

20MCA131 PROGRAMMING LAB

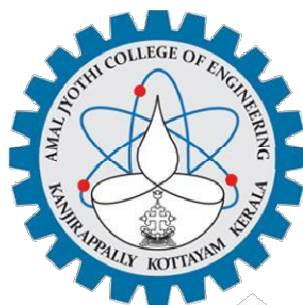
Lab Report Submitted By

ATHIRA BIJU
Reg. No.: AJC21MCA-2044

In Partial fulfillment for the Award of the Degree Of

MASTER OF COMPUTER APPLICATIONS (2 Year)
(MCA)

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY



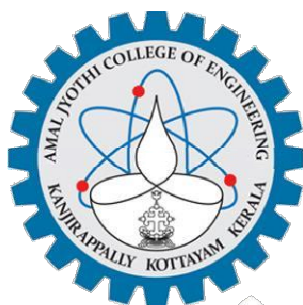
AMAL JYOTHI COLLEGE OF ENGINEERING
KANJIRAPPALLY

[Affiliated to APJ Abdul Kalam Technological University, Kerala. Approved by AICTE, Accredited by NAAC with 'A' grade. Koovapally, Kanjirappally, Kottayam, Kerala – 686518]

2021-2022

DEPARTMENT OF COMPUTER APPLICATIONS

AMAL JYOTHI COLLEGE OF ENGINEERING KANJIRAPPALLY



CERTIFICATE

This is to certify that the lab report, “**20MCA131 PROGRAMMING LAB**” is the bonafide work of **ATHIRA BIJU (Reg. No:AJC21MCA-2044)** in partial fulfillment of the requirements for the award of the Degree of Master of Computer Applications under APJ Abdul Kalam Technological University during the year 2021-22.

Sruthimol Kurian

Staff In-Charge

CONTENT

Sl.No	Content	Date	Page No:
1	Display future leap years from current year to a final year entered by user.	25-10-2021	6
2	List comprehensions	01-11-2021	7-8
3	Count the occurrences of each word in a line of text.	08-11-2021	9
4	Prompt the user for a list of integers. For all values greater than 100, store 'over' instead.	08-11-2021	10
5	Store a list of first names. Count the occurrences of 'a' within the list	08-11-2021	11
6	Lists of integers	11-11-2021	12-13
7	Get a string from an input string where all occurrences of first character replaced with '\$', except first character.	11-11-2021	14
8	Create a string from given string where first and last characters exchanged.	18-11-2021	15
9	Accept the radius from user and find area of circle.	22-11-2021	16
10	Find biggest of 3 numbers entered.	22-11-2021	17
11	Accept a file name from user and print extension of that.	25-11-2021	18
12	Create a list of colors from comma-separated color names entered by user. Display first and last colors.	25-11-2021	19
13	Accept an integer n and compute n+nn+nnn.	29-11-2021	20
14	Print out all colors from color-list1 not contained in color-list2.	06-12-2021	21
15	Create a single string separated with space from two strings by swapping the character at position 1.	1-12-2021	22
16	Sort dictionary in ascending and descending order.	09-12-2021	23
17	Merge two dictionaries.	13-12-2021	24
18	Find GCD of 2 numbers.	13-12-2021	25

19	From a list of integers, create a list removing even numbers.	16-12-2021	26
20	Program to find the factorial of a number	16-12-2021	27
21	Generate Fibonacci series of N terms	20-12-2021	28
22	Find the sum of all items in a list.	20-12-2021	29
23	Generate a list of four digit numbers in a given range with all their digits even and the number is a perfect square.	03-01-2022	30
24	Display the given pyramid with step number accepted from user.	06-01-2022	31
25	Count the number of characters (character frequency) in a string.	10-01-2022	32
26	Add 'ing' at the end of a given string. If it already ends with 'ing', then add 'ly'	10-01-2022	33
27	Accept a list of words and return length of longest word.	13-01-2022	34
28	Construct pattern using nested loop	13-01-2022	35
29	Generate all factors of a number.	17-01-2022	36
30	Write lambda functions to find area of square, rectangle and triangle.	17-01-2022	37
31	Create a package graphics with modules rectangle, circle and sub-package 3D-graphics with modules cuboid and sphere.	20-01-2022	38-40
32	Create Rectangle class with attributes length and breadth and methods to find area and perimeter. Compare two Rectangle objects by their area.	20-01-2022	41-42
33	Create a Bank account with members account number, name, type of account and balance. Write constructor and methods to deposit at the bank and withdraw an amount from the bank.	24-01-2022	43-44
34	Create a class Rectangle with private attributes length and width. Overload '<' operator to compare the area of 2 rectangles.	24-01-2022	45-46
35	Create a class Time with private attributes hour, minute and second. Overload '+' operator to find sum of 2 time.	27-01-2022	47-48
36	Create a class for Book Publisher and performance inheritance.	27-01-2022	49-50
37	Write a Python program to read a file line by line and store it into a list.	27-01-2022	51
38	Program to copy odd lines of one file to other	31-01-2022	52
39	Write a Python program to read each row from a given csv file and print a list of strings.	31-01-2022	53

40	Write a Python program to read specific columns of a given CSV file and print the content of the columns.	03-02-2022	54-55
41	Write a Python program to write a Python dictionary to a csv file. After writing the CSV file read the CSV file and display the content.	03-02-2022	56-57

Program no: 1**Aim:**

Display future leap years from current year to a final year entered by user.

Source Code:

```
print ("Print leap year between two given years")
print ("Enter start year")
startYear = int(input())
print ("Enter last year")
endYear = int(input())
print ("List of leap years:")
for year in range(startYear, endYear):
    if (0 == year % 4) and (0 != year % 100) or (0 == year % 400):
        print (year)
```

Output:

```
"C:\Program Files\Python39\python.exe" C:/Users/LENOVO/Desktop/pythonproject/login.py
Print leap year between two given years
Enter start year
2012
Enter last year
2020
List of leap years:
2012
2016

Process finished with exit code 0
```

Program no: 2**Aim:**

List comprehensions:

- a. Generate positive list of numbers from a given list of integers

Source Code:

```
list = [1, 2, 3, -5, -8, -6]
for x in list:
    if x > 0:
        print(x)
```

Output:

```
"C:\Program Files\Python39\python.exe" C:/Users/LENOVO/Desktop/pythonproject/login.py
1
2
3
Process finished with exit code 0
```

Aim:

- b. Square of N numbers

Source Code:

```
numbers = [1, 2, 3, 4, 5]
squared_numbers = [number ** 2 for number in numbers]
print(squared_numbers)
```

Output:

```
"C:\Program Files\Python39\python.exe" C:/Users/LENOVO/Desktop/pythonproject/login.py
[1, 4, 9, 16, 25]
Process finished with exit code 0
```

Aim:

c. : Form a list of vowels selected from a given word

Source Code:

```
words = ['apple', 'orange', 'pear', 'milk', 'otter']
for word in words:
    if word[0] in "aeiou":
        print(word)
```

Output:

```
"C:\Program Files\Python39\python.exe" C:/Users/LENOVO/Desktop/pythonproject/login.py
apple
orange
otter

Process finished with exit code 0
```

Aim:

d. List ordinal value of each element of a word (Hint: use ord() to get ordinal values)

Source Code:

```
x=["Athira"]
value=[ord(x) for x in x for x in x]
print(value)
```

Output:

```
"C:\Program Files\Python39\python.exe" C:/Users/LENOVO/Desktop/pythonproject/login.py
[65, 116, 104, 105, 114, 97]


Process finished with exit code 0
```


Program no: 3**Aim:**

Count the occurrences of each word in a line of text.

Source Code:

```
names=['athira','anjali','athira','ravi']  
nm=input('Enter name to count:')  
count=names.count(nm)  
print('count=',count)
```

Output:

```
"C:\Program Files\Python39\python.exe" C:/Users/LENOVO/Desktop/pythonproject/login.py  
[65, 116, 104, 105, 114, 97]  
  
Process finished with exit code 0
```

Program no: 4**Aim:**

Prompt the user for a list of integers. For all values greater than 100, store 'over' instead.

Source Code:

```
lst=[]  
l=int(input("enter the limit="))  
for i in range(l):  
    a=int(input("enter the value="))  
    lst.append('over' if a>100 else a)  
print(lst)
```

Output:

```
"C:\Program Files\Python39\python.exe" C:/Users/LENOVO/Desktop/pythonproject/login.py  
enter the limit=5  
enter the value=10  
enter the value=20  
enter the value=101  
enter the value=120  
enter the value=122  
[10, 20, 'over', 'over', 'over']  
  
Process finished with exit code 0  
|
```

Program no: 5**Aim:**

Store a list of first names. Count the occurrences of 'a' within the list

Source Code:

```
list1=['dency','akshara','shalvin','mridhula','nigi']
c=0
for i in list1: if
    "a" in i:
        c=c+1
print(c)
```

Output:

```
"C:\Users\NEHA-ANTONY\PycharmProjects\avodha_pythonprograms\venv\Scripts\python.exe" "C:/Users/NEHA-ANTONY/PycharmProjects/avodha_pythonclasses/PROGRAMMING LAB/labcycle1/22.py"
3
Process finished with exit code 0
```

Program no: 6**Aim:**

Enter 2 lists of integers.

- a. Whether list are of same length

Source Code:

```
a = [10, 10, 11, 12, 12, 13, 14, 14, 2, 3, 45]
b = [16, 12, 13, 14, 15, 16, 10, 11, 12, 10, 12]
print(len(a))
print(len(b))
if (len(a) > len(b) or len(a) < len(b)):
    print("lists are not of same length")
else:
    print("lists are of same length")
```

Output

```
"C:\Program Files\Python39\python.exe" C:/Users/LENOVO/Desktop/pythonproject/login.py
11
11
lists are of same length

Process finished with exit code 0
```

Aim: .

