L)):
            if L[min_idx] > L[j]:
                min_idx = j
        L[i], L[min_idx] = L[min_idx], L[i]
    return L

L = create_array()
while True:
    choice = input('Enter choice : ')
    if choice == '1':
        lin_search(L)
    elif choice == '2':
        x = int(input('Enter element to be searched : '))
        print(binary_search(L, len(L)-1 , 0 ,x))
    elif choice == '3':
        low(L)
    elif choice == '4':
        print(f'Sorted Array : {sel_sort(L)}')
    else:
        raise SystemExit

Ln: 56 Col: 22
```



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



```
Python 3.8.5 Shell
File Edit Shell Debug Options Window Help
Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020, 15:43:08) [MSC v.1926 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\Toshiba\Documents\python\Class 12\20.py =====
Various Array Operations
1. Linear Search
2. Binary Search
3. Lowest Number
4. Selection Sort
Press any other key to exit
Enter size 5
Enter element 8
Enter element 6
Enter element 4
Enter element 2
Enter element 1
[8, 6, 4, 2, 1]
Enter choice : 1
Enter element to be searched : 67
Element not found
Enter choice : 2
Enter element to be searched : 8
Element found at index position (in the sorted array[1, 2, 4, 6, 8]) at 4
Enter choice : 3
Lowest no : 1
Enter choice : 4
Sorted Array : [1, 2, 4, 6, 8]
Enter choice : f
>>> |
```

## 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\*\*\*\*\*

```
21.py - C:\Users\Toshiba\Documents\python\Class 12\21.py (3.8.5)
File Edit Format Run Options Window Help
QUEUE = [(1122, 'Ayaan Jilani', 24)]
#of the form MemberNo(integer) , MemberName(String), Age(integer)

def insert(QUEUE):
    x = (int(input('Enter employee no : ')),
        input('Enter employee name : '),
        int(input('Enter age : ')))
    QUEUE.append(x)

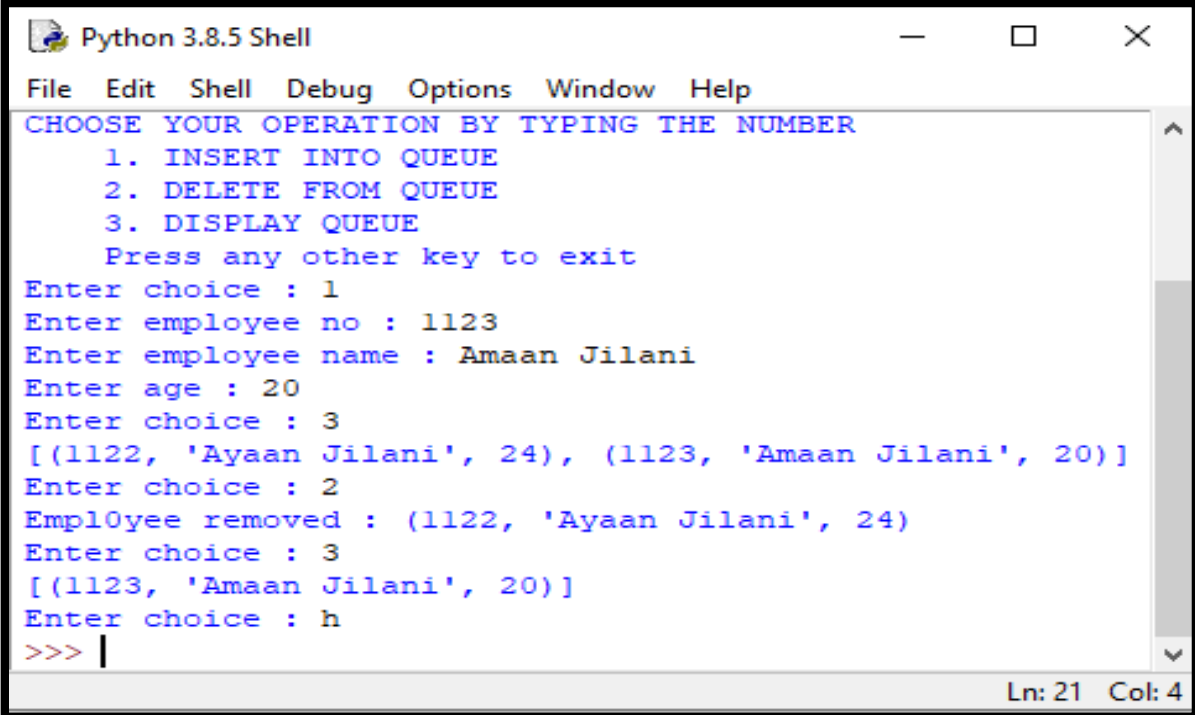
def delete(QUEUE):
    if QUEUE == []:
        print('Underflow alert')
        return
    popped = QUEUE.pop(0)
    print('Employee removed : ', popped)

