**Program 1:**

**Write a Python Program to Convert Celsius to Fahrenheit and vice-a-versa.**

=====================================================================

Fahrenheit = float(input("Enter temperature in fahrenheit: "))

Celsius = ((Fahrenheit-32)\*5)/9

print("Temperature in Celsius is: ");

print(Celsius);

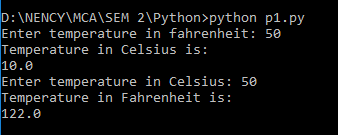
celsius = float(input('Enter temperature in Celsius: '))

fahrenheit = (celsius \* 1.8) + 32

print("Temperature in Fahrenheit is: ");

print(fahrenheit);

**Output:**



**Program 2:**

**Write a program in python to swap two variables without using temporary variable.**

=====================================================================

a,b=20,30

print ("Before swapping : ")

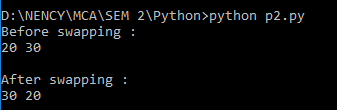
print(a,b)

print ("\nAfter swapping : ")

a,b=b,a

print(a,b)

**Output:**



**Program 3:**

**Write a Python Program to Convert Decimal to Binary, Octal and Hexadecimal.**

=====================================================================

dec = int(input("Enter Decimal Value="))

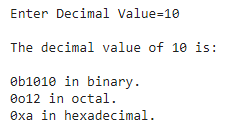
print("\nThe decimal value of", dec, "is:\n")

print(bin(dec), "in binary.")

print(oct(dec), "in octal.")

print(hex(dec), "in hexadecimal.")

**Output:**

****

**Program 4:**

**Write a program to make a simple calculator (using functions).**

=====================================================================

def add(x, y):

return x + y

def subtract(x, y):

return x - y

def multiply(x, y):

return x \* y

def divide(x, y):

return x / y

def modulo(x, y):

return x % y

def power(x, y):

return x \*\* y

print("Select operation.")

print("1.Add")

print("2.Subtract")

print("3.Multiply")

print("4.Divide")

print("5.Modulo")

print("6.Power")

while True:

choice = input("Enter choice(1/2/3/4/5/6): ")

if choice in ('1', '2', '3', '4', '5', '6'):

num1 = float(input("Enter first number: "))

num2 = float(input("Enter second number: "))

if choice == '1':

print(num1, "+", num2, "=", add(num1, num2))

elif choice == '2':

print(num1, "-", num2, "=", subtract(num1, num2))

elif choice == '3':

print(num1, "\*", num2, "=", multiply(num1, num2))

elif choice == '4':

print(num1, "/", num2, "=", divide(num1, num2))

elif choice == '5':

print(num1, "%", num2, "=", modulo(num1, num2))

elif choice == '6':

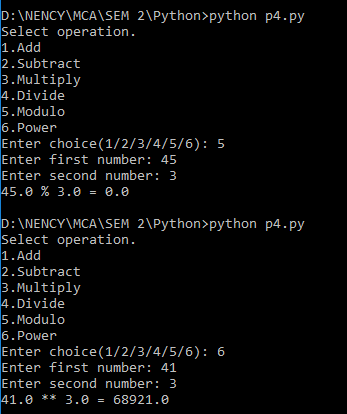
print(num1, "\*\*", num2, "=", power(num1, num2))

break

else:

print("Invalid Input")

**Output:**



**Program 5:**

**Write a program in python to find out maximum and minimum number out of three user entered number.**

=====================================================================

number1 = int(input('Enter First number : '))

number2 = int(input('Enter Second number : '))

number3 = int(input('Enter Third number : '))

def largest(num1, num2, num3):

if (num1 > num2) and (num1 > num3):

largest\_num = num1

elif (num2 > num1) and (num2 > num3):

largest\_num = num2

else:

largest\_num = num3

print("\nThe largest of the 3 numbers is : ", largest\_num)

def smallest(num1, num2, num3):

if (num1 < num2) and (num1 < num3):

smallest\_num = num1

elif (num2 < num1) and (num2 < num3):

smallest\_num = num2

else:

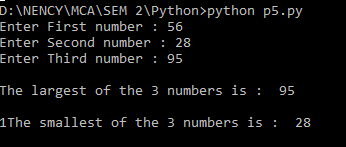
smallest\_num = num3

print("\n1The smallest of the 3 numbers is : ", smallest\_num)

largest(number1, number2, number3)

smallest(number1, number2, number3)

**Output:**



**Program 6:**

**Write a program which will allow user to enter 10 numbers and display largest odd number from them. It will display appropriate message in case if no odd number is found.**