- b. whether list sums to same value

Source Code:

```
list1 = [1, 2, 3, 4, 5, 1, 4, 5]
Sum1 = sum(list1)
print(Sum1)
list2=[2,6,2,4,1]
Sum2=sum(list2)
print(Sum2)
if((Sum1<Sum2) or (Sum1>Sum2)):
    print("sums are not equal")
else:
    print("sums are equal")
```

Output:

```
"C:\Program Files\Python39\python.exe" C:/Users/LENOVO/Desktop/pythonproject/login.py
25
15
sums are not equal

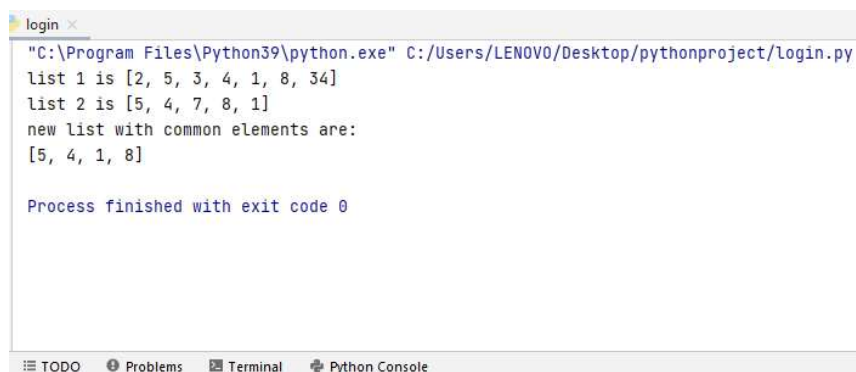
Process finished with exit code 0
```

Aim: .

c: whether any value occur in both

Source Code:

```
list1 = [2, 5, 3, 4, 1, 8, 34]
print("list 1 is", list1)
list2 = [5, 4, 7, 8, 1]
print("list 2 is", list2)
print("new list with common elements are:")
list = []
for i in list1:
    if i in list2:
        list.append(i)
print(list)
```

Output:

```
login x
"C:\Program Files\Python39\python.exe" C:/Users/LENOVO/Desktop/pythonproject/login.py
list 1 is [2, 5, 3, 4, 1, 8, 34]
list 2 is [5, 4, 7, 8, 1]
new list with common elements are:
[5, 4, 1, 8]

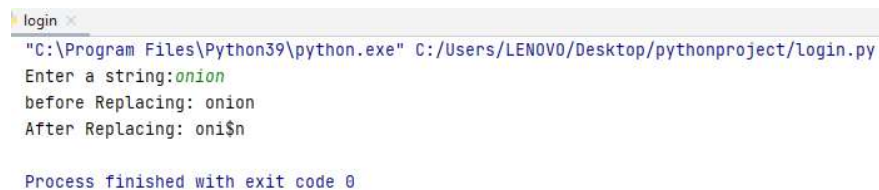
Process finished with exit code 0
```

Program no: 7**Aim:**

Get a string from an input string where all occurrences of first character replaced with '\$', except first character. [eg: onion -> oni\$n]

Source Code:

```
str=input("Enter a string:")
print("before Replacing:",str)
a=str[0]
rep=str.replace(a,'$')
string=a+rep[1:]
print("After Replacing:",string)
```

Output:

A screenshot of a Windows command prompt window titled 'login'. The command prompt shows the execution of a Python script 'login.py' located at 'C:/Users/LENOVO/Desktop/pythonproject/'. The script prompts the user to 'Enter a string:' and the user enters 'onion'. The script then prints 'before Replacing: onion' and 'After Replacing: oni\$n'. The command prompt also shows the exit code '0'.

```
login
"C:\Program Files\Python39\python.exe" C:/Users/LENOVO/Desktop/pythonproject/login.py
Enter a string:onion
before Replacing: onion
After Replacing: oni$n

Process finished with exit code 0
```

Program no: 8**Aim:**

Create a string from given string where first and last characters exchanged. [eg:
python>nythop]

Source Code:

```
text = input("Enter a string:")  
  
newtext = text[-1]+text[1:-1]+text[0]  
  
print("New string:",newtext)
```

OUTPUT

The screenshot shows a Python IDE window titled 'login'. The command prompt displays the execution of a Python script: `"C:\Program Files\Python39\python.exe" C:/Users/LEN0V0/Desktop/pythonproject/login.py`. The user is prompted to 'Enter a string:' and enters 'python'. The output is 'New string: nythop'. The process finished with exit code 0. The IDE interface includes a sidebar with 'TODO', 'Problems', 'Terminal', and 'Python Console' tabs.

```
login x  
"C:\Program Files\Python39\python.exe" C:/Users/LEN0V0/Desktop/pythonproject/login.py  
Enter a string:python  
New string: nythop  
  
Process finished with exit code 0  
|  
  
TODO Problems Terminal Python Console
```

Program no: 9

Aim: Accept the radius from user and find area of circle.

Source Code:

```
radius=int(input("Enter the radius of circle:"))  
  
area=3.14*(radius*radius)  
  
print("area of circle is:",area)
```

Output:

A screenshot of a Windows command prompt window titled 'login'. The window shows the execution of a Python script. The command prompt displays the full path to the Python interpreter and the script file: "C:\Program Files\Python39\python.exe" C:/Users/LENOVO/Desktop/pythonproject/login.py. The user input '15' is shown in green, followed by the output 'area of circle is: 706.5' in green. The window concludes with 'Process finished with exit code 0' in blue.

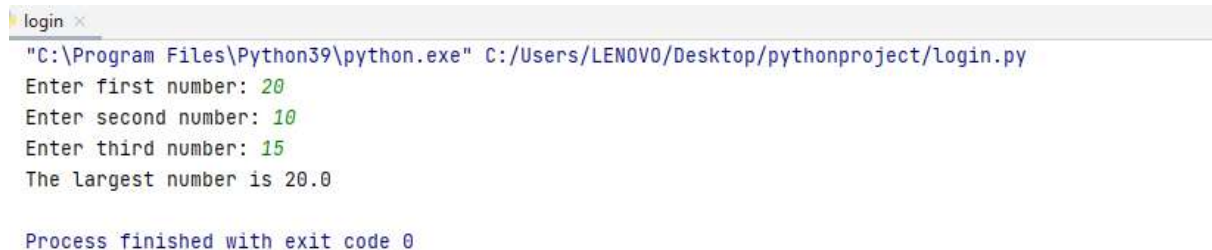
```
login  
"C:\Program Files\Python39\python.exe" C:/Users/LENOVO/Desktop/pythonproject/login.py  
Enter the radius of circle:15  
area of circle is: 706.5  
  
Process finished with exit code 0
```

Program no: 10**Aim:**

Find biggest of 3 numbers entered.

Source Code:

```
num1 = float(input("Enter first number: "))
num2 = float(input("Enter second number: "))
num3 = float(input("Enter third number: "))
if (num1 >= num2) and (num1 >= num3):
    largest = num1
elif (num2 >= num1) and (num2 >= num3):
    largest = num2
else:
    largest = num3
print("The largest number is", largest)
```

Output:

```
login x
"C:\Program Files\Python39\python.exe" C:/Users/LENOVO/Desktop/pythonproject/login.py
Enter first number: 20
Enter second number: 10
Enter third number: 15
The largest number is 20.0

Process finished with exit code 0
```

Program no:11**Aim:**

Accept a file name from user and print extension of that.

Source Code:

```
n=input("Enter a filename with extension:")
```

```
x=n.split(".")
```

```
print("Extension of file name is:",x[-1])
```

Output:


```
login x
"C:\Program Files\Python39\python.exe" C:/Users/LENOVO/Desktop/pythonproject/login.py
Enter a filename with extension:image.jpeg
Extension of file name is: jpeg
Process finished with exit code 0
```

Program no: 12**Aim:**

Create a list of colors from comma-separated color names entered by user. Display first and last colors.

Source Code:

```
list1=[]  
  
list1=input("enter the colors with seperated commas:")  
  
list2=list1.split(",")  
  
print(list2)  
  
print("First color:",list2[0])  
  
print("Last color:",list2[-1])
```

Output:

```
"C:\Program Files\Python39\python.exe" C:/Users/LENOVO/Desktop/pythonproject/login.py  
enter the colors with seperated commas:yellow,green,red  
['yellow', 'green', 'red']  
First color: yellow  
Last color: red  
  
Process finished with exit code 0  
|
```

Program no: 13**Aim:**

Accept an integer n and compute $n+nn+nnn$.

Source Code:

```
n=int(input("enter the value for n"))
temp=str(n)
t1=temp+temp
t2=temp+temp+temp
sum=n+int(t1)+int(t2)
print("the value is",sum)
```

Output:A screenshot of a Windows command prompt window titled 'login'. The window shows the execution of a Python script. The command prompt displays the file path and filename: '"C:\Program Files\Python39\python.exe" C:/Users/LENOVO/Desktop/pythonproject/login.py'. Below this, the program's input and output are shown: 'enter the value for n 4' followed by 'the value is 492'. At the bottom, it states 'Process finished with exit code 0'.

```
login X
"C:\Program Files\Python39\python.exe" C:/Users/LENOVO/Desktop/pythonproject/login.py
enter the value for n 4
the value is 492

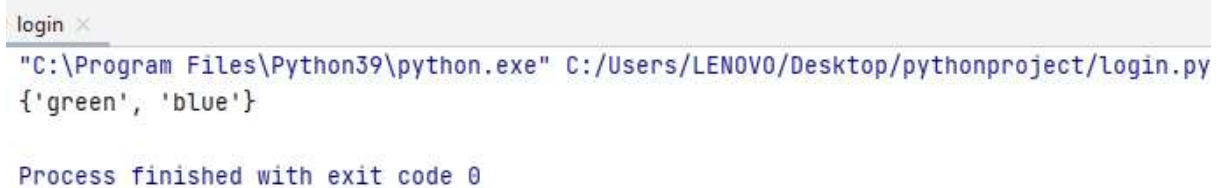
Process finished with exit code 0
```

Program no: 14**Aim:**

Print out all colors from color-list1 not contained in color-list2.