def display(QUEUE):
    print(QUEUE)

print('''CHOOSE YOUR OPERATION BY TYPING THE NUMBER
1. INSERT INTO QUEUE
2. DELETE FROM QUEUE
3. DISPLAY QUEUE
Press any other key to exit''')

while True:
    choice = input('Enter choice : ')
    if choice == '1':
        insert(QUEUE)
    elif choice == '2':
        delete(QUEUE)
    elif choice == '3':
        display(QUEUE)
    else:
        raise SystemExit
```

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



The image shows a screenshot of a Python 3.8.5 Shell window. The window has a title bar with the text "Python 3.8.5 Shell" and standard window controls (minimize, maximize, close). Below the title bar is a menu bar with the following items: File, Edit, Shell, Debug, Options, Window, and Help. The main area of the window displays the following text in a monospaced font:

```
CHOOSE YOUR OPERATION BY TYPING THE NUMBER
1. INSERT INTO QUEUE
2. DELETE FROM QUEUE
3. DISPLAY QUEUE
Press any other key to exit
Enter choice : 1
Enter employee no : 1123
Enter employee name : Amaan Jilani
Enter age : 20
Enter choice : 3
[(1122, 'Ayaan Jilani', 24), (1123, 'Amaan Jilani', 20)]
Enter choice : 2
Employee removed : (1122, 'Ayaan Jilani', 24)
Enter choice : 3
[(1123, 'Amaan Jilani', 20)]
Enter choice : h
>>> |
```

At the bottom right of the window, there is a status bar that reads "Ln: 21 Col: 4".

## 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\*\*\*\*\*



```
22.py - C:\Users\Toshiba\Documents\python\Class 12\22.py (3.8.5)
File Edit Format Run Options Window Help
VIP = [111]
Balcony = [222]
Regular = [333]

def insert(tokenId,priority):
    if priority.upper() == 'H':
        VIP.append(tokenId)
    elif priority.upper() == 'N':
        Balcony.append(tokenId)
    elif priority.upper() == 'L':
        Regular.append(tokenId)
    print('Operation Successful')

def search(t):
    if (t not in VIP) and (t not in Balcony) and (t not in Regular):
        print('invalid token')
        return
    if t in VIP:
        print('tokenId found in highest priority (VIP)')
    elif t in Balcony:
        print('tokenId found in normal priority (Balcony)')
    elif t in Regular:
        print('tokenId found in lowest priority (Regular)')

def change(t,p):
    if (t not in VIP) and (t not in Balcony) and (t not in Regular):
        print('invalid token')
        return
    if t in VIP:
        VIP.remove(t)
    elif t in Balcony:
        Balcony.remove(t)
    elif t in Regular:
        Regular.remove(t)
    insert(t,p)
```

```

def display():
    print(VIP)
    print(Balcony)
    print(Regular)

print('''CHOOSE THE FOLLOWING OPERATIONS
1. Insert tokenId
2. Search for an Id
3. Change Priority
4. Display''')

while True:
    choice = input('Enter choice : ')
    if choice == '1':
        tokenId = int(input('Enter tokenId : '))
        priority = input('Priority(Highest/Normal/lowest(H/N/L): ')
        insert(tokenId,priority)
    elif choice == '2':
        token = int(input('Enter tokenId : '))
        search(token)
    elif choice == '3':
        token = int(input('Enter tokenId : '))
        priority = input('Priority(Highest/Normal/lowest(H/N/L): ')
        change(token,priority)
    elif choice == '4':
        display()
    else:
        raise SystemExit

```

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

```

Python 3.8.5 Shell
File Edit Shell Debug Options Window Help
Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020, 15:43:
tel)] on win32
Type "help", "copyright", "credits" or "license()" fo
>>>
===== RESTART: C:\Users\Toshiba\Documents\pytho
CHOOSE THE FOLLOWING OPERATIONS
1. Insert tokenId
2. Search for an Id
3. Change Priority
4. Display
Enter choice : 1
Enter tokenId : 112
Priority(Highest/Normal/lowest(H/N/L): H
Operation Successful
Enter choice : 2
Enter tokenId : 222
tokenId found in normal priority (Balcony)
Enter choice : 3
Enter tokenId : 333
Priority(Highest/Normal/lowest(H/N/L): H
Operation Successful
Enter choice : 4
[111, 112, 333]
[222]
[]
Enter choice : y
>>> |

```

## 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\*\*\*\*\*

