20MCA131 PROGRAMMING LAB

Lab Report Submitted By

SANDRA P M

Reg. No.: AJC21MCA-2092

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]

2021-2023

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 **SANDRA PM(Reg No.:AJC21MCA-2092)** 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 2021-23.

Ms Rini Kurian

Staff In-Charge

CONTENT

Sr. No	Content	Date	Pg. 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**	**7-9**
3.	Count the occurrences of each word in a line of text.	**01-11-2021**	**10**
4.	Prompt the user for a list of integers. For all values greater than 100, store ‘over’ instead.	**08-11-2021**	**11**
5.	Store a list of first names. Count the occurrences of ‘a’ within the list	**08-11-2021**	**12**
6.	Lists of integers	**11-11-2021**	**13-15**
7.	Get a string from an input string where all occurrences of first character replaced with ‘$’, except first character.	**11-11-2021**	**16**
8.	Create a string from given string where first and last characters exchanged.	**18-11-2021**	**17**
9.	Accept the radius from user and find area of circle.	**22-11-2021**	**18**
10.	Find biggest of 3 numbers entered.	**22-11-2021**	**19**
11.	Accept a file name from user and print extension of that.	**25-11-2021**	**20**
12.	Create a list of colors from comma-separated color names entered by user. Display first and last colors.	**25-11-2021**	**21**
13.	Accept an integer n and compute n+nn+nnn.	**29-11-2021**	**22**

14.	Print out all colors from color-list1 not contained in color-list2.	**06-12-2021**	**23**
15.	Create a single string separated with space from two strings by swapping the character at position 1.	**09-12-2021**	**24**
16.	Sort dictionary in ascending and descending order.	**09-12-2021**	**25**
17.	Merge two dictionaries.	**13-12-2021**	**26**
18.	Find GCD of 2 numbers.	**13-12-2021**	**27**
19.	From a list of integers, create a list removing even numbers.	**16-12-2021**	**28**
20.	Program to find the factorial of a number	**16-12-2021**	**29**
21.	Generate Fibonacci series of N terms	**20-12-2021**	**30**
22.	Find the sum of all items in a list.	**20-12-2021**	**31**
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**	**32**
24.	Display the given pyramid with step number accepted from user.	**06-01-2022**	**33**
25.	Count the number of characters (character frequency) in a string.	**06-01-2022**	**34**
26.	Add ‘ing’ at the end of a given string. If it already ends with ‘ing’, then add ‘ly’	**10-01-2022**	**35**
27.	Accept a list of words and return length of longest word.	**10-01-2022**	**36**
28.	Construct pattern using nested loop	**13-01-2022**	**37-38**
29.	Generate all factors of a number.	**13-01-2022**	**39**

30.	Write lambda functions to find area of square, rectangle and triangle.	**17-01-2022**	**40**
31.	Create a package graphics with modules rectangle, circle and sub-package 3D- graphics with modules cuboid and sphere.	**17-01-2022**	**41-42**
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**	**43-44**
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.	**20-01-2022**	**45-46**
34.	Create a class Rectangle with private attributes length and width. Overload '<' operator to compare the area of 2 rectangles.	**24-01-2022**	**47**
35.	Create a class Time with private attributes hour, minute and second. Overload '+' operator to find sum of 2 time.	**24-01-2022**	**48-49**
36.	Create a class for Book Publisher and performance inheritance.	**27-01-2022**	**50-51**
37.	Write a Python program to read a file line by line and store it into a list.	**27-01-2022**	**52**
38.	Program to copy odd lines of one file to other	**31-01-2022**	**53**
39.	Write a Python program to read each row from a given csv file and print a list of strings.	**31-01-2022**	**54-55**
40.	Write a Python program to read specific columns of a given CSV file and print the content of the columns.	**03-02-2022**	**56-57**
41.	Write a Python program to write a Python dictionary to a csv file. After writing the CSV file read the CSV fileand display the content.	**03-02-2022**	**58-59**