Source Code:

```
colorlist1 = set(["red","blue","green"])  
colorlist2 = set(["red","yellow","pink",])  
a=colorlist1.difference(colorlist2)  
  
print(a)
```

Output:

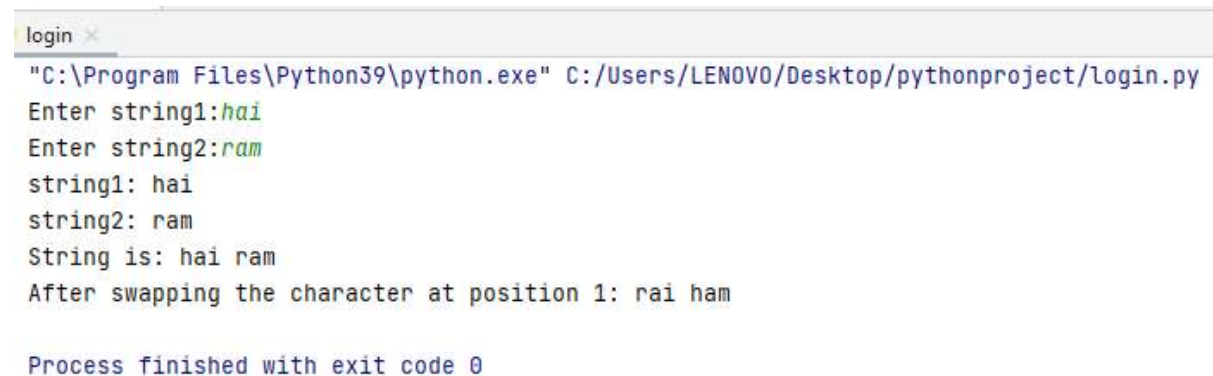
```
login ×  
"C:\Program Files\Python39\python.exe" C:/Users/LENOVO/Desktop/pythonproject/login.py  
{'green', 'blue'}  
  
Process finished with exit code 0
```

Program no: 15**Aim:**

Create a single string separated with space from two strings by swapping the character at position 1.

Source Code:

```
str1=input("Enter string1:")
str2=input("Enter string2:")
print("string1:",str1)
print("string2:",str2)
s=str1+' '+str2
print("String is:",s)
a=str1[0]
m=str1[1:]
b=str2[0]
n=str2[1:]
print("After swapping the character at position 1:",b+m+' '+a+n)
```

Output:

```
login x
"C:\Program Files\Python39\python.exe" C:/Users/LENOVO/Desktop/pythonproject/login.py
Enter string1:hai
Enter string2:ram
string1: hai
string2: ram
String is: hai ram
After swapping the character at position 1: rai ham

Process finished with exit code 0
```

Program no: 16**Aim:**

Sort dictionary in ascending and descending order.

Source Code:

```
s={'manu':42,'sam':32,'arya':12,'david':3}
y=list(s.items())
y.sort()
temp=dict(y)
print("Ascending order is",temp)
y.sort(reverse=True)
dict=dict(y)
print("Descending order is",temp)
```

Output:

```
"C:\Program Files\Python39\python.exe" C:/Users/LENOVO/Desktop/pythonproject/login.py
Ascending order is {'arya': 12, 'david': 3, 'manu': 42, 'sam': 32}
Descending order is {'arya': 12, 'david': 3, 'manu': 42, 'sam': 32}

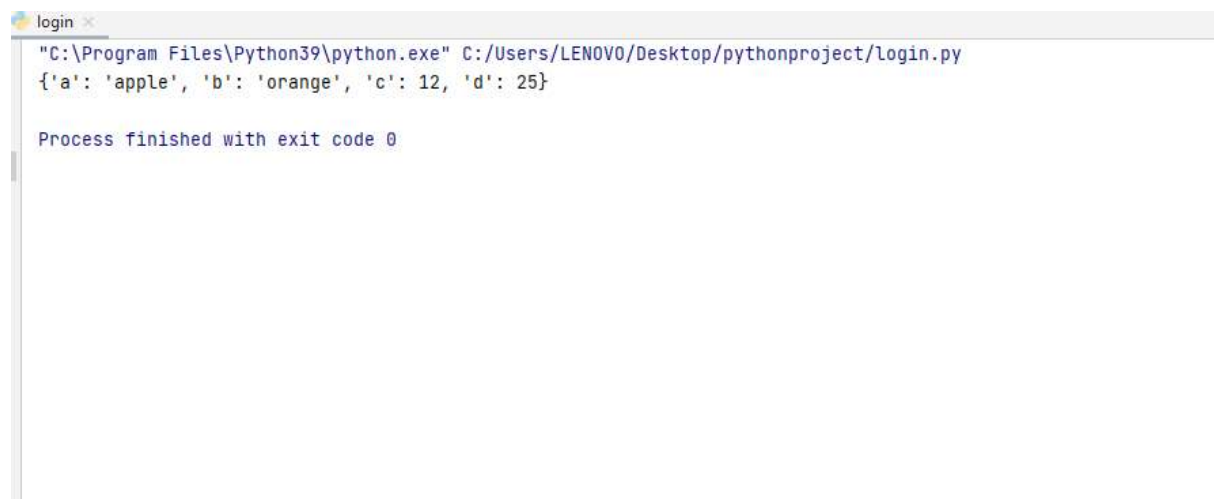
Process finished with exit code 0
```

Program no: 17**Aim:**

Merge two dictionaries.

Source Code:

```
d1 = {'a':'apple','b':'apple'}  
d2 = {'c':12,'d':25}  
d1.update(d2)  
print(d1)
```

Output:

```
login  
"C:\Program Files\Python39\python.exe" C:/Users/LENOVO/Desktop/pythonproject/login.py  
{'a': 'apple', 'b': 'orange', 'c': 12, 'd': 25}  
  
Process finished with exit code 0
```

Program no: 18**Aim:**

Find gcd of 2 numbers.

Source Code:

```
import math  
  
a = int(input("Enter the 1 st number:"))  
b = int(input("Enter the 2 nd number:"))  
print("gcd of",a,"and",b,"is:",math.gcd(a,b))
```

Output:

```
login x  
"C:\Program Files\Python39\python.exe" C:/Users/LENOVO/Desktop/pythonproject/login.py  
Enter the 1 st number:20  
Enter the 2 nd number:35  
gcd of 20 and 35 is: 5  
  
Process finished with exit code 0
```

Program no: 19**Aim:**

From a list of integers, create a list removing even numbers.

Source Code:

```
n=int(input("Enter the limit:"))

list1=[]

print("Enter the values:")

for i in range(0,n):

    m=int(input())

    list1.append(m)

print("list is:",list1)

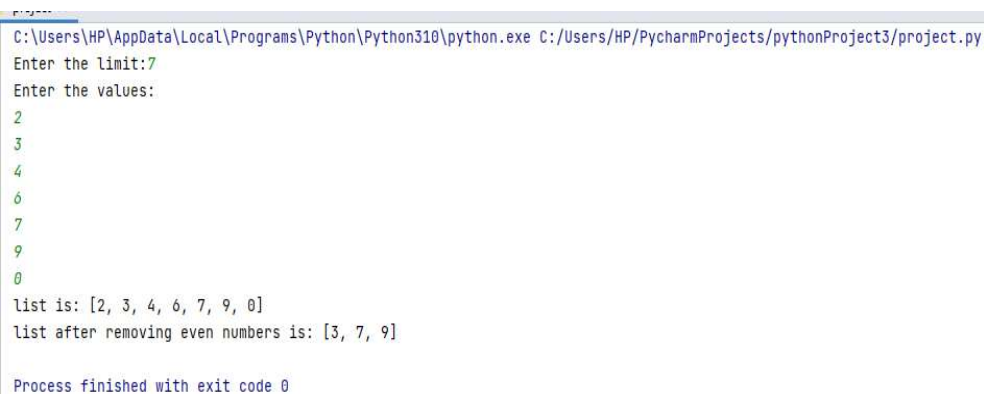
list2=[]

for i in list1:

    if i%2!=0:

        list2.append(i)

print("list after removing even numbers is:",list2)
```

Output:

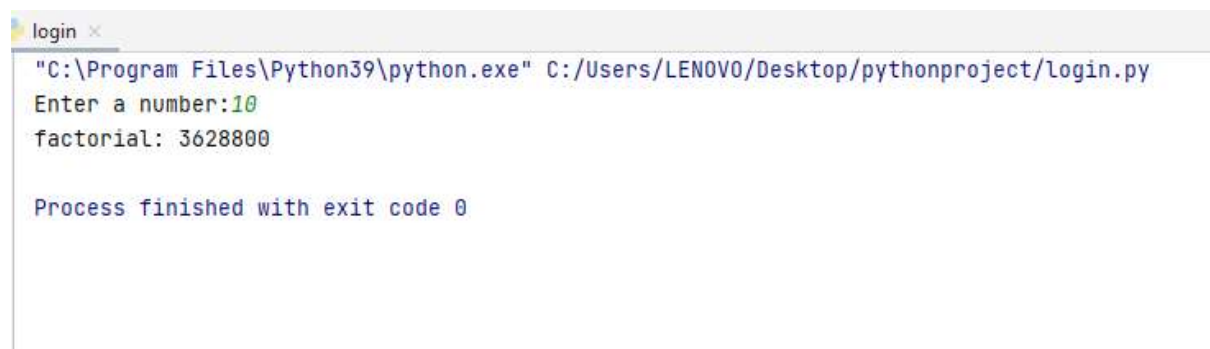
```
C:\Users\HP\AppData\Local\Programs\Python\Python310\python.exe C:/Users/HP/PycharmProjects/pythonProject3/project.py
Enter the limit:7
Enter the values:
2
3
4
6
7
9
0
list is: [2, 3, 4, 6, 7, 9, 0]
list after removing even numbers is: [3, 7, 9]
Process finished with exit code 0
```

Program no: 20**Aim:**

Program to find the factorial of a number

Source Code:

```
num=int(input("Enter a number:"))
def factorial(n):
    fact = 1
    for i in range(1,n+1):
        fact = fact * i
    print("factorial:",fact)
factorial(num)
```

Output:A screenshot of a Windows command prompt window titled 'login'. The command executed is '"C:\Program Files\Python39\python.exe" C:/Users/LENOVO/Desktop/pythonproject/login.py'. The output shows the program prompting 'Enter a number:' where the user has entered '10'. The program then outputs 'factorial: 3628800'. At the bottom, it states 'Process finished with exit code 0'.

```
login x
"C:\Program Files\Python39\python.exe" C:/Users/LENOVO/Desktop/pythonproject/login.py
Enter a number:10
factorial: 3628800

Process finished with exit code 0
```

Program no: 21**Aim:**

Generate Fibonacci series of N terms

Source Code:

```
n=int(input("Enter the limit:"))

n1=0

n2=1

s=0

print("fibonacci series:")

print(n1)

print(n2)

for i in range(1,n-1):

    s=n1+n2

    n1=n2

    n2=s

    print(s)
```

Output:

The screenshot shows a terminal window titled 'login' with the following content:

```
"C:\Program Files\Python39\python.exe" C:/Users/LENOVO/Desktop/pythonproject/login.py
Enter the limit:6
fibonacci series:
0
1
1
2
3
5

Process finished with exit code 0
```

Program no: 22**Aim:**

Find the sum of all items in a list.