```
23.py - C:\Users\Toshiba\Documents\python\Class 12\23.py (3.8.5)
File Edit Format Run Options Window Help

print('''MENU DRIVEN PROGRAM
1. ADD STUDENT
2. DISPLAY STUDENT
3. SEARCH STUDENT
Press any other key to exit''')

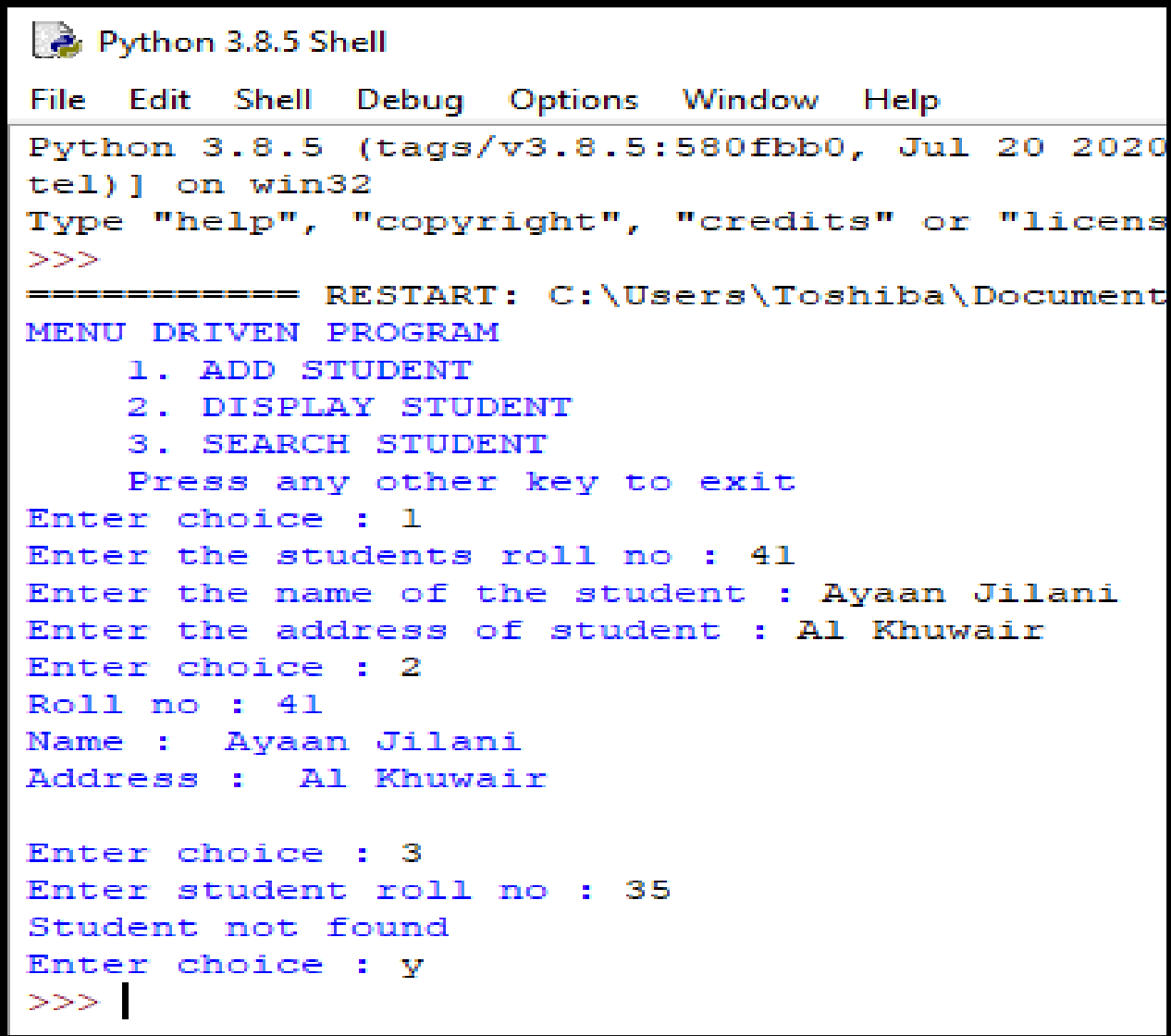
def add_stud(file):
    roll_no = int(input('Enter the students roll no : '))
    name = input('Enter the name of the student : ')
    address = input('Enter the address of student : ')
    file.write(f"{roll_no} , {name} , {address}\n")

def disp_stud(file):
    for I in file.readlines():
        print(f'''Roll no : {I.split(',')[0]}
Name : {I.split(',')[1]}
Address : {I.split(',')[2]} ''')

def search_stud(file):
    search_no = input('Enter student roll no : ')
    for I in file.readlines():
        if search_no in I.split(',')[0]:
            print(f'''Roll no : {I.split(',')[0]}
Name : {I.split(',')[1]}
Address : {I.split(',')[2]} ''')
            return
    print('Student not found')

while True:
    choice = input('Enter choice : ')
    if choice == '1':
        add_stud(open('student.txt','a+'))
    elif choice == '2':
        disp_stud(open('student.txt','r'))
    elif choice == '3':
        search_stud(open('student.txt','r'))
    else:
        raise SystemExit
```

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