Program no:1

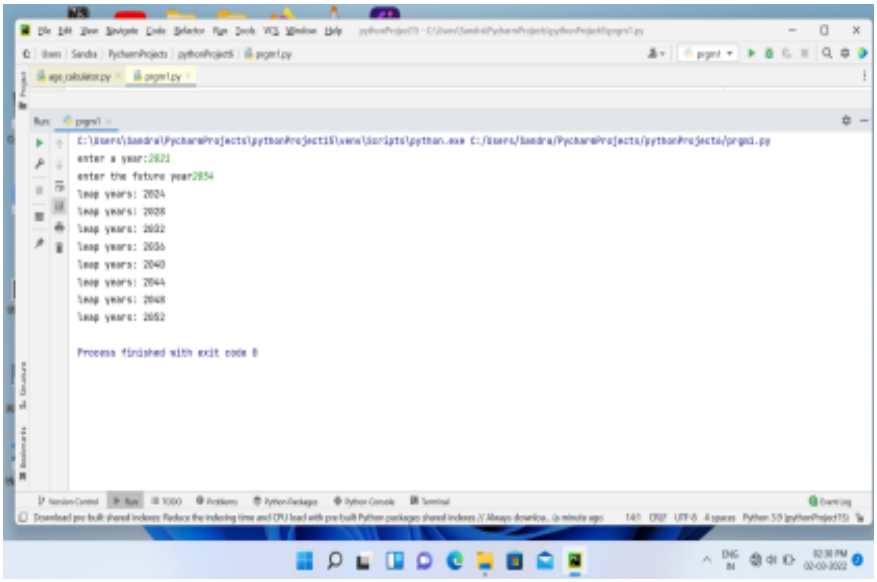
Aim:

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

Source Code:

```
a=int(input("enter the starting year:"))
b=int(input("enter the ending year:"))
for year in range(a, b):
    if year%400==0 or year%4==0 and year%100!=0:
        print("leap years are:", year)
```

Output:



Program no:2

Aim:

List comprehensions:

a. Generate positive list of numbers from a given list of integers
b. Square of N numbers
c. Form a list of vowels selected from a given word
d. List ordinal value of each element of a word

Source Code:

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

```
list1=[-1,0,3,4,-5,6,7,8]
print(“The given list are:”,list1)
 for pos in list1:
      if pos>0:
            print("positive numbers are:", pos)
```

b). Square of N numbers

```
numbers = [1, 2, 3, 4, 5]
print ("numbers in a list are:", numbers)
sq_ N = [number ** 2 for number in numbers]
 print("squared numbers:", sq_ N)
```

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

```
 a=input("enter the words:")
 b=input("enter word:")
 if b in a:
```

```
    for vowel in b:
        if vowel[0] in 'aeiou':
            print(list(vowel))
else:
    print("Element not found.")
```

d). List ordinal value of each element of a word

```
myinput=input("enter the message\t")
mylist =list(myinput)
mylist = [ord(character) + 1 for character in mylist]
print(mylist)
```

Output:

```
The given list are: [-1, 0, 3, 4, -5, 6, 7, 8]
positive numbers are: 3
positive numbers are: 4
positive numbers are: 6
positive numbers are: 7
positive numbers are: 8
```

```
numbers in a list are: [1, 2, 3, 4, 5]
squared numbers: [1, 4, 9, 16, 25]

Process finished with exit code 0
```

```
enter the words:Hai, how are you!
enter word:Hai
['a']
['i']

Process finished with exit code 0
```

```
enter the message   A program
[66, 33, 113, 115, 112, 104, 115, 98, 110]

Process finished with exit code 0
```

Program no:3

Aim:

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

Source Code:

```
a = input("enter a sentence")
b = input("enter the word")
print("occurrences of the word", a.count(b))
```

Output:

```
enter a sentence This is a simple program.This program name is count of occurences
 This is a simple program.This program name is count of occurences
enter the wordThis
occurences of the word 2

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:

```
list1=[]
n=int(input("enter the limit:"))
for i in range(n):
    a=int(input("enter the value:"))
    list1.append('over' if a>100 else a)
print(list1)
```

Output:

```
enter the limit:4
enter the value:100
enter the value:105
enter the value:80
enter the value:60
[100, 'over', 80, 60]
```

Program no:5

Aim:

Store a list of first names. Count the occurrences of ‘a’ within the list.

Source Code:

```
l=int(input("enter the num of first names\t"))
counts=0
for i in range(l):
    a=str(input("enter the name\t").split(" ")[0])
    counts+=a.count('a');
print(“repeated counts ”,counts)
```

Output:

```
enter the num of first names    3
enter the name  ponnu
enter the name  Rosna
enter the name  ann
repeated counts 2

Process finished with exit code 0
```

Program no:6

Aim:

Enter 2 lists of integers. Check:

a. Whether list are of same length
b. whether list sums to same value
c. whether any value occur in both

Source Code:

a).Whether list are of same length

```
list1=[1,2,3,4]

list2=[3,4,5,6]

print("list of 2 integers are:")

print("list1",list1)

print("list2",list2)

a=len(list1)

b=len(list2)

if a==b:

    print("length of 'list1' and 'list2' are same:",a,b)

else:

    print("length is not same in 'list1' and 'list2' ",a,b)
```

b).whether list sums to same value

```
import math

list1=[2,3,4,]

list2=[1,3,5]
```

```
print("list1",list1)

print("list2",list2)

if(sum(list1)==sum(list2)):

    print("sum of list1 and list2",sum(list1),sum(list2))

else:

    print("sum is not equal")
```

c). whether any value occur in both

```
list1=[2,3,4,]

list2=[1,3,5]

print("list1",list1)

print("list2",list2)

flag=1

for i in list1:

    for j in list2:

        if(i==j):

            print("value in both list1 and list2 is:",i)

            flag=1

            exit(0)

        else:

            flag=0

if(flag==0):
    print("not equal")
```

Output:

a).Whether list are of same length

b).whether list sums to same value

```
2lists_sum
C:\Users\jomol\AppData\Local\Programs\Python\Python310\python.exe
sum of list1 and list2 9 9

Process finished with exit code 0
```

c). whether any value occur in both

```
list1 [2, 3, 4]
list2 [1, 3, 5]
value in both list1 and list2 3

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.

Source Code:

```
string1=input("enter the string\t")
ch=string1[0]
for c in string1[1:-1]:
    if c==ch:
        b=string1.replace(c,'$')
print(ch+b[1:])
```

Output:

```
enter the string    onion
oni$n

Process finished with exit code 0
```

Program no:8

Aim:

Create a string from given string where first and last characters exchanged.

Source Code:

```
str = input("Enter a string :-")
new_str = str[-1] + str[1:-1] + str[0]
print(new_str)
```

Output:

```
Enter a string :-python
nythop

Process finished with exit code 0
```

Program no:9

Aim:

Accept the radius from user and find area of circle.

Source Code:

```
r=int(input("enter the radius of circle"))
area=3.14*r*r
print("area of a circle",area)
```

Output:

```
enter the radius of circle 4
area of a circle 50.24

Process finished with exit code 0
```

Program no:10

Aim:

Find biggest of 3 numbers entered.

Source Code:

```
a=int(input("enter the first number"))
b=int(input("enter the second number"))
c=int(input("enter the third number"))
if(a>b and a>c):
    print("biggest number is",a)
elif(b>c and b>a):
    print("biggest number is",b)
else:
    print("biggest number is",c)
```

Output:

```
enter the first number 3
enter the second number2
enter the third number1
biggest number is 3

Process finished with exit code 0
```

Program no:11

Aim:

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

Source Code:

```
filename=input("Enter the file name:")
f_extns=filename.split(".")
print("Extension.",f_extns[-1])
```

Output:

```
Enter the file name:samplefile.py
Extension. py

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:

```
color=input("Enter the list of colours separated by commas:")
color_list=color.split(",")
print(color_list)
print("First color:",color_list[0])
print("Second color:",color_list[-1])
```

Output:

```
Enter the list of colours separated by commas:red,green,blue
['red', 'green', 'blue']
First color: red
Second color: blue

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 a number n: "))
temp=str(n)
a=temp+temp
b=temp+temp+temp
c=n+int(a)+int(b)
print("The value is:",c)
```

Output:

```
Enter a number n: 5
The value is: 615

Process finished with exit code 0
```

Program no:14

Aim:

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

Source Code:

```
list1=set(["red","white","blue"])
list2=set(["red","white","black","brown","green"])
a=list1.difference(list2)
b=list(a)
print("The colour not found in list2 is:",b)
```

Output:

```
The colour not found in list2 is: ['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:

```
str = input("Enter a string :-")
new_str = str[-1] + str[1:-1] + str[0]
print(new_str)
```

Output:

```
Enter a string :-python
nythop

Process finished with exit code 0
```

Program no:16

Aim:

Sort dictionary in ascending and descending order.