Source Code:

```
def sum_list(items):  
    sum_numbers = 0  
    for x in items:  
        sum_numbers += x  
    return sum_numbers  
print(sum_list([1,2,-8]))
```

Output:

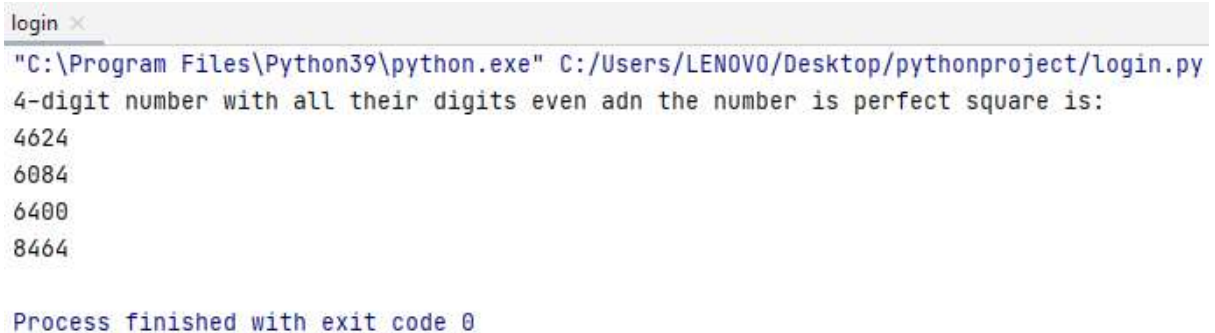
```
login ×  
"C:\Program Files\Python39\python.exe" C:/Users/LENOVO/Desktop/pythonproject/login.py  
-5  
  
Process finished with exit code 0
```

Program no: 23**Aim:**

Generate a list of four digit numbers in a given range with all their digits even and the number is a perfect square.

Source Code:

```
print("4-digit number with all their digits even adn the number is perfect square is:")  
for i in range(1000, 10000, 1):  
    for j in range(32, 100, 1):  
        if i == j * j:  
            string = str(i)  
            if int(string[0]) % 2 == 0 and int(string[1]) % 2 == 0 and int(string[2]) % 2 == 0 and  
int(string[3]) % 2 == 0:  
                print(i)
```

Output:

```
login x  
"C:\Program Files\Python39\python.exe" C:/Users/LENOVO/Desktop/pythonproject/login.py  
4-digit number with all their digits even adn the number is perfect square is:  
4624  
6084  
6400  
8464  
  
Process finished with exit code 0
```

Program no: 24**Aim:**

Display the given pyramid with step number accepted from user.

Eg: N=4

```
1
2 4
3 6 9
4 8 12 16
```

Source Code:

```
row=int(input("enter the no of rows"))
for i in range(1,row+1):
    for j in range(1,i+1):
        print(i*j, end=' ')
    print()
```

Output:

```
project x
C:\Users\HP\AppData\Local\Programs\Python\Python310\python.exe C:/Users/HP/PycharmProjects/pythonProject3/project.py
1
2 4
3 6 9
4 8 12 16

Process finished with exit code 0
```

Program no:25

Aim: Count the number of characters (character frequency) in a string.

Source Code:

```
a = input("Enter the string:")

count = 0

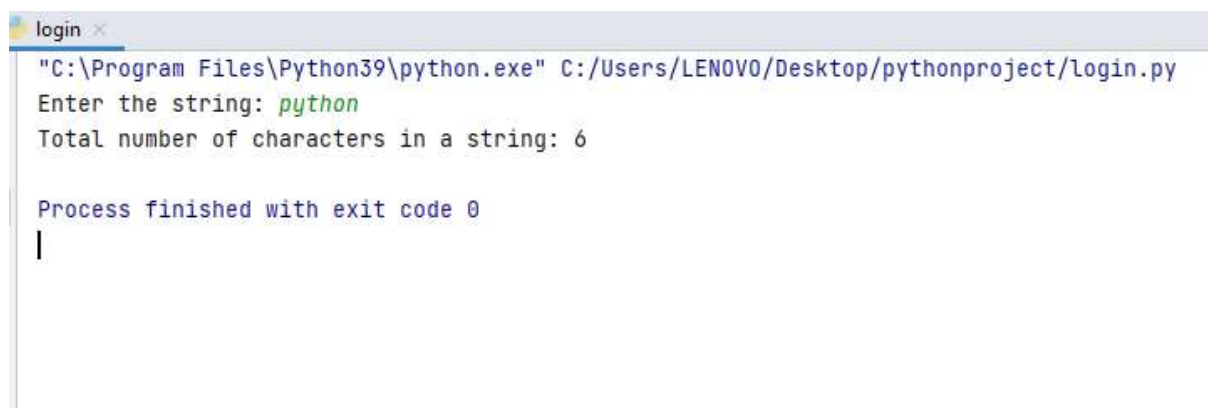
for i in range(0, len(a)):

    if (a[i] != ' '):

        count = count + 1

print("Total number of characters in a string: " + str(count))
```

Output:



```
login x
"C:\Program Files\Python39\python.exe" C:/Users/LENOVO/Desktop/pythonproject/login.py
Enter the string: python
Total number of characters in a string: 6

Process finished with exit code 0
|
```

Program no:26

Aim: Add 'ing' at the end of a given string. If it already ends with 'ing', then add 'ly'

Source Code:

```
str=input("Enter a string:")
```

```
def fun(str1):
```

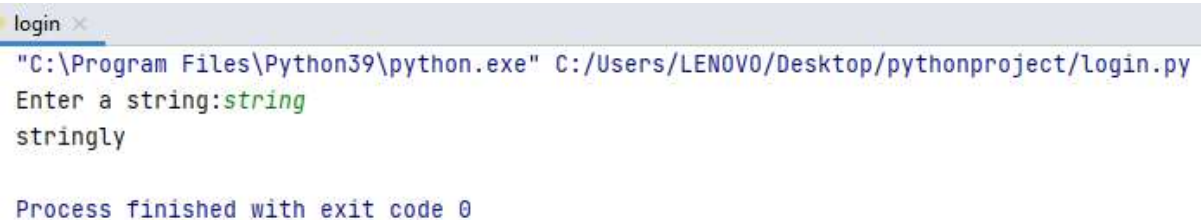
```
    if str[-3:]!='ing':
```

```
        print(str+"ing")
```

```
    else:
```

```
        print(str + "ly")
```

```
fun(str)
```

Output:

```
login x
"C:\Program Files\Python39\python.exe" C:/Users/LENOVO/Desktop/pythonproject/login.py
Enter a string:string
stringly

Process finished with exit code 0
```

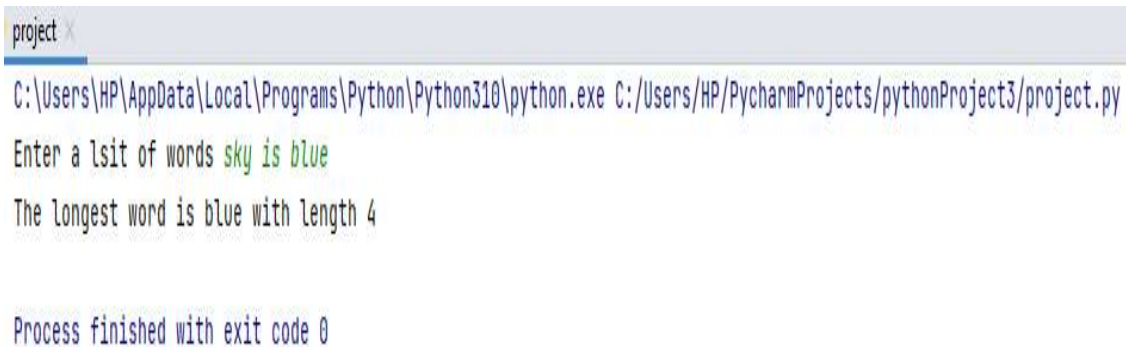
Program no:27

Aim: Accept a list of words and return length of longest word.

Source Code:

```
def word():  
    text = input("Enter a list of words ")  
    longest = 0  
    for words in text.split(' '):  
        if len(words) > longest:  
            longest = len(words)  
    longest_word = words  
    print("The longest word is", longest_word, "with length", len(longest_word))  
word()
```

Output:



```
project x  
C:\Users\HP\AppData\Local\Programs\Python\Python310\python.exe C:/Users/HP/PycharmProjects/pythonProject3/project.py  
Enter a list of words sky is blue  
The longest word is blue with length 4  
  
Process finished with exit code 0
```

Program no:28

Aim: Construct following pattern using nested loop

```
  *
 * *
* * *
* * * *
  * * *
    * *
      *
```

Source Code:

```
n=4;
for i in range(n):
    for j in range(i):
        print('*', end='')
    print("")
for i in range(n,0,-1):
    for j in range(i):
        print('*', end='')
    print("")
```

Output:



```
login
"C:\Program Files\Python39\python.exe" C:/Users/LENOVO/Desktop/pythonproject/login.py

*
* *
* * *
* * * *
  * * *
    * *
      *

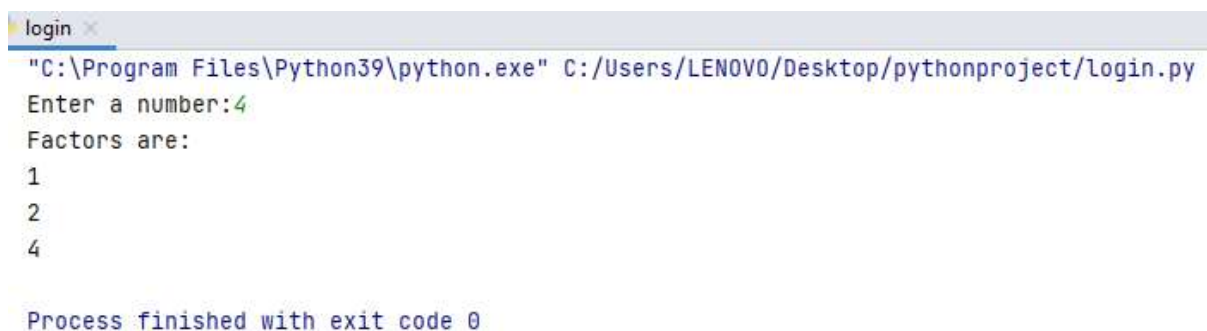
Process finished with exit code 0
```

Program no:29**Aim:**

Generate all factors of a number.