=====================================================================

n=int(input("Enter the number of elements to be in the list:"))

b=[]

for i in range(0,n):

a=int(input("Element: "))

b.append(a)

c=[]

for i in b:

if(i%2!=0):

c.append(i)

c.sort()

count1=0

for k in c:

count1=count1+1

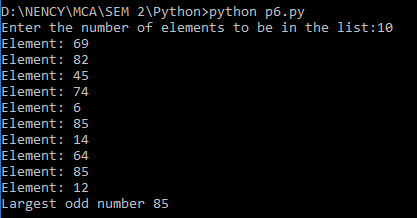
if count1>0:

print("Largest odd number",c[count1-1])

else:

print("There is no odd number is available in list")

**Output:**



**Program 7:**

**Write a Python program to check if the number provided by the user is an Armstrong number.**

=====================================================================

num = int(input("Enter a number: "))

sum = 0

temp = num

while temp > 0:

digit = temp % 10

sum += digit \*\* 3

temp //= 10

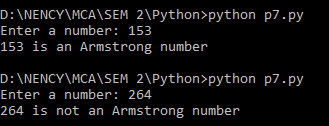
if num == sum:

print(num,"is an Armstrong number")

else:

print(num,"is not an Armstrong number")

**Output:**



**Program 8:**

**Write a Python program to check if the number provided by the user is a palindrome or not.**

=====================================================================

n=int(input("Enter number:"))

temp=n

rev=0

while(n>0):

dig=n%10

rev=rev\*10+dig

n=n//10

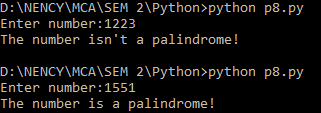
if(temp==rev):

print("The number is a palindrome!")

else:

print("The number isn't a palindrome!")

**Output:**



**Program 9:**

**Write a Python program to perform following operation on given string input:**

**a) Count Number of Vowel in given string**

**b) Count Length of string (do not use Len ())**

**c) Reverse string**

**d) Find and replace operation**

**e) Check whether string entered is a palindrome or not.**

=====================================================================

**a) Count Number of Vowel in given string**

string=input("Enter string:")

vowels=0

a=[]

for i in string:

if(i=='a' or i=='e' or i=='i' or i=='o' or i=='u' or i=='A' or i=='E' or i=='I' or i=='O' or i=='U'):

vowels=vowels+1

a.append(i)

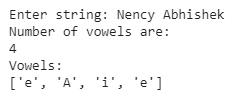
print("Number of vowels are:")

print(vowels)

print("Vowels:")

print(a)

**Output:**



**b) Count Length of string (do not use Len ())**

str = input("Enter a string: ")

counter = 0

for s in str:

counter = counter+1

print("Length of the input string is:", counter)

**Output:**

****

**c) Reverse string**

s = input("Enter a string: ")

print (s[::-1])

**Output:**

****

**d) Find and replace operation**

string = "Nency Nency Nency Nency Nency"

print(string.replace("Nency", "Abhi"))

# Prints the string by replacing only 3 occurrence

print(string.replace("Nency", "Abhishek", 3))

**Output:**



**e) check whether string entered is a palindrome or not**

string=input("Enter string:")

if(string==string[::-1]):

print("The string is a palindrome")

else:

print("The string isn't a palindrome")

**Output:**

****

**Program 10:**

**Define a procedure histogram () that takes a list of integers and prints a histogram to the screen. For example, histogram ([4, 9, 7]) should print the following:**

**\*\*\*\***

**\*\*\*\*\*\*\*\*\***

**\*\*\*\*\*\*\***

=====================================================================

def histogram( items ):

for n in items:

output = ''

times = n

while( times > 0 ):

output += '\*'

times = times - 1

print(output)

histogram([4, 9, 7])

**Output:**

****

**Program 11:**

**Write a program in python to implement Fibonacci series up to user entered number. (Use recursive Function)**

=====================================================================

def Fibonacci\_series(Number):

if(Number == 0):

return 0

elif(Number == 1):

return 1

else:

return (Fibonacci\_series(Number - 2) + Fibonacci\_series(Number - 1))

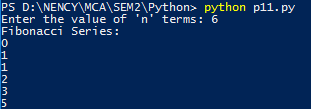
n = int(input("Enter the value of 'n' terms: "))

print("Fibonacci Series:")

for n in range(0, n):

print(Fibonacci\_series(n))