Source Code:

```
s={'maths':40,'physics':24,'english':41}
l=list(s.items())
l.sort()
dict1=dict(l)
print('Ascending order of the items in dictionary',dict1)
l=list(s.items())
l.sort(reverse=True)
dict=dict(l)
print("Descending order of the items in dictionary",dict)
```

Output:

```
Ascending order of the items in dictionary {'english': 41, 'maths': 40, 'physics': 24}
Descending order of the items in dictionary {'physics': 24, 'maths': 40, 'english': 41}

Process finished with exit code 0
```

Program no:17

Aim:

Merge two dictionaries.

Source Code:

```
d1={'a':10,'b':12}
d2={'c':13,'d':14}
print("first items in the dictionary",d1)
print("second items in the dictionary",d2)
d1.update(d2)
print("After merging the dictionary items:",d1)
```

Output:

```
first items in the dictionary {'a': 10, 'b': 12}
second items in the dictionary {'c': 13, 'd': 14}
After merging the dictionary items: {'a': 10, 'b': 12, 'c': 13, 'd': 14}

Process finished with exit code 0
```

Program no:18

Aim:

Find gcd of 2 numbers.

Source Code:

```
import math
a=int(input("enter first number:"))
b=int(input("enter second number:"))
print("gcd of",a,"and",b,"is",math.gcd(a,b))
```

Output:

```
enter first number:9
enter second number:6
gcd of 9 and 6 is 3

Process finished with exit code 0
```

Program no:19

Aim:

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

Source Code:

```
list = [3,4,5,2,1,8]
print("Original list:",list)
for i in list:
   if i%2 == 0:
      list.remove(i)
print("after deleted even numbers:",list)
```

Output:

```
Original list: [3, 4, 5, 2, 1, 8]
after deleted even numbers: [3, 5, 1]

Process finished with exit code 0
```

Program no:20

Aim:

Program to find the factorial of a number

Source Code:

```
n=int(input("enter the number:"))
fact=1
for i in range(1,n+1):
    fact = fact*i
print("the factorial of",n,"is:",fact)
```

Output:

```
enter the number:5
the factorial of 5 is: 120

Process finished with exit code 0
```

Program no:21

Aim:

Generate Fibonacci series of N terms

Source Code:

```
n=int(input("enter the number:"))
a=0
b=1
sum=0
count=1
print("fibonacci series are:")
while(count<=n):
    print(sum)
    count+=1
    a=b
    b=sum
    sum=a+b
```

Output:

```
enter the number:5
fibonacci series are:
0
1
1
2
3

Process finished with exit code 0
```

Program no:22

Aim:

Find the sum of all items in a list.

Source Code:

```
list1=[2,4,6,8]
print("the given list are",list1)
print("sum of all lists are:",sum(list1))
```

Output:

```
the given list are [2, 4, 6, 8]
sum of all lists are: 20

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:

```
lower=int(input("Enter the lower limit:"))
upper=int(input("Enter the upper limit:"))
list=[x for x in range(lower,upper+1) if x**0.5==int(x**0.5)]
print("The four digit number form a perfect square is",list)
```

Output:

```
Enter the lower limit:2010
Enter the upper limit:2030
The four digit number form a perfect sqaure is: [2025]

Process finished with exit code 0
```

Program no:24

Aim:

Display the given pyramid with step number accepted from user.

Source Code:

```
num=int(input("Enter the number:"))
for i in range(1,num+1):
        for j in range(1,i+1):
            print(i*j,end=" ")
        print("\n")
```

Output:

```
display_pyramid
Enter the number:4
1

2 4

3 6 9

4 8 12 16
```

Program no:25

Aim:

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

Source Code:

```
str1=input("Enter a string:")
count=0
for i in range(0,len(str1)):
    if(str1[i]!=''):
        count+=1
print("number of characters in string:",str(count))
```

Output:

```
Enter a string:python
number of characters in 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:

```
a=input("Enter a string")
if a[-3:]=='ing' and len(a)>4:
    print(a[:-3:]+'ly')
elif a[-3:]!='ing' or len(a)>=0:
    print(a+'ing')
```