Source Code:

```
n=int(input("Enter a number:"))  
def factor(num):  
    for i in range(1,n+1):  
        if n%i==0:  
            print(i)  
print("Factors are:")  
factor(n)
```

Output:

```
login x  
"C:\Program Files\Python39\python.exe" C:/Users/LENOVO/Desktop/pythonproject/login.py  
Enter a number:4  
Factors are:  
1  
2  
4  
  
Process finished with exit code 0
```

Program no:30

Aim: Write lambda functions to find area of square, rectangle and triangle.

Source Code:

```
t_area = lambda b, h: 0.5 * (b * h)

r_area = lambda len, ht: len * ht

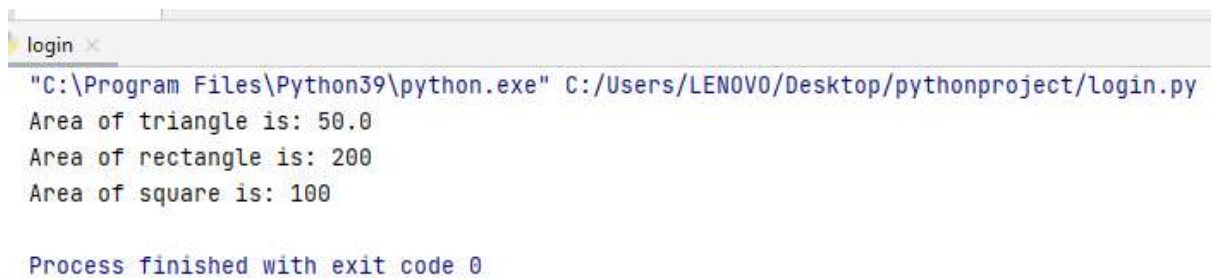
s_area = lambda leng: leng * leng

print("Area of triangle is:", t_area(10, 20))

print("Area of rectangle is:", r_area(30, 20))

print("Area of square is:", s_area(15))
```

Output:



```
login ×
"C:\Program Files\Python39\python.exe" C:/Users/LENOVO/Desktop/pythonproject/login.py
Area of triangle is: 50.0
Area of rectangle is: 200
Area of square is: 100

Process finished with exit code 0
```

Program no:31**Aim:**

Create a package graphics with modules rectangle, circle and sub-package 3D-graphics with modules cuboid and sphere. Include methods to find area and perimeter of respective figures in each module. Write programs that finds area and perimeter of figures by different importing statements. (Include selective import of modules and import * statements).

Source Code:**graphpack.py**

```
from graphics import circle,rectangle

from graphics.dgraphics import cuboid,sphere

r=int(input("Enter the radius of circle:"))

circle.areac(r)

circle.peric(r)

l=int(input("Enter the length of rectangle:"))

b=int(input("Enter the breadth of rectangle:"))

rectangle.areas(l,b)

rectangle.perir(l,b)

l1=int(input("Enter the length of cuboid:"))

b1=int(input("Enter the breadth of cuboid:"))

h1=int(input("Enter the height of cuboid:"))

cuboid.areacub(l1,b1,h1)

cuboid.pericub(l1,b1,h1)

r1=int(input("Enter the radius of sphere:"))

sphere.areas(r1)

sphere.peris(r1)
```

graphics**circle.py**

```
def areac(r):

    a=3.14*r*r

print("Area of Circle is:",a)
```

```
def peric(r):
```

```
    p=2*3.14*r
```

```
print("Perimeter of Circle is:",p)
```

rectangle.py

```
def arear(l,b):
```

```
    a=l*b
```

```
print("Area of Rectangle is:",a)
```

```
def perir(l,b):
```

```
    p=2*(l+b)
```

```
print("Area of Rectangle is:",p)
```

dgraphics

sphere.py

```
def areas(r):
```

```
    a = 4*3.14*r*r
```

```
print("Area of Sphere is:", a)
```

```
def peris(r):
```

```
    p = 6.2832*r
```

```
print("Perimeter of Sphere is:", p)
```

cuboid.py

```
def areacub(l,b,h):
```

```
    a = 2*((l*b) + (b*h) + (h*l))
```

```
print("Area of Cuboid is:", a)
```

```
def pericub(l,b,h):
```

```
    p = 4*(l+b+h)
```

```
print("Perimeter of Cuboid is:", p)
```

Output:

```
"C:\Program Files\Python39\python.exe" "C:/Users/Hp/Desktop/python programs/New folder/07-01-2022/graphpack.py"
Enter the radius of circle:2
Area of Circle is: 12.56
Perimeter of Circle is: 12.56
Enter the length of rectangle:3
Enter the breadth of rectangle:2
Area of Rectangle is: 6
Area of Rectangle is: 10
Enter the length of cuboid:4
Enter the breadth of cuboid:3
Enter the height of cuboid:5
Area of Cuboid is: 94
Perimeter of Cuboid is: 48
Enter the radius of sphere:3
Area of Sphere is: 113.03999999999999
Perimeter of Sphere is: 18.8496

Process finished with exit code 0
|
```

Program no:32**Aim:**

Create Rectangle class with attributes length and breadth and methods to find area and perimeter. Compare two Rectangle objects by their area.

Source Code:

```
class rect:

    def __init__(self,l,b):

self.a1=l

self.a2=b

    def area(self):

self.m=self.a1*self.a2

    def peri(self):

self.n=2*(self.a1 + self.a2)

    def disp(self):

print("Area of rectangle:", self.m)

print("Perimeter of rectangle:", self.n)

    def compare(self,obj2):

    if self.m == obj2.m:

print("Areas are equal")

elifself.m> obj2.m:

print("Area1 is greater than Area2")

    else:

print("Area2 is greater than Area1")

l1=int(input("Enter length1:"))

b1=int(input("Enter breadth1:"))

l2=int(input("Enter length2:"))

b2=int(input("Enter breadth2:"))

obj1=rect(l1,b1)

obj2=rect(l2,b2)
```

obj1.area()

obj1.peri()

obj2.area()

obj2.peri()

obj1.disp()

obj2.disp()

obj1.compare(obj2)

Output:



```
project x
C:\Users\HP\AppData\Local\Programs\Python\Python310\python.exe C:/Users/HP/PycharmProjects/pythonProject3/project.py
Enter length1:10
Enter breadth1:7
Enter length2:12
Enter breadth2:8
Area of rectangle: 70
Perimeter of rectangle: 34
Area of rectangle: 96
Perimeter of rectangle: 40
Area2 is greater than Area1
Process finished with exit code 0
```

Program no:33**Aim:**

Create a Bank account with members account number, name, type of account and balance.
Write constructor and methods to deposit at the bank and withdraw an amount from the bank.

Source Code:

```
class bank:

    def __init__(self,a,n,t,b):

        self.ac = a

        self.name = n

    self.type = t

    self.bal = b

    def depo(self,a1):

self.bal += a1

        print("Balance:",self.bal)

    def widthdraw(self,a2):

        if self.bal<a2:

            print("Invalid")

        else:

self.bal -= a2

            print("Balance:",self.bal)

    def disp(self):

print("Acc No:",self.ac)

print("Name:", self.name)

print("Acc Type:", self.type)

print("Acc Balance:", self.bal)

a=int(input("Enter acc no:"))

n=input("Enter name:")

t=input("Enter acc type:")

b=int(input("Enter balance:"))
```

```
obj1=bank(a,n,t,b)

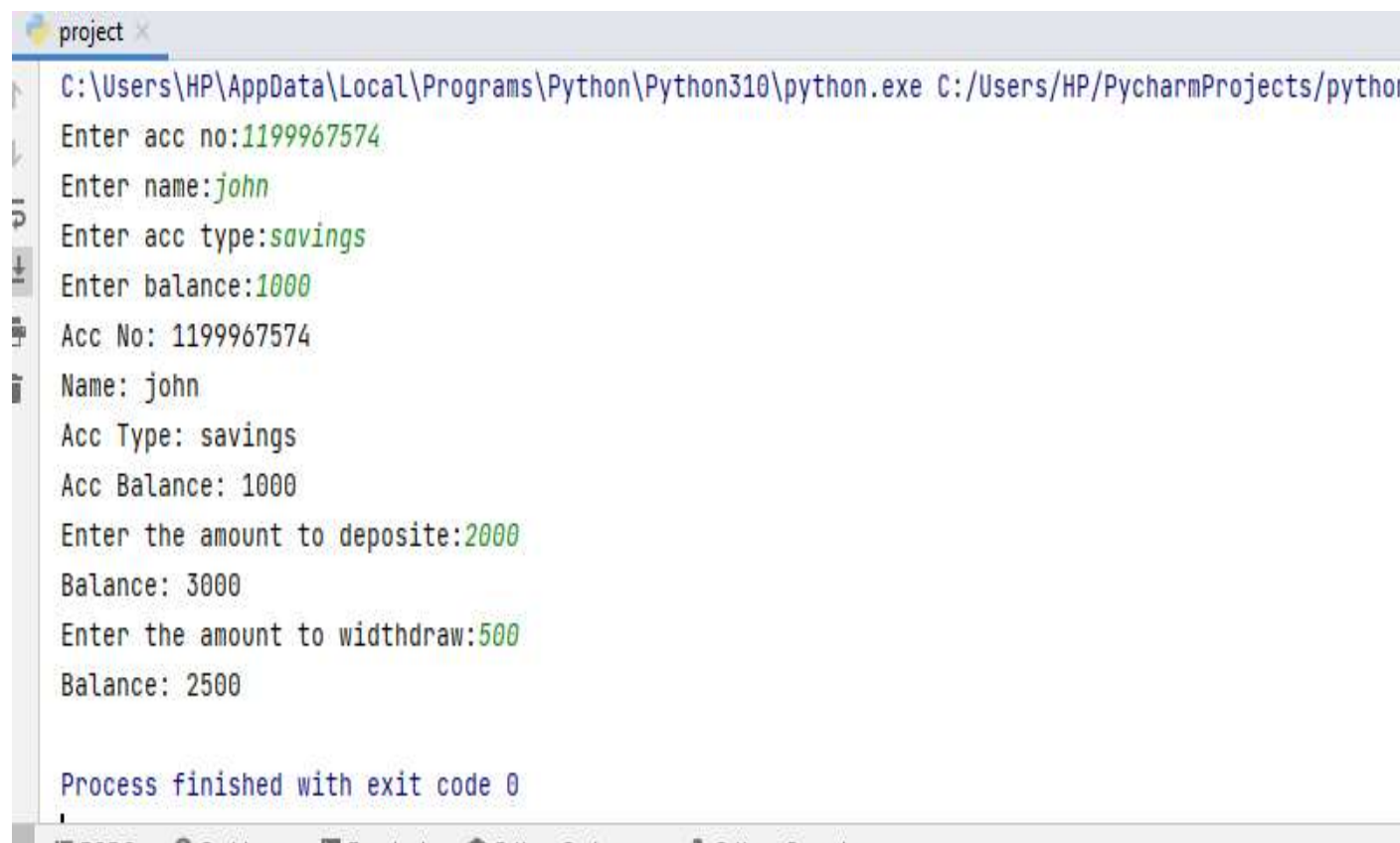
obj1.disp()

a1=int(input("Enter the amount to deposit:"))

obj1.depo(a1)

a2=int(input("Enter the amount to widthdraw:"))

obj1.widthdraw(a2)
```

Output:

```
project x
C:\Users\HP\AppData\Local\Programs\Python\Python310\python.exe C:/Users/HP/PycharmProjects/python
Enter acc no:1199967574
Enter name:john
Enter acc type:savings
Enter balance:1000
Acc No: 1199967574
Name: john
Acc Type: savings
Acc Balance: 1000
Enter the amount to deposit:2000
Balance: 3000
Enter the amount to widthdraw:500
Balance: 2500

Process finished with exit code 0
```

Program no:34**Aim:**

Create a class Rectangle with private attributes length and width. Overload '<' operator to compare the area of 2 rectangles.