**Output:**



**Program 12:**

**Write a program in python to implement Factorial series up to user entered number. (Use recursive Function)**

=====================================================================

def recur\_factorial(n):

if n == 1:

return n

else:

return n\*recur\_factorial(n-1)

num = int(input("Enter a number: "))

if num < 0:

print("Sorry, factorial does not exist for negative numbers")

elif num == 0:

print("The factorial of 0 is 1")

else:

print("The factorial of",num,"is",recur\_factorial(num))

**Output:**

****

**Program 13:**

**Write a program in Python to implement readline, readlines, write line and writelines file handling mechanisms.**

=====================================================================

try:

file=open('tmp.txt', 'r')

file2=open('tmp2.txt', 'w')

print("\nreadline")

r1=file.readline()

print(r1)

file2.write(r1)

print("\nreadlines")

r2=file.readlines()

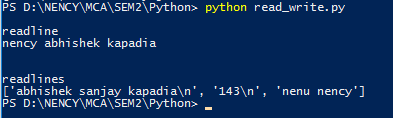
print(r2)

file2.writelines(r2)

except FileNotFoundError:

print ("File not found")

**Output:**



**Program 14:**

**Write a program in python to implement Salary printing file read operation. (File format: Employee No, name, deptno, basic, DA, HRA, Conveyance) should perform below operations.**

**a) Print Salary Slip for given Employee Number**

**b) Print Employee List for Given Department Number .**

=====================================================================

import pickle

ch=0

def insert():

emp= open('p14.dat','wb')

n = int(input("Enter number of data to be added -> "))

for i in range(n):

eid = int(input("\n\nEnter employee id -> "))

ename = input("Enter employee name -> ")

edeptno = int(input("Enter department no -> "))

ebasic = int(input("Enter basic salary -> "))

eda = ebasic\*0.05;

ehra = ebasic\*0.1;

econveyance = ebasic\*0.12;

pickle.dump([eid,ename,edeptno,ebasic,eda,ehra,econveyance],emp)

emp.close()

def eno():

f= open('p14.dat','rb')

id = int(input("Enter employee id -> "))

while True:

try:

obj = pickle.load(f)

if obj[0] == id:

salary = obj[3]+obj[4]+obj[5]+obj[6]

print("\n\nEmployee Id = ",obj[0])

print("Employee Name = ",obj[1])

print("Department No = ",obj[2])

print("Salary = ",salary)

except EOFError:

exit()

break

f.close()

def dno():

f= open('p14.dat','rb')

id = int(input("Enter department no -> "))

while True:

try:

obj = pickle.load(f)

if obj[2] == id:

salary = obj[3]+obj[4]+obj[5]+obj[6]

print("\n\nEmployee Id = ",obj[0])

print("Employee Name = ",obj[1])

print("Department No = ",obj[2])

print("Salary = ",salary)

except EOFError:

exit()

break

f.close()

print("\n\n Select operation")

print("===============================================")

print(" 1.Add Record")

print(" 2.Salary Slip for given Employee Number")

print(" 3.Employee List for Given Department Number")

print(" 4.Exit\n\n")

while True:

choice = int(input("Enter choice(1/2/3/4): "))

if choice in (1, 2, 3, 4):

if choice == 1:

insert()

elif choice == 2:

eno()

elif choice == 3:

dno()

elif choice == 4:

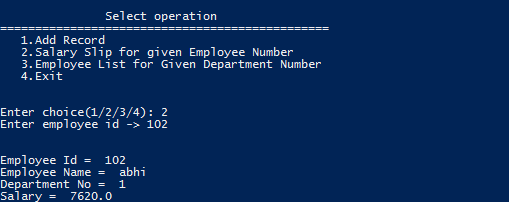
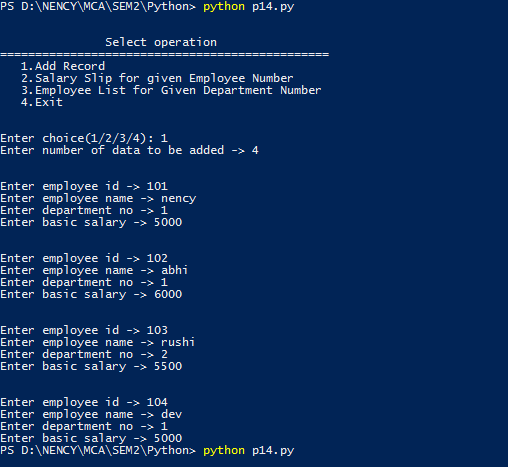
exit()