Output:

```
Enter a string programming
 programmly

Process finished with exit code 0
```

Program no:27

Aim:

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

Source Code:

```
list1=['apple','orange','mango']
longest = 0
for words in list1:
    if len(words) > longest:
        longest = len(words)
        longest_word = words
print("The longest word is", longest_word, "with length",
len(longest_word))
```

Output:

```
The longest word is orange with length 6

Process finished with exit code 0
```

Program no:28

Aim:

Construct following pattern using nested loop

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

Source Code:

```
a=int(input("Enter the limit:"))
for i in range(1,a):
   for j in range(1,i+1):
        print("*",end="")
   print("\n")
for i in range(a,0,-1):
    for j in range(1,i+1):
        print("*",end=" ")
    print("\n")
```

Output:

```
Enter the limit:4
*

**

***

* * * *

* * *

* *

*


Process finished with exit code 0
```

Program no:29

Aim:

Generate all factors of a number.

Source Code:

```
n=int(input("enter the number:"))
print("the factors of",n,"is")
for i in range(1,n+1):
    if n%i==0:
        print(i)
```

Output:

```
enter the number:6
the factors of 6 is
1
2
3
6

Process finished with exit code 0
```

Program no:30

Aim:

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

Source Code:

```
sq=lambda x:x**2
print("area of a square is",sq(2))
rect=lambda l,b:l*b
print("area of a rectangle is",rect(2,4))
tri=lambda b,h:0.5*b*h
print("area of a triangle is",tri(10,5))
```

Output:

```
area of a square is 4
area of a rectangle is 8
area of a triangle is 25.0

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:

```
def Circle(r):
    print("Area = ", 3.14*r**2)
    print("Perimeter = ", 2*3.14*r)
def Rectangle(l,b):
    print("Area = ", l*b)
    print("Perimeter = ", (2*l)+(2*b))

def Cuboid(l,w,h):
    print("Perimeter of cuboid= ", 4*(l+w+h))
    print("Area of cuboid= ", 2*l*w + 2*l*h + 2*h*w)

def Sphere(r):
    print("Perimeter of sphere = ", 2*3.14*r)
    print("Area of sphere= ", 4*3.14*r**2)

import Graphics.Circle
print("CIRCLE")
r = int(input("Enter radius "))
Graphics.Circle.Circle(r)

import Graphics.Rectangle
print("RECTANGLE")
```

```
l = int(input("Enter length "))
b = int(input("Enter breadth "))
Graphics.Rectangle.Rectangle(l,b)

from Graphics.ThreeDgraphics import Cuboid
print("CUBOID")
l = int(input("Enter length "))
w = int(input("Enter width "))
h = int(input("Enter height "))
Graphics.ThreeDgraphics.Cuboid.Cuboid(l,w,h)

from Graphics.ThreeDgraphics import Sphere
print("SPHERE")
r = int(input("Enter radius "))
Graphics.ThreeDgraphics.Sphere.Sphere(r)
```

Output:

```
PS D:\clg labs\python prgrm\programming Lab\lab-git\labs> pyt
enter the length        10
enter the width 20
Area is 200.0 and Perimeter is 60.0
enter the radius        5
Area is 78.53981633974483 and Perimeter is 31.41592653589793
enter the length        10
enter the breadth       5
enter the height        9
Area is 308.0 and Perimeter is 96.0
enter the radius        10
Area is 20.0 and Perimeter is 1256.6370614359173
```

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 rectangle:
    def_init_(self,l1,b1):
        self.length=l1
        self.breadth=b1
    def area(self):
        return(self.length*self.breadth)
    def perimeter(self):
        print("Perimeter=", 2*(self.length+self.breadth))
    def compare(self,obj):
        if(self.area()>obj.area()):
            print("Rectangle1 is large!!")
        else:print("Rectangle2 is large!!")
R1=rectangle(2,5)
R1.area()
R1.perimeter()
R2=rectangle(4,8)
R2.area()
R2.perimeter()
R1.compare(R2)
```

Output:

```
PS D:\clg labs\python prgrm\programming Lab\lab-git\labs>
Enter length of rectangle: 10
Enter breadth of rectangle: 20
Enter length of rectangle: 5
Enter breadth of rectangle: 30
Area of rectangle 1 is 200.0 and perimeter is 60.0:
Area of rectangle 2 is 150.0 and perimeter is 70.0:
True
```