Source Code:

```
class bank:

def __init__(self,a,n,t,b):

    self.ac = a

    self.name = n

    self.type = t

    self.bal = b

def depo(self,a1):

    self.bal += a1

    print("Balance:",self.bal)

def widthdraw(self,a2):

    print("Invalid")

else: self.bal -= a2

    print("Balance:",self.bal)

def disp(self):

    print("Acc No:",self.ac)

    print("Name:", self.name)

    print("Acc Type:", self.type)

    print("Acc Balance:", self.bal)

a=int(input("Enter acc no:"))

n=input("Enter name:")

t=input("Enter acc type:")

b=int(input("Enter balance:"))

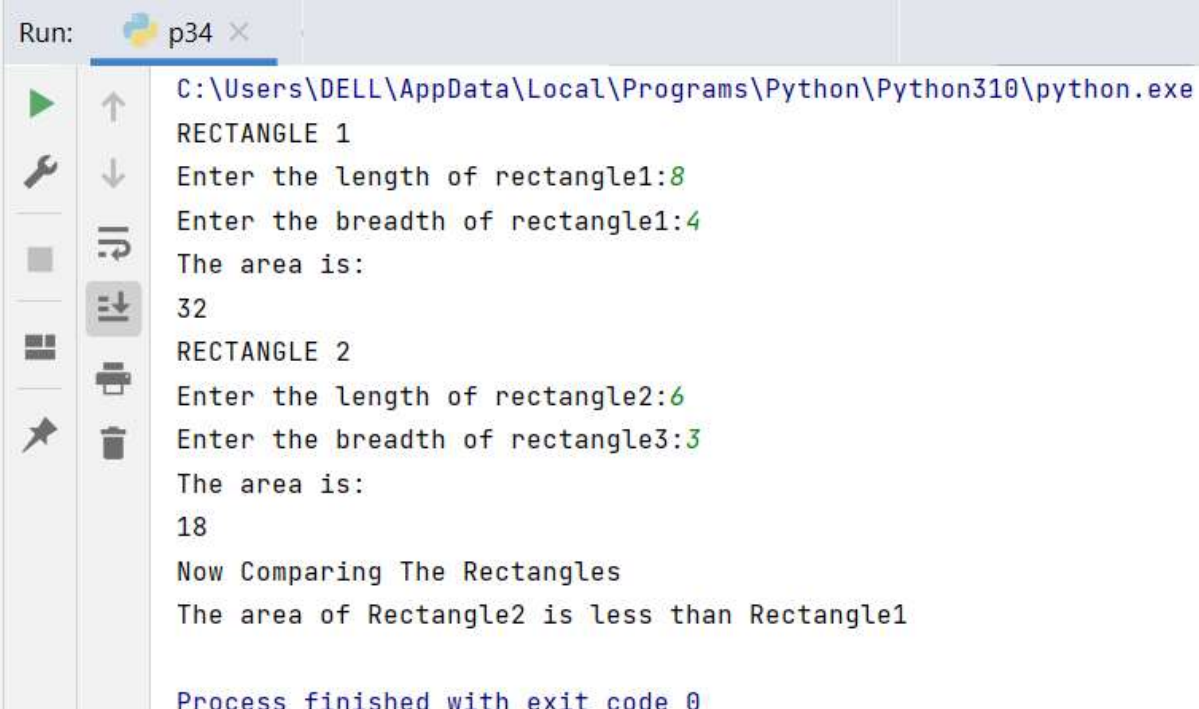
obj1=bank(a,n,t,b) obj1.disp()

a1=int(input("Enter the amount to deposite:"))
```

```
obj1.depo(a1)
```

```
a2=int(input("Enter the amount to widthdraw:"))
```

```
obj1.widthdraw(a2)
```

Output:

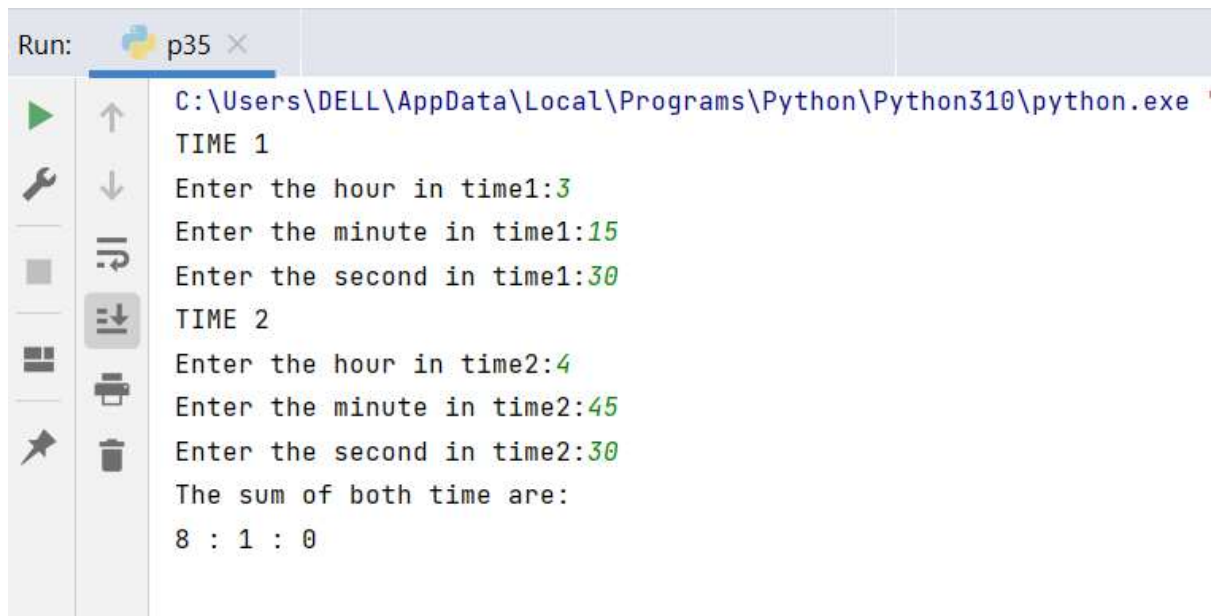
```
Run: p34 x
C:\Users\DELL\AppData\Local\Programs\Python\Python310\python.exe
RECTANGLE 1
Enter the length of rectangle1:8
Enter the breadth of rectangle1:4
The area is:
32
RECTANGLE 2
Enter the length of rectangle2:6
Enter the breadth of rectangle3:3
The area is:
18
Now Comparing The Rectangles
The area of Rectangle2 is less than Rectangle1
Process finished with exit code 0
```

Program no:35**Aim:**

Create a class Time with private attributes hour, minute and second. Overload '+' operator to find sum of 2 time.

Source Code:**class Time:**

```
class Time:
    def __init__(self, h, m, s):
        self._h1 = h
        self._m1 = m
        self._s1 = s
    def __add__(self, x):
        sum1 = self._h1 + x._h1
        sum2 = self._m1 + x._m1
        sum3 = self._s1 + x._s1
        if sum3 >= 60:
            sum3 = sum3 - 60
            sum2 = sum2 + 1
        if sum2 >= 60:
            sum2 = sum2 - 60
            sum1 = sum1 + 1
        print(sum1, ":", sum2, ":", sum3);
    print("TIME 1")
h1 = int(input("Enter the hour in time1:"))
m1 = int(input("Enter the minute in time1:"))
s1 = int(input("Enter the second in time1:"))
obj1 = Time(h1, m1, s1)
print("TIME 2")
h2 = int(input("Enter the hour in time2:"))
m2 = int(input("Enter the minute in time2:"))
s2 = int(input("Enter the second in time2:"))
obj2 = Time(h2, m2, s2)
print("The sum of both time are:")
obj1 + obj2
```

Output:

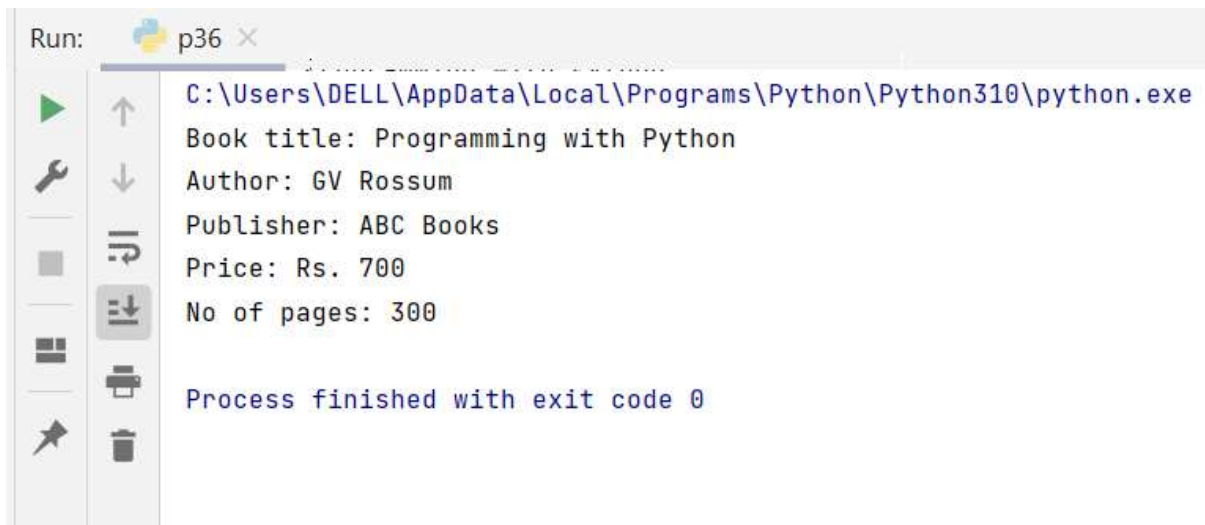
```
Run: p35 x
C:\Users\DELL\AppData\Local\Programs\Python\Python310\python.exe
TIME 1
Enter the hour in time1:3
Enter the minute in time1:15
Enter the second in time1:30
TIME 2
Enter the hour in time2:4
Enter the minute in time2:45
Enter the second in time2:30
The sum of both time are:
8 : 1 : 0
```

Program no:36**Aim:**

Create a class Publisher (name). Derive class Book from Publisher with attributes title and author. Derive class Python from Book with attributes price and no_of_pages. Write a program that displays information about a Python book. Use base class constructor invocation and method overriding.