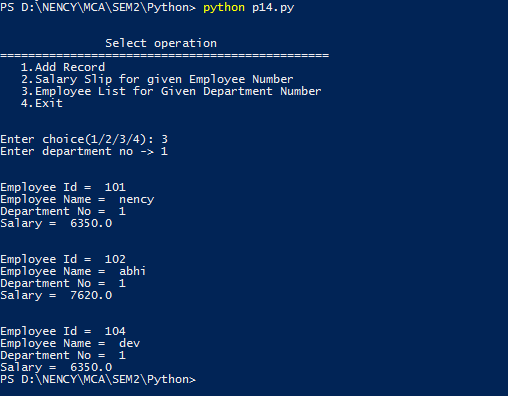
break

else:

print("Invalid Input")

**Output:**

****

****

**Program 15:**

**Write a program in python to implement Railway Reservation System using file handling technique. System should perform below operations.**

**a. Reserve a ticket for a passenger.**

**b. List information all reservations done for today’s trains**

=====================================================================

import pickle

from datetime import date, datetime

def insert():

ticket= open('p15.dat','ab')

n = int(input("Enter number of ticket to be reserved -> "))

for i in range(n):

name = input("\n\nEnter passenger name -> ")

trainno = int(input("Enter train no -> "))

date1 = str(input("Enter date(yyyy-mm-dd) -> "))

date = str(datetime.strptime(date1, "%Y-%m-%d"))

#print(date)

pickle.dump([name,trainno,date],ticket)

ticket.close()

def list():

f= open('p15.dat','rb')

today = str(date.today())

date2 = today + " 00:00:00";

while True:

try:

obj = pickle.load(f)

date1 = str(obj[2])

if date1 == date2:

print("\n\nPassenger name -> ",obj[0])

print("Train no -> ",obj[1])

print("Date -> ",obj[2])

except EOFError:

exit()

break

f.close()

print("\n\n Select operation")

print("===============================================")

print(" 1.Reserve a ticket for a passenger")

print(" 2.List information all reservations done for today’s trains")

print(" 3.Exit\n\n")

while True:

choice = int(input("Enter choice(1/2/3): "))

if choice in (1, 2, 3):

if choice == 1:

insert()

elif choice == 2:

list()

elif choice == 3:

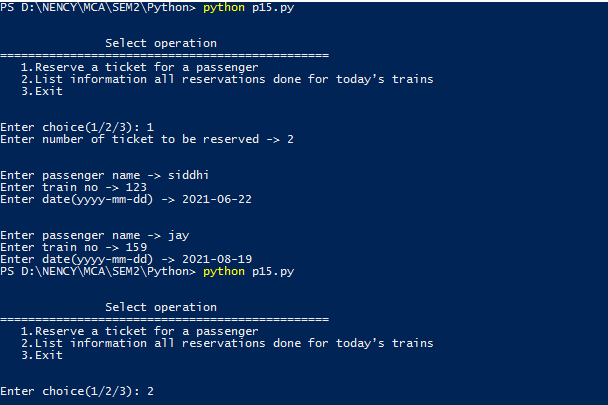
exit()

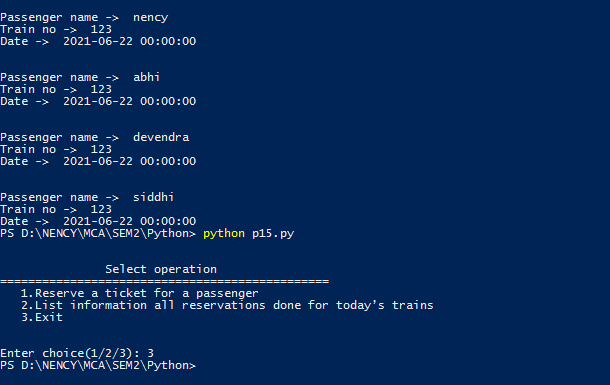
break

else:

print("Invalid Input")

**Output:**

****

****

**Program 16:**

**Create Web Database Application “Address Book” with options to**

**a) add/ insert a record**

**b) modify a record**

**c) display a record**

**d) delete a record** =====================================================================

import mysql.connector

#Create the connection object

myconn = mysql.connector.connect(

host = "localhost",

user = "root",

passwd = "",

database="test"

)

mycursor= myconn.cursor()

mycursor.execute("""create table addressbook(

id int primary key,

name varchar(20),

address varchar(30),

city varchar(20),

phoneno varchar(10))""")