```
Python 3.8.5 Shell
File Edit Shell Debug Options Window Help
Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020
tel)] on win32
Type "help", "copyright", "credits" or "licens
>>>
===== RESTART: C:\Users\Toshiba\Document
MENU DRIVEN PROGRAM
    1. ADD STUDENT
    2. DISPLAY STUDENT
    3. SEARCH STUDENT
    Press any other key to exit
Enter choice : 1
Enter the students roll no : 41
Enter the name of the student : Ayaan Jilani
Enter the address of student : Al Khuwair
Enter choice : 2
Roll no : 41
Name : Ayaan Jilani
Address : Al Khuwair

Enter choice : 3
Enter student roll no : 35
Student not found
Enter choice : y
>>> |
```

## Ex.24

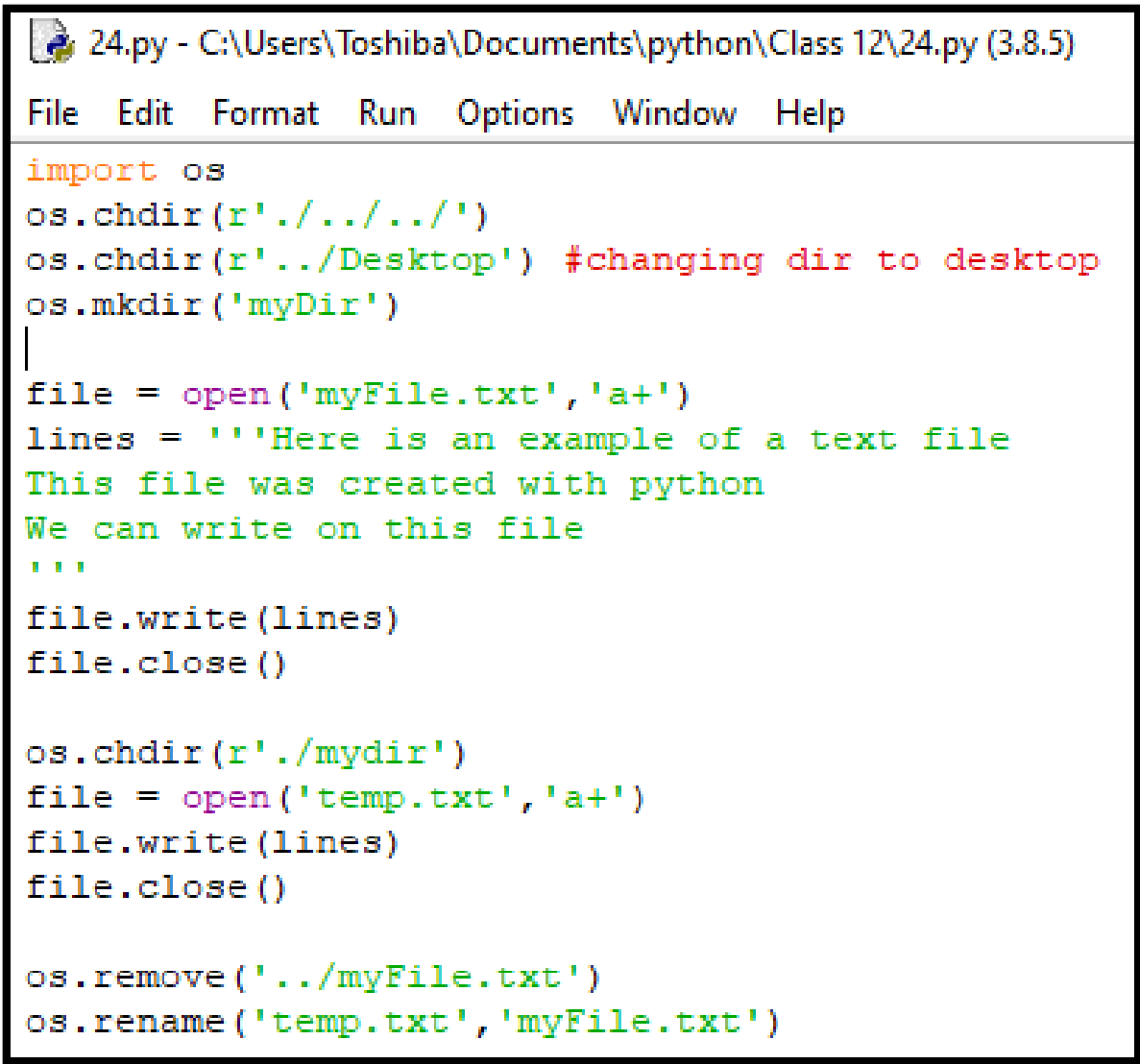
Write a python program that allowing you to create a directory in the desktop called myDir.

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\*\*\*\*\*



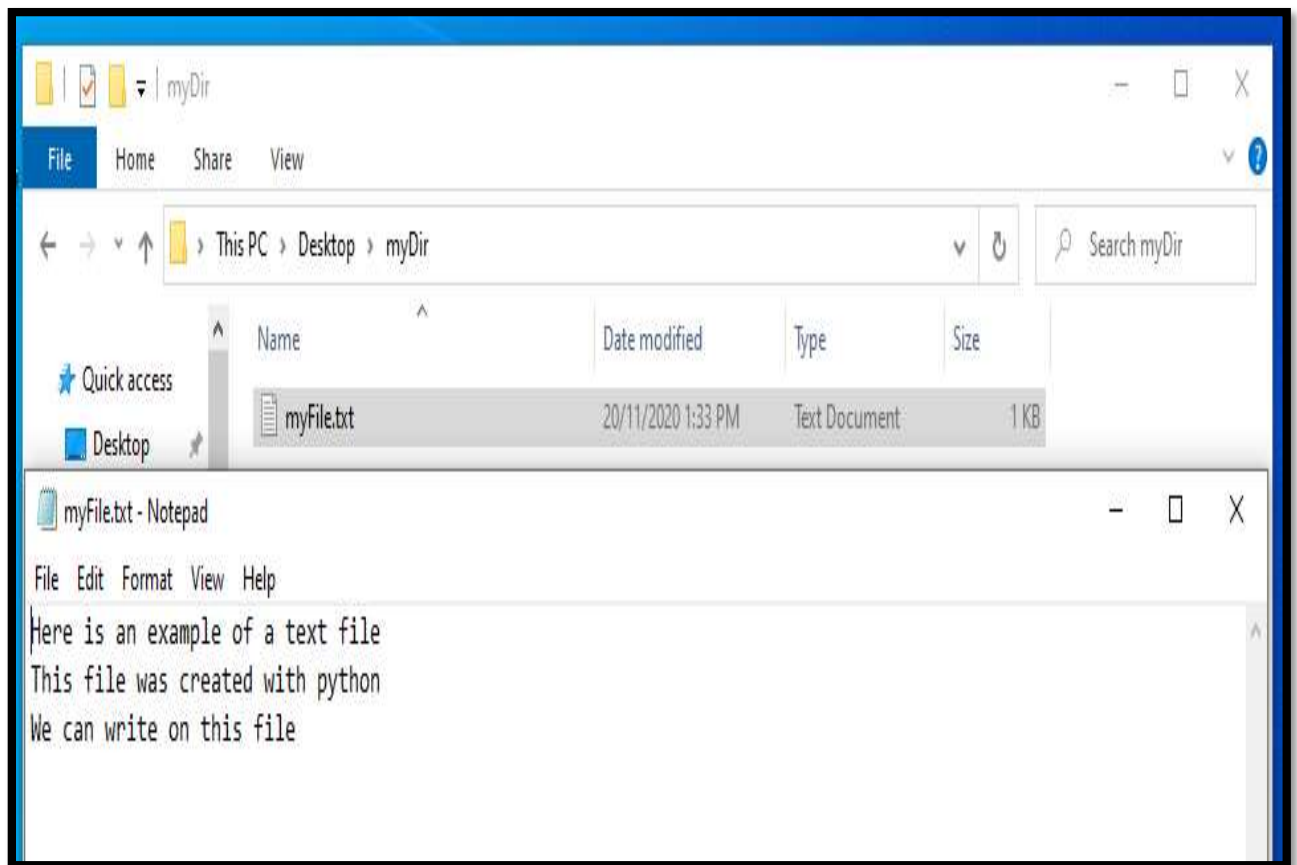
```
import os
os.chdir(r'../../..')
os.chdir(r'../Desktop') #changing dir to desktop
os.mkdir('myDir')
|
file = open('myFile.txt', 'a+')
lines = '''Here is an example of a text file
This file was created with python
We can write on this file
'''
file.write(lines)
file.close()

os.chdir(r'../mydir')
file = open('temp.txt', 'a+')
file.write(lines)
file.close()