Source Code:

```
class Publisher:
    def __init__(self,name1):
        self.name=name1
    def show(self):
        pass
class Book(Publisher):
    def __init__(self,title1,author1,name1):
        self.title=title1
        self.author=author1
        Publisher.__init__(self,name1)
    def show(self):
        pass
class Python(Book):
    def __init__(self,p,no,title1,author1,name1):
        self.price=p
        self.no_of_pages=no
        Book.__init__(self,title1,author1,name1)
    def show(self):
        print('Book title:',self.title)
        print('Author:',self.author)
        print('Publisher:',self.name)
        print('Price: Rs.',self.price)
        print('No of pages:',self.no_of_pages)
P1=Python(700,300,'Programming with Python','GV Rossum','ABC Books')
P1.show()
```

Output:

The screenshot shows a terminal window titled 'Run: p36'. The output text is as follows:

```
C:\Users\DELL\AppData\Local\Programs\Python\Python310\python.exe
Book title: Programming with Python
Author: GV Rossum
Publisher: ABC Books
Price: Rs. 700
No of pages: 300

Process finished with exit code 0
```

The interface includes a vertical toolbar on the left with icons for running, stepping through code, and other debugging functions.

Program no:37**Aim:**

Write a Python program to read a file line by line and store it into a list.

Source Code:**demo.txt**

Python

Interpreted high-level language.

Python is object oriented programming language

line.py

```
def fread(fname):
```

```
    with open(fname) as f:
```

```
        c = f.readlines()
```

```
    print(c)
```

```
fread("demo")
```

Output:

```
"C:\Program Files\Python39\python.exe" "C:/Users/Hp/Desktop/python programs/New folder/28-01-22/File/1.py"
['Python\n', 'Interpreted high-level language.\n', 'Python is object oriented programming language']

Process finished with exit code 0
```

Program no:38

Aim: Python program to copy odd lines of one file to other

Source Code:

demo.txt

Python

Interpreted high-level language.

Python is object oriented programming language

line.py

```
a = open("demo", "r")
```

```
b = open("t", "w")
```

```
c = a.readlines()
```

```
d = len(c)
```

```
for i in range(0, d):
```

```
    if i % 2 == 0:
```

```
        b.write(c[i])
```

```
    else:
```

```
        pass
```

```
b.close()
```

```
b = open("t", "r")
```

```
e = b.read()
```

```
print(e)
```

```
a.close()
```

```
b.close()
```

Output:

```
"C:\Program Files\Python39\python.exe" "C:/Users/Hp/Desktop/python programs/New folder/28-01-22/File/1.py"
['Python\n', 'Interpreted high-level language.\n', 'Python is object oriented programming language']

Process finished with exit code 0
```

Program no:39

Aim: Write a Python program to read each row from a given csv file and print a list of strings.

Source Code:

CSV

Series_reference	Period	Data_value	Suppressed	STATUS	UNITS	Magnitude	Subject	Group	Series_title_1	Series_title_2	Series_title_3
BDCQ.SF1AA2CA	2016.06	1116.386	,F	Dollars	6	Business Data Collection	- BDC,Industry by financial variable,Sales (operating income),F				
BDCQ.SF1AA2CA	2016.09	1070.874	,F	Dollars	6	Business Data Collection	- BDC,Industry by financial variable,Sales (operating income),F				
BDCQ.SF1AA2CA	2016.12	1054.408	,F	Dollars	6	Business Data Collection	- BDC,Industry by financial variable,Sales (operating income),F				
BDCQ.SF1AA2CA	2017.03	1010.665	,F	Dollars	6	Business Data Collection	- BDC,Industry by financial variable,Sales (operating income),F				
BDCQ.SF1AA2CA	2017.06	1233.7	,F	Dollars	6	Business Data Collection	- BDC,Industry by financial variable,Sales (operating income),F				
BDCQ.SF1AA2CA	2017.09	1282.436	,F	Dollars	6	Business Data Collection	- BDC,Industry by financial variable,Sales (operating income),F				
BDCQ.SF1AA2CA	2017.12	1290.82	,F	Dollars	6	Business Data Collection	- BDC,Industry by financial variable,Sales (operating income),F				
BDCQ.SF1AA2CA	2018.03	1412.007	,F	Dollars	6	Business Data Collection	- BDC,Industry by financial variable,Sales (operating income),F				
BDCQ.SF1AA2CA	2018.06	1488.055	,F	Dollars	6	Business Data Collection	- BDC,Industry by financial variable,Sales (operating income),F				
BDCQ.SF1AA2CA	2018.09	1497.678	,F	Dollars	6	Business Data Collection	- BDC,Industry by financial variable,Sales (operating income),F				
BDCQ.SF1AA2CA	2018.12	1570.507	,F	Dollars	6	Business Data Collection	- BDC,Industry by financial variable,Sales (operating income),F				
BDCQ.SF1AA2CA	2019.03	1393.749	,F	Dollars	6	Business Data Collection	- BDC,Industry by financial variable,Sales (operating income),F				
BDCQ.SF1AA2CA	2019.06	1517.143	,F	Dollars	6	Business Data Collection	- BDC,Industry by financial variable,Sales (operating income),F				
BDCQ.SF1AA2CA	2019.09	1381.514	,F	Dollars	6	Business Data Collection	- BDC,Industry by financial variable,Sales (operating income),F				
BDCQ.SF1AA2CA	2019.12	1370.985	,F	Dollars	6	Business Data Collection	- BDC,Industry by financial variable,Sales (operating income),F				
BDCQ.SF1AA2CA	2020.03	1073.017	,F	Dollars	6	Business Data Collection	- BDC,Industry by financial variable,Sales (operating income),F				
BDCQ.SF1AA2CA	2020.06	1131.445	,F	Dollars	6	Business Data Collection	- BDC,Industry by financial variable,Sales (operating income),F				
BDCQ.SF1AA2CA	2020.09	1440.101	,F	Dollars	6	Business Data Collection	- BDC,Industry by financial variable,Sales (operating income),F				
BDCQ.SF1AA3CA	2016.06	1189.735	,F	Dollars	6	Business Data Collection	- BDC,Industry by financial variable,Sales (operating income),F				
BDCQ.SF1AA3CA	2016.09	1144.938	,F	Dollars	6	Business Data Collection	- BDC,Industry by financial variable,Sales (operating income),F				

line.py

```
import csv
```

with `open("csv", newline=") as csvfile:`

```
d = csv.reader(csvfile, delimiter=' ', quotechar='"')
```

```
for i in d:
```

```
print(', '.join(i))
```

Output:

```
"C:\Program Files\Python39\python.exe" "C:/Users/Hp/Desktop/python programs/New folder/28-01-22/File3.py"
Series_reference,Period,Data,value,Suppressed,STATUS,UNITS,Magnitude,Subject,Group,Series,title_1,Series,title_2,Series,title_3,Series,title_4,Series,title_5
BQCQ.SF1A2C8,2016.86,1116.366,F,Dollars,&Business,Data,Collection,-,BQC,Industry,by,financial,variable,Sales,(operating,income),Forestry,and,Logging,Current,prices,U
BQCQ.SF1A2C8,2016.89,1070.874,F,Dollars,&Business,Data,Collection,-,BQC,Industry,by,financial,variable,Sales,(operating,income),Forestry,and,Logging,Current,prices,U
BQCQ.SF1A2C8,2016.12,1054.408,F,Dollars,&Business,Data,Collection,-,BQC,Industry,by,financial,variable,Sales,(operating,income),Forestry,and,Logging,Current,prices,U
BQCQ.SF1A2C8,2017.83,1020.645,F,Dollars,&Business,Data,Collection,-,BQC,Industry,by,financial,variable,Sales,(operating,income),Forestry,and,Logging,Current,prices,U
BQCQ.SF1A2C8,2017.86,1233.7,F,Dollars,&Business,Data,Collection,-,BQC,Industry,by,financial,variable,Sales,(operating,income),Forestry,and,Logging,Current,prices,U
BQCQ.SF1A2C8,2017.89,1282.436,F,Dollars,&Business,Data,Collection,-,BQC,Industry,by,financial,variable,Sales,(operating,income),Forestry,and,Logging,Current,prices,U
BQCQ.SF1A2C8,2017.12,1292.82,F,Dollars,&Business,Data,Collection,-,BQC,Industry,by,financial,variable,Sales,(operating,income),Forestry,and,Logging,Current,prices,U
BQCQ.SF1A2C8,2018.83,1422.007,F,Dollars,&Business,Data,Collection,-,BQC,Industry,by,financial,variable,Sales,(operating,income),Forestry,and,Logging,Current,prices,U
BQCQ.SF1A2C8,2018.86,1488.955,F,Dollars,&Business,Data,Collection,-,BQC,Industry,by,financial,variable,Sales,(operating,income),Forestry,and,Logging,Current,prices,U
BQCQ.SF1A2C8,2018.89,1497.678,F,Dollars,&Business,Data,Collection,-,BQC,Industry,by,financial,variable,Sales,(operating,income),Forestry,and,Logging,Current,prices,U
BQCQ.SF1A2C8,2018.12,1570.587,F,Dollars,&Business,Data,Collection,-,BQC,Industry,by,financial,variable,Sales,(operating,income),Forestry,and,Logging,Current,prices,U
BQCQ.SF1A2C8,2019.83,1393.749,F,Dollars,&Business,Data,Collection,-,BQC,Industry,by,financial,variable,Sales,(operating,income),Forestry,and,Logging,Current,prices,U
BQCQ.SF1A2C8,2019.86,1517.143,F,Dollars,&Business,Data,Collection,-,BQC,Industry,by,financial,variable,Sales,(operating,income),Forestry,and,Logging,Current,prices,U
BQCQ.SF1A2C8,2019.89,1381.54,F,Dollars,&Business,Data,Collection,-,BQC,Industry,by,financial,variable,Sales,(operating,income),Forestry,and,Logging,Current,prices,U
BQCQ.SF1A2C8,2019.12,1370.985,F,Dollars,&Business,Data,Collection,-,BQC,Industry,by,financial,variable,Sales,(operating,income),Forestry,and,Logging,Current,prices,U
```

Program no:40

Aim: Write a Python program to read specific columns of a given CSV file and print the content of the columns.