#insert record

mycursor.execute("insert into addressbook(id,name,address,city,phoneno) values (1,'Nency','Bhatar','Surat','7245782432');")

mycursor.execute("insert into addressbook(id,name,address,city,phoneno) values (2,'Pooja','Chala','Vapi','9832452334');")

mycursor.execute("insert into addressbook(id,name,address,city,phoneno) values (3,'Priya','Udhna','Surat','7945874330');")

mycursor.execute("insert into addressbook(id,name,address,city,phoneno) values (4,'Khushi','Chanod','Vapi','9016635109');")

mycursor.execute("insert into addressbook(id,name,address,city,phoneno) values (5,'Neema','chhiri','vapi','7945874330');")

print("Data Inserted")

#update record

mycursor.execute("Update addressbook set name='Riya',address='Gunjan' where id=5")

print("Data Updated")

#delete record

mycursor.execute("Delete from addressbook where id=2")

print("Data Deleted")

#select record

mycursor.execute("select \* from addressbook;")

print("Addressbook Records:-:\n")

print(mycursor.fetchall(),end="\n")

**Output:**

Data Inserted

Data Updated

Data Deleted

Addressbook Records:-:

[(1, 'Nency', 'Bhatar', 'Surat', '7245782432'), (3, 'Priya', 'Udhna', 'Surat', '7945874330'), (4, 'Khushi', 'Chanod', 'Vapi', '9016635109'), (5, 'Riya', 'Gunjan', 'vapi', '7945874330')]

**Program 17:**

**Create Web Database Application “Event Registration” with options to**

**a) Event Registration**

**b) Cancel Registration**

**c) display a record**

=====================================================================

import mysql.connector

#Create the connection object

myconn = mysql.connector.connect(

host = "localhost",

user = "root",

passwd = "",

database="test"

)

mycursor= myconn.cursor()

mycursor.execute("""create table EventRegistration(

id int primary key,

name varchar(20),

event\_date date,

charge int )""")

#Event Registration

mycursor.execute("insert into EventRegistration(id,name,event\_date,charge) values (1,'Nency','01-5-2012',900);")

mycursor.execute("insert into EventRegistration(id,name,event\_date,charge) values (2,'Rina','11-2-2021',1000);")

print("Data Inserted")

# Cancel Registration

mycursor.execute("Delete from EventRegistration where id=2")

print("Data Deleted")

# display a record

mycursor.execute("select \* from EventRegistration;")

print("Event Registration Records:-:\n")

print(mycursor.fetchall(),end="\n")

**Output:**

Data Inserted

Data Deleted

Event Registration Records:-

[(1,'Nency','01-5-2012',900)]

**Program 18:**

**Using a data file, draw**

**a. Bar Graph**

**b. Histogram**

**c. Pie Chart**

**d. Line Chart**

=====================================================================

from matplotlib import pyplot as plt

import numpy as np

import pandas as pd

df = pd.read\_excel (r'D:\NENCY\MCA\SEM2\Python\p4\_2.xlsx')

print (df)

# Bar chart

plt.title("Simple Bar")

plt.xlabel('Year')

plt.ylabel('Sales')

plt.bar(df["Year"], df["Sales"], color='red')

plt.show()

# Histogram chart

population\_ages = [22,55,62,45,21,22,34,42,42,4,99,102,110,120,121,122,130,111,115,112,80,75,65,54,44,43,42,48]

bins = [0,10,20,30,40,50,60,70,80,90,100,110,120,130]

plt.title("Simple Histogram")

plt.xlabel('Year')

plt.ylabel('Sales') # Assign the name of the y axis always incresing

plt.hist(population\_ages, bins, histtype='bar', rwidth=0.8)

plt.show()

# Pie chart

ax = df.plot(y="Sales", kind="pie", labels = df["Year"], autopct = '%1.0f%%', legend=False, title='Sales Pie chart')

ax.set(ylabel='') # Hide y-axis label

plt.show()

# Line chart

plt.plot(df["Year"], df["Sales"])

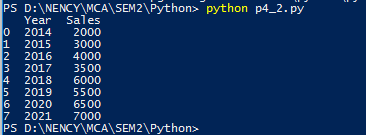
plt.title("Simple Line Plot")