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, account_number, name, account_type, balance):
        self.account_number = account_number
        self.name = name
        self.account_type = account_type
        self.balance = balance

    def deposit(self, amount):
        self.balance += amount
        print("Deposit of {} successful".format(amount))
        print("Current balance is {}".format(self.balance))

    def withdraw(self, amount):
        if amount > self.balance:
            print("Insufficient balance")
        else:
            self.balance -= amount
            print("Withdrawal of {} successful".format(amount))
            print("Current balance is {}".format(self.balance))

num=int(input("Enter account number: "))
name=input("Enter name: ")
acctype=input("Enter account type: ")
```

```
bal=int(input("Enter balance: "))
bnk=Bank(num,name,acctype,bal)
print("Account number: ",bnk.account_number)
print("Name: ",bnk.name)
print("Account type: ",bnk.account_type)
print("Balance: ",bnk.balance)
bnk.withdraw(int(input("Enter amount to withdraw: ")))
bnk.deposit(int(input("Enter amount to deposit: ")))
```

Output:

```
Enter account number: 302314
Enter name: ponnu
Enter account type: savings account
Enter balance: 30000
Account number:  302314
Name:  ponnu
Account type:  savings account
Balance:  30000
Enter amount to withdraw: 1000
Withdrawal of 1000 successful
Current balance is 29000
Enter amount to deposit: 2000
Deposit of 2000 successful
Current balance is 31000

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 Rectangle:
    def_init_(self, length, width):
        self._length = length
        self._width = width
        self.area=length*width

    def_lt_(self, other):
        if self.area<other.area:
            return "Reactangle 1 is smaller in Area"
        else:
            return "Reactangle 2 is smaller in Area"
r1=Rectangle(50,20)
r2=Rectangle(30,10)
print(r1<r2)
```

Output:

```
Reactangle 2 is smaller in Area

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:
    def_init_(self, hour, minute, second):
        self._hour = hour
        self._minute = minute
        self._second = second

    def_add_(self, other):
        return 'time is: ' + str(self._hour + other._hour) + ':' + str(self._minute +
other._minute) + ':' + str(self._second + other._second)


h=int(input("enter the hour: "))
m=int(input("enter the minute: "))
s=int(input("enter the second: "))
h1=int(input("enter the hour: "))
m1=int(input("enter the minute: "))
s1=int(input("enter the second: "))
t1=Time(h,m,s)
t2=Time(h1,m1,s1)
print(t1+t2)
```

Output:

```
enter the hour: 1
enter the minute: 20
enter the second: 5
enter the hour: 3
enter the minute: 30
enter the second: 10
time is: 4:50:15

Process finished with exit code 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, Pubname):
        self.Pubname = Pubname
    def display(self):
        print("Publisher name is:", self.Pubname)
class Book(Publisher):
    def_init_(self, Pubname, title, author):
        Publisher._init_(self, Pubname)
        self.title = title
        self.author = author
    def display(self):
        print("Title:", self.title)
        print("Author:", self.author)
class Python(Book):
    def_init_(self, Pubname, title, author, price, no_of_pages):
        Book._init_(self, Pubname, title, author)
        self.price = price
        self.no_of_pages = no_of_pages
    def display(self):
        print("Title:", self.title)
        print("Author:", self.author)
        print("Publisher:",self.Pubname)
```

```
        print("Price:", self.price)
        print("No of pages:", self.no_of_pages)
b1 = Python("New India", "Python For Babies", "Mark Liyo", 600, 900)
b1.display()
```

Output:

```
Title: Python For Babies
Author: Mark Liyo
Publisher: New India
Price: 600
No of pages: 900

Process finished with exit code 0
```

Program no: 37

Aim:

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

Source Code:

```
file=open('demo.txt')
lst=[]
for line in file:
    lst.append(line)
print(lst)
file.close()
```

Output:

```
PS D:\clg labs\python prgrm\programming Lab\lab-git\labs> python
['python \n', 'Django \n', 'flask\n', 'Java \n', 'Javascript']
```

Program no: 38

Aim:

Python program to copy odd lines of one file to other

Source Code:

```
newFile = open("States.txt","w")
newFile.write("Goa \nKerala \nTamilnadu \nBangal\nPunjab")
newFile.close()

readFile = open("States.txt","r")
lines = readFile.readlines()
print(lines)
readFile.close()

oddFile = open("oddcontent.txt","w")
for i in range(0,len(lines),2):
    oddFile.write(lines[i])
oddFile.close()

readFile = open("oddcontent.txt","r")
lines = readFile.readlines()
print(lines)
readFile.close()
```

Output:

```
PS D:\clg labs\python prgrm\programming Lab\lab-git\labs> pyth
['Goa \n', 'Kerala \n', 'Tamilnadu \n', 'Bangal\n', 'Punjab']
['Goa \n', 'Tamilnadu \n', 'Punjab']
```

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_
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),For
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),Fo
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),"
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.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/File/3.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
BDCQ.SF1AA2CA,2016.06,1116.386,,F,Dollars,6,Business, Data, Collection, -, BDC,Industry, by, financial, variable,Sales, (operating, income),Forestry, and, Logging,Current, prices,U
BDCQ.SF1AA2CA,2016.09,1070.874,,F,Dollars,6,Business, Data, Collection, -, BDC,Industry, by, financial, variable,Sales, (operating, income),Forestry, and, Logging,Current, prices,U
BDCQ.SF1AA2CA,2016.12,1054.408,,F,Dollars,6,Business, Data, Collection, -, BDC,Industry, by, financial, variable,Sales, (operating, income),Forestry, and, Logging,Current, prices,U
BDCQ.SF1AA2CA,2017.03,1010.665,,F,Dollars,6,Business, Data, Collection, -, BDC,Industry, by, financial, variable,Sales, (operating, income),Forestry, and, Logging,Current, prices,U
BDCQ.SF1AA2CA,2017.06,1233.7,,F,Dollars,6,Business, Data, Collection, -, BDC,Industry, by, financial, variable,Sales, (operating, income),Forestry, and, Logging,Current, prices,Una
BDCQ.SF1AA2CA,2017.09,1282.436,,F,Dollars,6,Business, Data, Collection, -, BDC,Industry, by, financial, variable,Sales, (operating, income),Forestry, and, Logging,Current, prices,U
BDCQ.SF1AA2CA,2017.12,1290.82,,F,Dollars,6,Business, Data, Collection, -, BDC,Industry, by, financial, variable,Sales, (operating, income),Forestry, and, Logging,Current, prices,Una
BDCQ.SF1AA2CA,2018.03,1412.007,,F,Dollars,6,Business, Data, Collection, -, BDC,Industry, by, financial, variable,Sales, (operating, income),Forestry, and, Logging,Current, prices,U
BDCQ.SF1AA2CA,2018.06,1488.055,,F,Dollars,6,Business, Data, Collection, -, BDC,Industry, by, financial, variable,Sales, (operating, income),Forestry, and, Logging,Current, prices,U
BDCQ.SF1AA2CA,2018.09,1497.678,,F,Dollars,6,Business, Data, Collection, -, BDC,Industry, by, financial, variable,Sales, (operating, income),Forestry, and, Logging,Current, prices,U
BDCQ.SF1AA2CA,2018.12,1570.507,,F,Dollars,6,Business, Data, Collection, -, BDC,Industry, by, financial, variable,Sales, (operating, income),Forestry, and, Logging,Current, prices,U
BDCQ.SF1AA2CA,2019.03,1393.749,,F,Dollars,6,Business, Data, Collection, -, BDC,Industry, by, financial, variable,Sales, (operating, income),Forestry, and, Logging,Current, prices,U
BDCQ.SF1AA2CA,2019.06,1517.143,,F,Dollars,6,Business, Data, Collection, -, BDC,Industry, by, financial, variable,Sales, (operating, income),Forestry, and, Logging,Current, prices,U
BDCQ.SF1AA2CA,2019.09,1381.514,,F,Dollars,6,Business, Data, Collection, -, BDC,Industry, by, financial, variable,Sales, (operating, income),Forestry, and, Logging,Current, prices,U
BDCQ.SF1AA2CA,2019.12,1370.985,,F,Dollars,6,Business, Data, Collection, -, BDC,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),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),For
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),Fo
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),"
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_
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),For
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),Fo
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),"
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
```