Source Code:

csv

Series_reference	Period	Data_value	Suppressed	STATUS	UNITS	Magnitude	Subject	Group	Series_title_1	Series_title_2	Series
BDCQ.SF1AA2CA	2016.06	1116.386	,F	Dollars	6	Business Data Collection - BDC	Industry by financial variable	Sales (operating income)			
BDCQ.SF1AA2CA	2016.09	1070.874	,F	Dollars	6	Business Data Collection - BDC	Industry by financial variable	Sales (operating income)			
BDCQ.SF1AA2CA	2016.12	1054.408	,F	Dollars	6	Business Data Collection - BDC	Industry by financial variable	Sales (operating income)			
BDCQ.SF1AA2CA	2017.03	1010.665	,F	Dollars	6	Business Data Collection - BDC	Industry by financial variable	Sales (operating income)			
BDCQ.SF1AA2CA	2017.06	1233.7	,F	Dollars	6	Business Data Collection - BDC	Industry by financial variable	Sales (operating income)			
BDCQ.SF1AA2CA	2017.09	1282.436	,F	Dollars	6	Business Data Collection - BDC	Industry by financial variable	Sales (operating income)			
BDCQ.SF1AA2CA	2017.12	1290.82	,F	Dollars	6	Business Data Collection - BDC	Industry by financial variable	Sales (operating income)			
BDCQ.SF1AA2CA	2018.03	1412.007	,F	Dollars	6	Business Data Collection - BDC	Industry by financial variable	Sales (operating income)			
BDCQ.SF1AA2CA	2018.06	1488.055	,F	Dollars	6	Business Data Collection - BDC	Industry by financial variable	Sales (operating income)			
BDCQ.SF1AA2CA	2018.09	1497.678	,F	Dollars	6	Business Data Collection - BDC	Industry by financial variable	Sales (operating income)			
BDCQ.SF1AA2CA	2018.12	1570.507	,F	Dollars	6	Business Data Collection - BDC	Industry by financial variable	Sales (operating income)			
BDCQ.SF1AA2CA	2019.03	1393.749	,F	Dollars	6	Business Data Collection - BDC	Industry by financial variable	Sales (operating income)			
BDCQ.SF1AA2CA	2019.06	1517.143	,F	Dollars	6	Business Data Collection - BDC	Industry by financial variable	Sales (operating income)			
BDCQ.SF1AA2CA	2019.09	1381.514	,F	Dollars	6	Business Data Collection - BDC	Industry by financial variable	Sales (operating income)			
BDCQ.SF1AA2CA	2019.12	1370.985	,F	Dollars	6	Business Data Collection - BDC	Industry by financial variable	Sales (operating income)			
BDCQ.SF1AA2CA	2020.03	1073.017	,F	Dollars	6	Business Data Collection - BDC	Industry by financial variable	Sales (operating income)			
BDCQ.SF1AA2CA	2020.06	1131.445	,F	Dollars	6	Business Data Collection - BDC	Industry by financial variable	Sales (operating income)			
BDCQ.SF1AA2CA	2020.09	1440.101	,F	Dollars	6	Business Data Collection - BDC	Industry by financial variable	Sales (operating income)			
BDCQ.SF1AA3CA	2016.06	1189.735	,F	Dollars	6	Business Data Collection - BDC	Industry by financial variable	Sales (operating income)			
BDCQ.SF1AA3CA	2016.09	1144.938	,F	Dollars	6	Business Data Collection - BDC	Industry by financial variable	Sales (operating income)			

line.py

```
import csv
```

```
with open("csv", newline="") as csvfile:
```

```
    d = csv.DictReader(csvfile)
```

```
    print("Period    Subject")
```

```
    print("-----")
```

```
    for i in d:
```

```
        print(i['Period'], i['Subject'])
```

Output:

```
"C:\Program Files\Python39\python.exe" "C:/Users/Hp/Desktop/python programs/New folder/28-01-22/File/4.py"
```

```
Period      Subject
```

```
-----
```

```
2016.06 Business Data Collection - BDC
```

```
2016.09 Business Data Collection - BDC
```

```
2016.12 Business Data Collection - BDC
```

```
2017.03 Business Data Collection - BDC
```

```
2017.06 Business Data Collection - BDC
```

```
2017.09 Business Data Collection - BDC
```

```
2017.12 Business Data Collection - BDC
```

```
2018.03 Business Data Collection - BDC
```

```
2018.06 Business Data Collection - BDC
```

Program no:41

Aim: Write a Python program to write a Python dictionary to a csv file. After writing the CSV file read the CSV file and display the content.

Source Code:

csv

Series_reference	Period	Data_value	Suppressed	STATUS	UNITS	Magnitude	Subject	Group	Series_title_1	Series_title_2	Series_title_3
BDCQ.SF1AA2CA	2016.06	1116.386	,F	Dollars	6	Business Data Collection - BDC	Industry by financial variable	Sales (operating income)			
BDCQ.SF1AA2CA	2016.09	1070.874	,F	Dollars	6	Business Data Collection - BDC	Industry by financial variable	Sales (operating income)			
BDCQ.SF1AA2CA	2016.12	1054.408	,F	Dollars	6	Business Data Collection - BDC	Industry by financial variable	Sales (operating income)			
BDCQ.SF1AA2CA	2017.03	1010.665	,F	Dollars	6	Business Data Collection - BDC	Industry by financial variable	Sales (operating income)			
BDCQ.SF1AA2CA	2017.06	1233.7	,F	Dollars	6	Business Data Collection - BDC	Industry by financial variable	Sales (operating income)			
BDCQ.SF1AA2CA	2017.09	1282.436	,F	Dollars	6	Business Data Collection - BDC	Industry by financial variable	Sales (operating income)			
BDCQ.SF1AA2CA	2017.12	1290.82	,F	Dollars	6	Business Data Collection - BDC	Industry by financial variable	Sales (operating income)			
BDCQ.SF1AA2CA	2018.03	1412.807	,F	Dollars	6	Business Data Collection - BDC	Industry by financial variable	Sales (operating income)			
BDCQ.SF1AA2CA	2018.06	1488.855	,F	Dollars	6	Business Data Collection - BDC	Industry by financial variable	Sales (operating income)			
BDCQ.SF1AA2CA	2018.09	1497.678	,F	Dollars	6	Business Data Collection - BDC	Industry by financial variable	Sales (operating income)			
BDCQ.SF1AA2CA	2018.12	1570.507	,F	Dollars	6	Business Data Collection - BDC	Industry by financial variable	Sales (operating income)			
BDCQ.SF1AA2CA	2019.03	1393.749	,F	Dollars	6	Business Data Collection - BDC	Industry by financial variable	Sales (operating income)			
BDCQ.SF1AA2CA	2019.06	1517.143	,F	Dollars	6	Business Data Collection - BDC	Industry by financial variable	Sales (operating income)			
BDCQ.SF1AA2CA	2019.09	1381.514	,F	Dollars	6	Business Data Collection - BDC	Industry by financial variable	Sales (operating income)			
BDCQ.SF1AA2CA	2019.12	1370.985	,F	Dollars	6	Business Data Collection - BDC	Industry by financial variable	Sales (operating income)			
BDCQ.SF1AA2CA	2020.03	1073.817	,F	Dollars	6	Business Data Collection - BDC	Industry by financial variable	Sales (operating income)			
BDCQ.SF1AA2CA	2020.06	1131.445	,F	Dollars	6	Business Data Collection - BDC	Industry by financial variable	Sales (operating income)			
BDCQ.SF1AA2CA	2020.09	1440.101	,F	Dollars	6	Business Data Collection - BDC	Industry by financial variable	Sales (operating income)			
BDCQ.SF1AA3CA	2016.06	1189.735	,F	Dollars	6	Business Data Collection - BDC	Industry by financial variable	Sales (operating income)			
BDCQ.SF1AA3CA	2016.09	1144.938	,F	Dollars	6	Business Data Collection - BDC	Industry by financial variable	Sales (operating income)			

line.py

```
import csv

field_name = ['No', 'Company', 'Car Model']

car = [{ 'No': 1, 'Company': 'Ferrari', 'Car Model': 'GH'},
        { 'No': 2, 'Company': 'BMW', 'Car Model': 'X5'},
        { 'No': 3, 'Company': 'Maruti Suzuki', 'Car Model': 'Swift'},
        { 'No': 4, 'Company': 'Audi', 'Car Model': 'RS7'},
        { 'No': 5, 'Company': 'Toyota', 'Car Model': 'Fortuner'}]

with open('b.csv', 'w') as csvfile:

    write = csv.DictWriter(csvfile, fieldnames=field_name)
```



```
write.writeheader()

write.writerows(car)

with open('b.csv', newline='') as csvfile:

    d = csv.reader(csvfile, delimiter=',')

    for r in d:

        print(', '.join(r))
```

Output:

```
"C:\Program Files\Python39\python.exe" "C:/Users/Hp/Desktop/python programs/New folder/28-01-22/File/5.py"
No,Company,Car Model

1,Ferrari,GH

2,BMW,X5

3,Maruti Suzuki,Swift

4,Audi,RS7

5,Toyota,Fortuner

Process finished with exit code 0
```