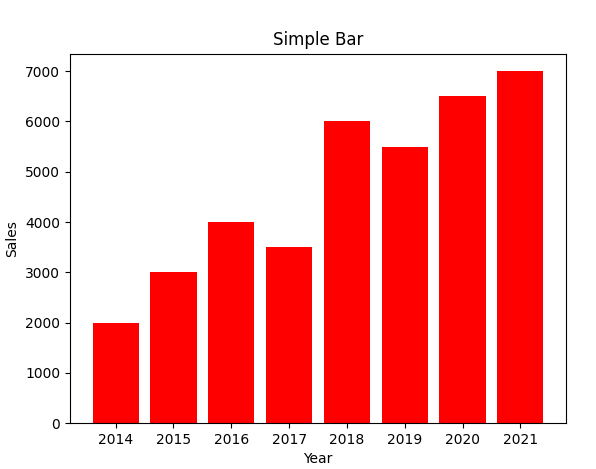
plt.xlabel('Year')

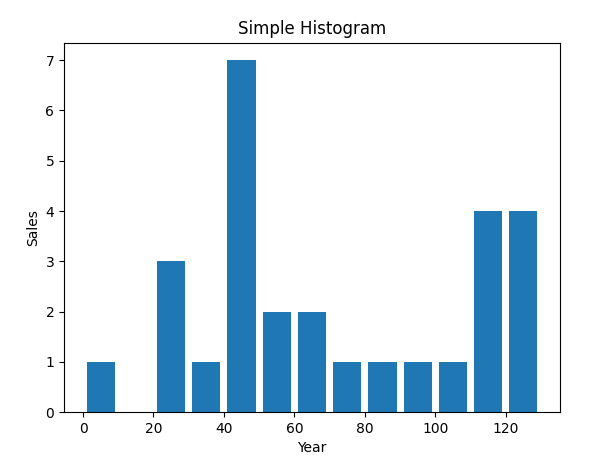
plt.ylabel('Sales')

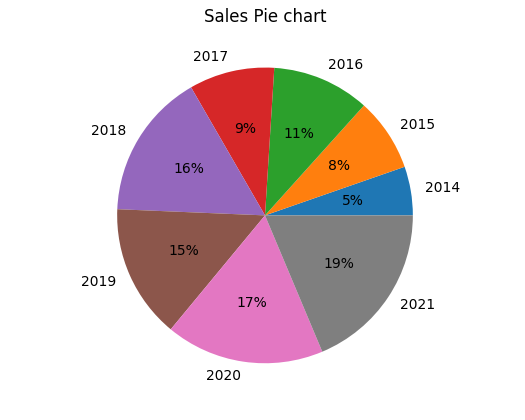
plt.show()

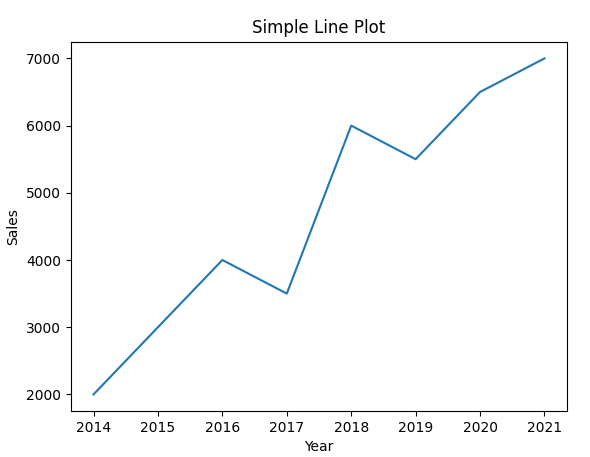
**Output:**











**Program 19:**

**Perform following operations on a CSV file**

**a. Create a data frame from csv file, dictionary, List of tuples**

**b. Operations on Data Frame Shape, head, tail**

**c. Retrieving rows / columns from data frame**

**d. Finding maximum and minimum values**

**e. Displaying statistical information**

**f. Performing queries**

**g. Handling missing data**

=====================================================================

import pandas as pd

data = [('Nency',1 ,21, 9.5),

('Pooja',2, 26, 8),

('Khushboo',3,29,7.6),

('Heena',4,30,9.9),

('Harsh',5,33,8),

]

df = pd.DataFrame(data, columns =['Name', 'Rollno',

'Age','score'])

print ("\n1)Print CSV file:\n")

print(df)

print ("\n2a)The shape of the object is:\n")

print(df.shape)

print ("\n2b)The first two rows of the data frame is:\n")

print(df.head(2))

print ("\n2c)The last two rows of the data frame is:\n")

print(df.tail(2))

print("\n3)Retrieving rows / columns from data frame:\n")

result = df.loc[[0,1,2,3,4]]

print(result)

print("\n4)Minimum Age and Maximum Score:\n")

col1 = df["Age"]

col2 = df["score"]

min = col1.min()

max = col2.max()

print("Age:",min)

print("\nScore:",max)

print("\n5)Display Statistical Information:\n")

info = df.describe(include='all')

print (info)