os.remove('../myFile.txt')
os.rename('temp.txt', 'myFile.txt')
```



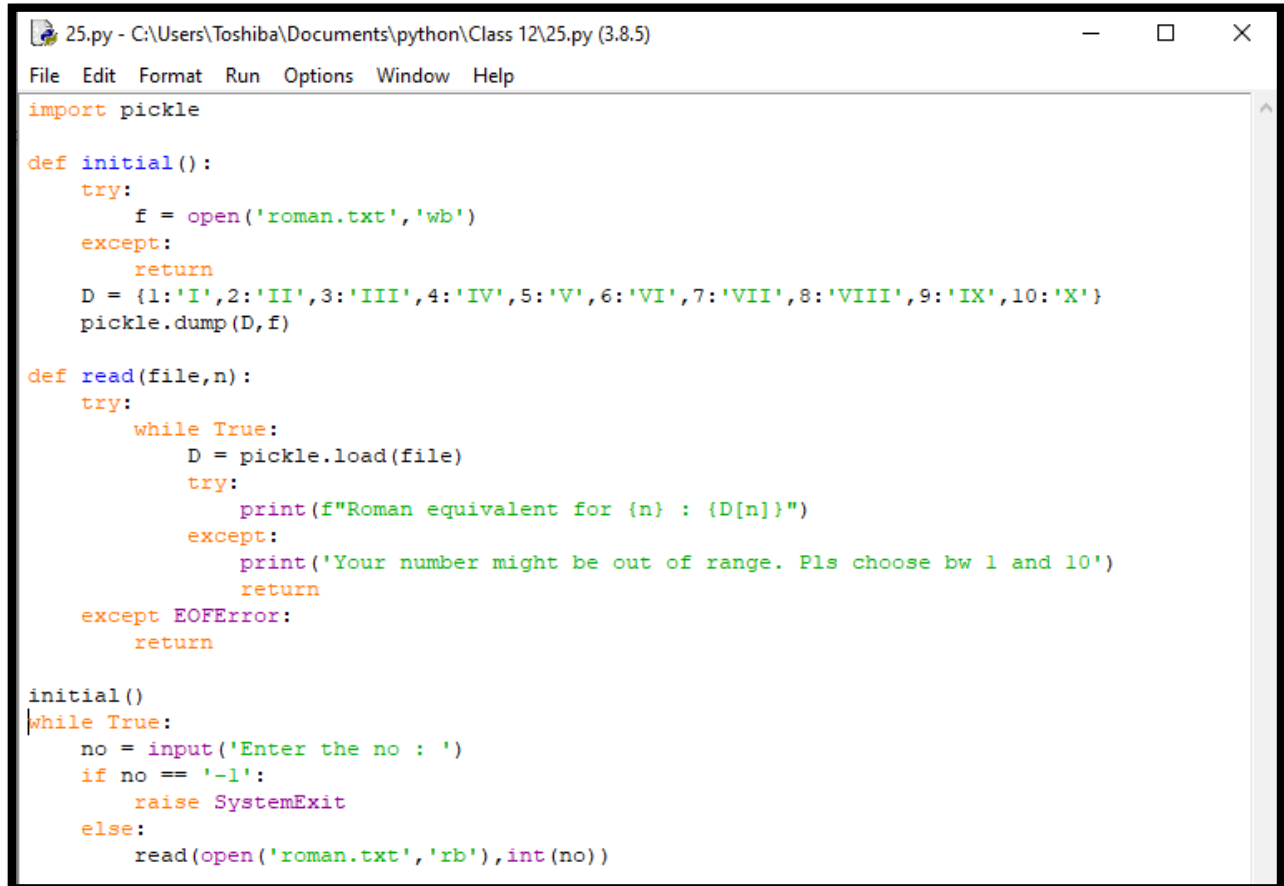
\*\*\*\*\*OUTPUT\*\*\*\*\*



## 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\*\*\*\*\*

A screenshot of a Python IDE window titled '25.py - C:\Users\Toshiba\Documents\python\Class 12\25.py (3.8.5)'. The window contains the following Python code:

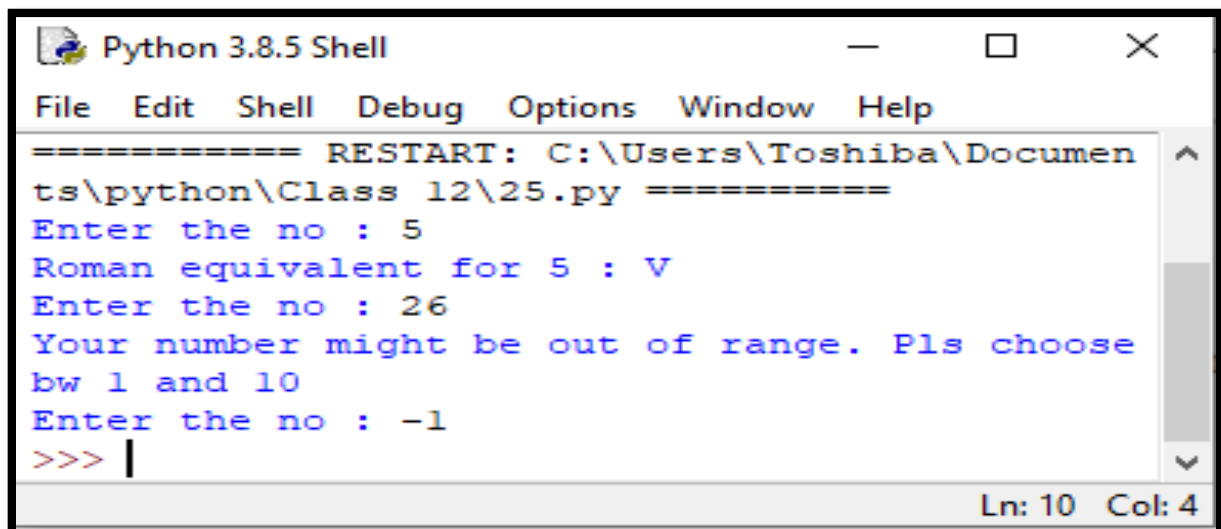
```
import pickle

