R.D & S.H National College & S.W.A Science College

PRACTICAL JOURNAL

OF

INTRODUCTION TO PROGRAMMING WITH PYTHON

NAME: RATHORE SAHIL PRABHUSINGH

ROLL NO. :CS23038

FY.Bsc(CS)

R.D.&S.H.NationalCollege&S.W.A.ScienceCollege

Bandra,Mumbai-400050

DepartmentofComputerScience

CERTIFICATE

ThisistocertifythatMr./Ms.____,

RollNo./SeatNo.__________________of____________classhassatisfactorilycompleted

_____________Practicals,inthesubjectof____,

duringthe **Semester____**ofacademicyear**2023-2024**

FacultyIncharge

Co-ordinator,

DepartmentComputerScience

SignatureofExaminer

INDEX

PRACTICAL:1

1. Writeapythonprogramto readtwo variables with values (i.e.Num1and Num2) and perform basic arithmetic operation on it display the results.

PROGRAM:

```
Num1=float(input('Enterthe valueforNum1:'))
Num2=float(input("EnterthevalueforNum2:"))

sum_result=Num1+Num2
difference_result=Num1-Num2
product_result = Num1 * Num2

ifNum2!=0:
   division_result=Num1/Num2
else:
   division_result="Cannotdividebyzero"

print(f"SumofNum1andNum2:{sum_result}")
print(f"Difference of Num1 and Num2:{difference_result}")
print(f"product of Num1 and Num2:{product_result}")
print(f"DivisionofNum1andNum2:{division_result}")
```

OUTPUT:

```
EnterthevalueforNum1:5
Enter the value for Num2 :3
Sum of Num1 and Num2:8
DifferenceofNum1andNum2:2
productofNum1andNum2:15
DivisionofNum1andNum2:1.66666667
```

2. Write a python program to read three variables with values (i.e. Principal , Rate of Interest and Number of Years) and display the simple interest amount.

PROGRAM:

```
Principal=float(input("EnterthePrincipalamount:"))
Rate_of_Interest = float(input("Enter the Rate of Interest : "))
Number_of_Years = float(input("Enter theNumber ofYears : "))

simple_interest=(Principal*Rate_of_Interest*Number_of_Years)/100

print(f"PrincipalAmount:{Principal}")
print(f"RateofInterest : {Rate_of_Interest}% ")
print(f"NumberofYears:{Number_of_Years}")
print(f"SimpleInterestAmount:{simple_interest}")
```

OUTPUT:

```
EnterthePrincipalamount:15000
Enter the Rate of Interest: 5
EntertheNumberofYears:2
Principal Amount: 15000.0
Rate of Interest: 5.0 %
Number of Years: 2.0
SimpleInterestAmount:1500.0
```

3. Write a python program to read two numbers and swap them:

(i) With temp variable

PROGRAM:

```
num1= float(input("Enter the first number : "))
num2=float(input("Enterthesecondnumber:"))

print(f"Beforeswapping:num1={num1},num2={num2}")
temp=num1
num1=num2
num2=temp

print(f"Afterswappingwithatemporaryvariable: num1={num1},num2={num2}"
```

OUTPUT:

```
Enter the first number: 10

Enterthesecondnumber:20

Beforeswappingnum1=10.0,num2=20.0

Afterswappingwithatemporaryvariable:num1=20.0,num2=10.0
```

(ii) Without temp variable

PROGRAM:

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

num2=float(input("Enterthesecondnumber:"))

print(f"Beforeswapping:num1={num1},num2={num2}")


num1=num1+num2

num2=num1-num2

num1=num1-num2


print(f"Afterswappingwithoutatemporaryvariable:num1={num1},num2={num2}")
```

OUTPUT:

```
Enter the first number: 3

Enterthesecondnumber:2

Beforeswapping:num1=3.0,num2=2.0

Afterswappingwithoutatemporaryvariable:num1=2.0,num2=3.0
```

4. Write a python program to read two numbers and display the larger of the two.

PROGRAM:

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

num2=float(input("Enterthesecondnumber:")) if

num1 > num2 :

        print(f"Thelargernumberis:{num1}") elif

num2 > num1 :

        print(f"Thelargernumberis:{num2}")

else:

        print("Boththenumbersareequal")
```

OUTPUT:

```
Enter the first number: 55.8

Enterthesecondnumber:-55.8

The larger number is: 55.8
```

5. Write a python program to read two numbers and display the smaller of the two.

PROGRAM:

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

num2=float(input("Enterthesecondnumber:")) if

num1 < num2 :

        print(f"Thesmallernumberis:{num1}")

elif num2 > num1 :

        print(f"Thesmallernumberis:{num2}")

else:

        print("Boththenumbersareequal")
```

OUTPUT:

```
Enter the first number: 98

Enterthesecondnumber:30

Both the numbers are equal
```

PRACTICAL:2

1. **Write a python program to read the age of a person and display whether the person is eligible to vote or not.**

PROGRAM:

```
age=int(input("Entertheage:")) if

age >= 18:

        print("Youareeligibletovote.")

else:

        print("Youarenoteligiblevote.")
```

OUTPUT:

```
Entertheage:3

Youarenoteligiblevote.