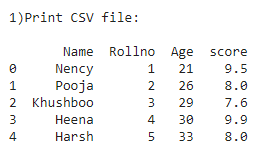
print("\n6)Performing queries(Age should be greater than 28):\n")

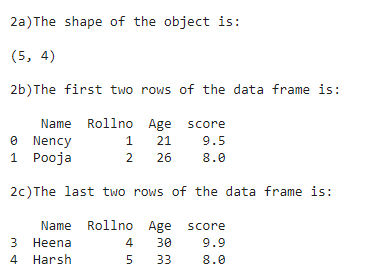
print(df.query('Age > 28'))

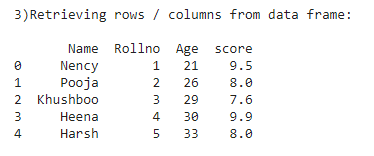
print("\n7)Handling missing data:\n")

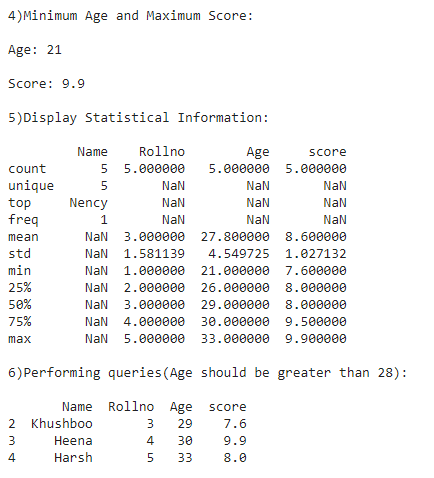
print(df.isnull())

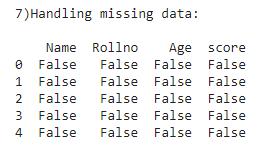
**Output:**











**Program 20:**

**Write a program in python to implement simple interest and compound interest values on chart using PyLab. Show the difference between both. (Note: Use of object oriented paradigm is compulsory.)**

**a) Recognize following strings bit, but, bat, hit, hat or hut**

**b) Match any pair of words separated by a single space, that is, first and last names.**

**c) Match any word and single letter separated by a comma and single space, as in last name, first initial.**

**d) Match simple Web domain names that begin with www and end with a “.com” suffix; for example, www.yahoo.com. Extra Credit: If your regex also supports other high-level domain names, such as .edu, .net, etc. (for example:** [**www.foothill.edu**](http://www.foothill.edu)**).**

**e) Match a street address according to your local format (keep your regex general enough to match any number of street words, including the type designation). For example, American street addresses use the format: 1180 Bordeaux Drive. Make your regex flexible enough to support multi-word street names such as: 3120 De la Cruz Boulevard.**

=====================================================================

import re

#==========================================================================

# a) Recognize following strings bit, but, bat, hit, hat or hut

#==========================================================================

def test\_one(test\_string):

bt = 'bat|bit|but|hat|hit|hut'

m = re.match(bt,test\_string)

if m is not None:

print('Matching success %s'%(m.group()))

else:

print('Match unsuccessful')

if \_\_name\_\_ == '\_\_main\_\_':

print("\n\na) Recognize following strings bit, but, bat, hit, hat or hut")

print("====================================================================")

test\_string\_1 = 'bat'

test\_string\_2 = 'hct'

test\_one(test\_string\_1)

test\_one(test\_string\_2)

#==========================================================================

# b) Match any pair of words separated by a single space, that is, first and last names

#==========================================================================

def test\_two(test\_string):

reg = '[A-Za-z]+\s[A-Za-z]+'

name = re.match(reg,test\_string)

if name is not None:

print(name.group())

if \_\_name\_\_ == '\_\_main\_\_':

print("\n\nb) Match any pair of words separated by a single space, that is, first and last names")

print("============================================================================================")

test\_string\_1 = 'Nency Bhaisha'

test\_string\_2 = 'Abhishek Kapadia'

test\_string\_3 = 'Dev'

test\_two(test\_string\_1)

test\_two(test\_string\_2)

test\_two(test\_string\_3)

#==========================================================================

# c) Match any word and single letter separated by a comma and single space, as in last name, first initial.

#==========================================================================

def test\_three(test\_string):

name = re.findall('\w+',test\_string,re.I)

print(name)

if \_\_name\_\_ == '\_\_main\_\_':

print("\n\nc) Match any word and single letter separated by a comma and single space, as in last name, first initial.")

print("=================================================================================================================")

test\_string\_1 = 'Nency Bhaisha'

test\_string\_2 = 'Abhishek Kapadia'

test\_string\_3 = 'Dev,A,B,C'

test\_three(test\_string\_1)

test\_three(test\_string\_2)

test\_three(test\_string\_3)

#==========================================================================

# d) Match simple Web domain names that begin with www and end with a “.com” suffix;

#for example, www.yahoo.com.