def initial():
    try:
        f = open('roman.txt', 'wb')
    except:
        return
    D = {1:'I', 2:'II', 3:'III', 4:'IV', 5:'V', 6:'VI', 7:'VII', 8:'VIII', 9:'IX', 10:'X'}
    pickle.dump(D, f)

def read(file, n):
    try:
        while True:
            D = pickle.load(file)
            try:
                print(f"Roman equivalent for {n} : {D[n]}")
            except:
                print('Your number might be out of range. Pls choose bw 1 and 10')
            return
    except EOFError:
        return

initial()
while True:
    no = input('Enter the no : ')
    if no == '-1':
        raise SystemExit
    else:
        read(open('roman.txt', 'rb'), int(no))
```

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

A screenshot of a Python 3.8.5 Shell window titled 'Python 3.8.5 Shell'. The window shows the output of the program:

```
==== RESTART: C:\Users\Toshiba\Documents\python\Class 12\25.py =====
Enter the no : 5
Roman equivalent for 5 : V
Enter the no : 26
Your number might be out of range. Pls choose bw 1 and 10
Enter the no : -1
>>> |
```

The status bar at the bottom right indicates 'Ln: 10 Col: 4'.

## 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					
Field	Type	Null	Key	Default	Extra
RegNo	char(10)	NO	PRI		
RegDate	date	YES		NULL	
Owner	varchar(30)	YES		NULL	
Address	varchar(50)	YES		NULL	

CHALLAN					
Field	Type	Null	Key	Default	Extra
Challan_No	int(11)	NO	PRI	0	
Ch_Date	date	YES		NULL	
RegNo	char(10)	YES		NULL	
Offence	int(3)	YES		NULL	

OFFENCE					
Field	Type	Null	Key	Default	Extra
Offence_Code	int(3)	NO	PRI	0	
Off_desc	varchar(30)	YES		NULL	
Challan_Amt	int(4)	YES		NULL	

## VEHICLE

regno	regdate	owner	address
12333AD445	2009-06-07	srk	ghubra
12333AD446	2018-05-17	aamir	seeb
12333AD447	2013-04-27	hrithik	seeb
126C727272	2015-06-07	salman	al khuwair

## CHALLAN

challan_no	ch_date	regno	offence
1	2015-05-26	12333AD447	100
2	2020-05-26	12333AD446	101
3	2019-05-26	12333AD446	102
4	2019-08-26	12333AD445	102
5	2019-08-26	12333AD444	102

## OFFENCE

offence_code	offense_desc	challan_amt
100	overspeeding	1000
101	driving without license	1500
102	wrong parking	2000

3 rows in set (0.00 sec)

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

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

```
mysql> select max(regdate),min(regdate) from vehicle;
+-----+-----+
| max(regdate) | min(regdate) |
+-----+-----+
| 2020-11-04   | 2013-04-27   |
+-----+-----+
1 row in set (0.00 sec)
```

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

```
mysql> select count(challan_no),ch_date from challan group by ch_date;
+-----+-----+
| count(challan_no) | ch_date      |
+-----+-----+
| 1                 | 2015-05-26   |
| 1                 | 2020-05-26   |
| 1                 | 2019-05-26   |
| 1                 | 2010-04-03   |
| 1                 | 2019-08-26   |
+-----+-----+
5 rows in set (0.00 sec)
```

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

```
mysql> select count(offence),offense_desc,offence from c
hallan C,offense O where C.offence=O.offence_code group
by offence;
+-----+-----+-----+
| count(offence) | offense_desc          | offence |
+-----+-----+-----+
| 1              | speeding              | 100       |
| 1              | driving without license | 101       |
| 3              | wrong parking         | 102       |
+-----+-----+-----+
3 rows in set (0.00 sec)
```

4. Select count(regno) from vehicle where regno like '\_\_6C%';

```
mysql> select count(regno) from vehicle where regno like '__6C%';
+-----+
| count(regno) |
+-----+
|             1 |
+-----+
1 row in set (0.00 sec)
```

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

```
mysql> select sum(challan_amt) from challan C,offense O
where C.offence=O.offence_code and offence=101 group by
offence;
+-----+
| sum(challan_amt) |
+-----+
|             1500 |
+-----+
1 row in set (0.00 sec)
```

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

```
mysql> select regno,offence,offense_desc,challan_amt from challan C,offense O where C.offence=
O.offence_code and ch_date='2010-04-03';
+-----+-----+-----+-----+
| regno   | offence | offense_desc | challan_amt |
+-----+-----+-----+-----+
| 12333AD445 |      102 | wrong parking |          2000 |
+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

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

```
mysql> select regno from challan group by regno having count(regno)>1;
+-----+
| regno   |
+-----+
| 12333AD446 |
+-----+
1 row in set (0.00 sec)
```

## 8. 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;

```
mysql> 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;
```

regno	regdate	owner	address	ch_date	offence	offense_desc	challan_amt
12333AD447	2013-04-27	hrithik	seeb	2015-05-26	100	overspeeding	1000
12333AD446	2018-05-17	aamir	seeb	2020-05-26	101	driving without license	1500
12333AD446	2018-05-17	aamir	seeb	2019-05-26	102	wrong parking	2000
12333AD445	2009-06-07	srk	ghubra	2010-04-03	102	wrong parking	2000
126C727272	2015-06-07	salman	al khuwair	2019-08-26	102	wrong parking	2000

5 rows in set (0.00 sec)



## Ex.27

Table: Employee

No	Name	Salary	Zone	Age	Grade	Dept
1	Mukul	30000	West	28	A	10
2	Kritika	35000	Centre	30	A	10
3	Naveen	32000	West	40		20
4	Uday	38000	North	38	C	30
5	Nupur	32000	East	26		20
6	Moksh	37000	South	28	B	10
7	Shelly	36000	North	26	A	30

Table: Department

Dept	DName	MinSal	MaxSal	HOD
10	Sales	25000	32000	1
20	Finance	30000	50000	5
30	Admin	25000	40000	7

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

- Display the details of all the employees who work in Sales department.
- Display the Salary, Zone, and Grade of all the employees whose HOD is Nupur.
- Display the Name and Department Name of all the employees.
- Display the names of all the employees whose salary is not within the specified range for the corresponding department.
- Display the name of the department and the name of the corresponding HOD for all the departments.

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

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

```
mysql> select * from employee where dept=10;
+----+-----+-----+-----+-----+-----+-----+
| no  | name   | salary | zone   | age  | grade | dept |
+----+-----+-----+-----+-----+-----+-----+
| 1   | mukul  | 30000  | west   | 28   | A     | 10   |
| 2   | kritika | 35000  | centre | 30   | A     | 10   |
| 6   | moksh  | 37000  | south  | 28   | B     | 10   |
+----+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```



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

```
mysql> select name,salary,zone,grade from employee where dept=20 and no!=5;
+-----+-----+-----+
| name   | salary | zone  | grade |
+-----+-----+-----+
| naveen | 32000  | west  | NULL  |
+-----+-----+-----+
1 row in set (0.00 sec)
```

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

```
mysql> select name,Dname from employee E,department D where E.dept=D.dept;
+-----+-----+
| name   | Dname  |
+-----+-----+
| mukul  | sales  |
| kritika | sales  |
| naveen  | finance |
| uday    | admin  |
| nupur   | finance |
| moksh   | sales  |
| shetty  | admin  |
+-----+-----+
7 rows in set (0.00 sec)
```

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

```
mysql> select name from employee E,department D where E.dept=D.dept and (salary<minsal or salary>maxsal);
+-----+
| name   |
+-----+
| kritika |
| moksh   |
+-----+
2 rows in set (0.00 sec)
```

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

```
mysql> select name,Dname from employee E,department D where E.no=D.hod;
+-----+-----+
| name   | Dname  |
+-----+-----+
| mukul  | sales  |
| nupur   | finance |
| shetty  | admin  |
+-----+-----+
3 rows in set (0.00 sec)
```