#Extra Credit: If your regex also supports other high-level domain names, such as .edu, .net, etc. (for example: www.foothill.edu).

#==========================================================================

def four(test\_string):

name = re.match('www.+\.com',test\_string)

if name is not None:

print(name.group())

if \_\_name\_\_ == '\_\_main\_\_':

print("\n\nd) Match simple Web domain names that begin with www and end with a “.com” suffix; ")

print("=============================================================================================")

test\_string\_1 = 'www.yahoo.com/'

test\_string\_2 = 'www.amazon.com'

test\_string\_3 = 'www.facebookcom'

test\_four(test\_string\_1)

test\_four(test\_string\_2)

test\_four(test\_string\_3)

#==========================================================================

# e) Match a street address according to your local format (keep your regex general enough to match any number of street words, including the type designation). For example, American street addresses use the format: 1180 Bordeaux Drive. Make your regex flexible enough to support multi-word street names such as: 3120 De la Cruz Boulevard.

#==========================================================================

def test\_five(test\_string):

name = re.match('^[\d]+ [\w+ ]+',test\_string)

if name is not None:

print(name.group())

if \_\_name\_\_ == '\_\_main\_\_':

print("\n\ne) Match a street address according to your local format ")

print("==================================================================")

test\_string\_1 = '3120 De la Cruz Boulevard'

test\_string\_2 = '154 Vishal Society'

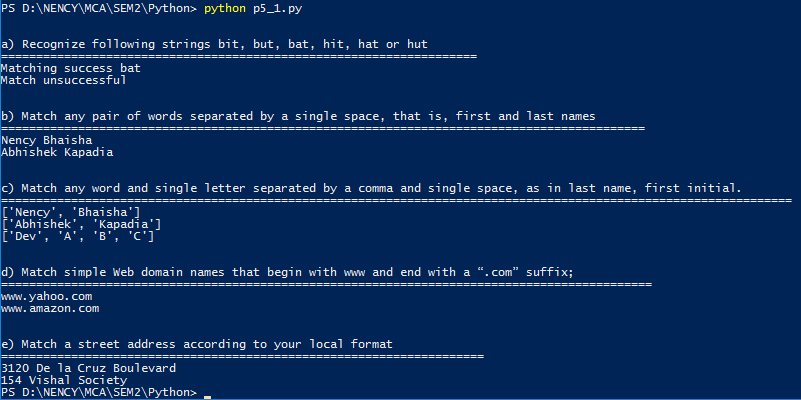
test\_string\_3 = 'abc def cdb'

test\_five(test\_string\_1)

test\_five(test\_string\_2)

test\_five(test\_string\_3)

**Output:**



**Program 21:**

**Create utility script to process telephone numbers such that Area codes (the first set of three-digits and the accompanying hyphen) are optional, that is, your regex should match both 800-555-1212 as well as just 555-1212. Either parenthesized or hyphenated area codes are supported, not to mention optional; make your regex match 800-555-1212, 555-1212, and also (800) 555-1212.**

=====================================================================

#a. Area codes (the first set of three-digits and the accompanying hyphen) are optional,

#that is, your regex should match both 800-555-1212 as well as just 555-1212

import re

text=input("Enter Phone Number : ")

pattern=re.match(r'^D\*(\d{3})\D\*(\d\*)\D\*(\d{4})$',text).group()

print (pattern)

**Output:-**

Enter Phone Number : 555-1212

555-1212

#b. Either parenthesized or hyphenated area codes are supported, not to mention optional;#make your regex match 800-555-1212, 555-1212, and also (800) 555-1212.

import re

text=input("Enter Phone Number : ")

pattern=re.match(r'(\(\d{3}\))?(\d{3})?-?\d{3}-\d{4}',text).group()

print (pattern)

**Output:-**

Enter Phone Number : (800)555-1212

(800)555-1212