Enter the age :18

Youareeligibletovote.
```

2. **Write a python program to read the age and gender of a person and display whether the person is eligible to get married or not.(Eligibility to get married:Male 21 years and Female 18 years)**

PROGRAM:

```
age=int(input("Entertheage:"))

gender=input("Enterthegender(Male/Female):")

if(gender.lower()=='male'andage>=21)or(gender.lower()=='female'andage>=18): print("Eligible to

        get married.")

else:

        print("Noteligibletogetmarriedyet.")
```

OUTPUT:

```
Entertheage: 12

Enterthegender(Male/Female):male

Not eligible to get married yet.
```

3. Write a python program to read the co-efficients of a quadratic equation and display its roots if they exist.

PROGRAM:

```
importmath
a=int(input("Enterthecoefficienta:"))
b=int(input("Enterthecoefficientb:"))
c=int(input("Enterthecoefficientc:"))

x =b**2-4*a*c

ifx>0:
   r1=(-b+math.sqrt(x))/(2*a)
   r2=(-b-math.sqrt(x))/(2*a)
   print(f"Therootsarerealanddistinct.Root1={r1},Root2={r2}") elif x
== 0:
   r =-b/ (2*a)
   print(f"Therootisrealandrepeated.Root={r}")
else:
   rP=-b/(2*a)
   iP=math.sqrt(abs(x))/(2*a)
   print(f"Therootsarecomplex.Root1={rP}+{iP}i,Root2={rP}-{iP}i") # Here
rP is real Part, iP is imaginary Part
```

OUTPUT:

```
Enterthecoefficienta:1
Enterthecoefficientb:5
Enterthecoefficientc:2
Therootsarerealanddistinct.Root1=-0.4384471871911697,Root2=-4.561552812808831
```

4. Write a python program to read the electricity units consumed and display the total bill as follows:

Unit Consumed	Fixed Charge	Rate per Unit
0-100	50	Rs.5/-
101-300	100	Rs.7/-
301-500	150	Rs.10/-

>500	200	Rs.15/-

PROGRAM:

```
unit=int(input("Entertheunitconsumed:")) if
(unit >= 0 and unit <= 100):
 pay_amt=unit*5 fixed_charge
 = 50

elif(unit >100andunit<= 300):
 pay_amt=(100*5)+(unit-100)*7
 fixed_charge=100

elif(unit >300andunit<= 500):
 pay_amt=(100*5)+(300-100)* 7+(unit-300)* 10
 fixed_charge=150

else:
 pay_amt=(100*5)+(300-100)*7+(500-300)*10+(unit-500)*15
 fixed_charge=200

Total=pay_amt+fixed_charge

GST = Total * 0.18
Total_with_GST=Total+GST

print("YOURELECTRICITYBILLWITHOUTGSTIS:",Total)
print("YOURELECTRICITYBILLWITHGSTOF18%IS:",Total_with_GST)
```

OUTPUT:

```
Entertheunitconsumed:575
YOUR ELECTRICITY BILL WITHOUT GST IS: 5225
YOURELECTRICITYBILLWITHGSTOF18%IS:6165.5
```

5. Write a python program to read the marks obtained and display the grade obtained as follows:

GRADE	MARKS
O	80.00&Above
A+	70to79.99
A	60to69.99
B+	54to59.99
B	50to53.99
C	45to49.99
D	40to44.99
F	Lessthan40

PROGRAM:

```
marks=float(input("Enterthemarks:"))


ifmarks>=80:
        print("OGRADE")
elifmarks>=70andmarks<=79.99:
        print("A+GRADE")
elifmarks>=60andmarks<=69.99:
        print("AGRADE")
elifmarks>=54andmarks<=59.99:
        print("B+GRADE")
elifmarks>=50andmarks<=53.99:
        print("B GRADE")
elifmarks>=45andmarks<=49.99:
        print("CGRADE")
elifmarks>=40andmarks<=44.99:
        print("DGRADE")
else:
        print("FGRADE")
```

OUTPUT:

```
Enterthemarks:60
A GRADE
Enterthemarks:45.99
```

CGRADE

PRACTICAL:3

1. Write a python program to read a number and display its factorial.

PROGRAM:

```
number=int(input("Enteranumber:")) factorial

= 1

ifnumber<0:

          print("Factorialdoes not exist fornegativenumbers.") elif

number == 0:

          print("Thefactorialof0is:")

else:

    foriinrange(1,number+1):

        factorial *= i

        print(f"Thefactorialof{number}is{factorial}")
```

OUTPUT:

```
Enter a number :4

The factorial of4 is 1

The factorial of4 is 2

The factorial of4 is 6

Thefactorialof4is24
```

2. Write a python program to read a number and display its reverse.

PROGRAM:

```
n=int(input("Enteranumber:"))

while (n != 0):

   a=n%10

   print(a,end="")

   n = n//10
```

OUTPUT:

```
Enteranumber:228
```

3. Write a python program to read a number and display its sum of the digits.

PROGRAM:

```
num=int(input("Enteranumber:"))

sum = 0

whilenum>0:

        a = num% 10

        sum=sum+a num

        //= 10

print("SUMOFDIGITSIS:",sum)
```

OUTPUT:

```
Enteranumber:234

SUM OFDIGITSIS:9
```

4. Write a python program to read a number n calculate

i) $\sum n$

PROGRAM:

```
n=int(input("Enteranumber:")) a

= sum(range(1, n + 1))

print("Sumofnumbersfrom1ton:",a)
```

OUTPUT:

```
Enteranumber:28

Sumofnumbers from1ton:406
```

ii) $\sum n$**2**

PROGRAM:

```
n=int(input("Enteranumber:"))

b=sum(i**2foriinrange(1,n+ 1))

print("Sumofsquaresofnumbersfrom1ton:",b)
```

OUTPUT:

```
Enteranumber:28

Sumofsquaresofnumbersfrom1ton:7714
```

iii) $\sum n$**3**

PROGRAM:

```
n=int(input("Enteranumber:"))

c =sum(i**3foriinrange(1,n+ 1))

print("Sumofcubesofnumbersfrom1ton:",c)
```

OUTPUT:

```
Enteranumber:28

Sumofcubesofnumbersfrom1ton: 164836
```

iv) $\sum n$**!**

PROGRAM:

```
importmath

n=int(input("Enteranumber:"))

d=sum(math.factorial(i)foriinrange(1,n+1))

print("Sumoffactorialsofnumbersfrom1ton:",d)
```

OUTPUT:

```
Enteranumber:28

Sumoffactorialsofnumbersfrom1ton:316196664211373618851684940313
```

v) 1×2+2×3+3×4+⋯+(n−1)×n

PROGRAM:

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

z=sum(i*(i+1)foriinrange(1,n))

print("Sumofproducts(i*(i+1))from1ton:", z)
```

OUTPUT:

```
Enteranumber:28

Sumofproducts(i*(i+1))from1ton:7308
```

5. Write a program to print the sum of all number between a and b.

PROGRAM:

```
a=int(input("Enterthefirstnumber:"))

b=int(input("Enterthefirstnumber:")) sum

= 0

forvalueinrange(a,b+1):

        sum=sum+ value

print("THESUMOFALLNO.SBETWEEN”,a,”AND”,b,“is",sum)
```

OUTPUT:

```
Enterthefirstnumber:-3

Enterthe firstnumber:3

THESUMOFALLNO.SBETWEEN-3AND3is0
```

6. Write a program to find the sum of square of all the number which are divisible by 5 between a and b.

PROGRAM:

```
a=int(input("Enterthenumber:"))

b=int(input("Enterthenumber:")) sum

= 0

fornuminrange(a,b+1): if

        (num % 5 == 0):

                sum=sum+num**2

print("SUM = ",sum)
```

OUTPUT:

```
Enter the number : 2

Enterthenumber:500

SUM =8458750
```

7. Write a program to read a number and state whether it is an Arm strong number or not.

PROGRAM:

```
num=int(input("Enteranumber:"))

sum = 0

a = num

whilea>0:
   digit = a % 10

   cube = digit**3

   sum=sum+cube

a=a//10

ifsum==num:

   print("ItisanArmstrongnumber.")

else:

   print("ItisnotaArmstrongnumber.")
```

OUTPUT:

```
Enteranumber:200

ItisnotaArmstrongnumber.
```

8. Write a program to read a number and state whether it is a Palindrome number or not.

PROGRAM:

```
n=int(input("Enteranumber:")) m

= n

r=0

while(n!=0):

         rem=n%10 print(rem)

         r=r*10+rem

         n=(int)(n/10)


         print("reverseis",r)

if (m == r):

         print("It'saPalindromenumber")

else:

          print("It'snotaPalindromenumber")
```

OUTPUT:

```
Enteranumber:1211

reverseis1
```

```
2

reverseis12

1

reverseis121

It'saPalindromenumber
```

PRACTICAL:4

Write a python program to display the following patterns:

```
i)    *
      **
      ***
      ****
      *****
```

PROGRAM:

```
n=int(input("Enterthenumberoflines:")) for i

in range (1 , n + 1):

  forjinrange(1,i+1):

    print("*",end="")

    i = i+1

  print()
```

OUTPUT:

```
Enterthenumberoflines:5

   *

   **

   ***

   ****

   *****
```

```
ii.)  #####

       ####

       ###

       ##

        #
```

PROGRAM:

```
n=int(input("Enterthenumberoflines: "))
```

```
foriin range(n,0,-1):
   forjinrange(1,i+1):
      print("#",end="")
      i = i-1
   print()
```

OUTPUT:

```
Enterthenumberoflines:5
#####
####
####
#
#
```

```
iii.)      1
          01
          101
          0101
          10101
```

PROGRAM:

```
n=int(input("Enterthenumberoflines:")) for i
in range (1,n + 1):
   if(i %2==0):
      c=0
   else:
      c=1
   forjinrange(1,i+1):
      print(c,end="")
      if(c== 0):
         c=1
      else:
         c = 0
   print("\n")
```

OUTPUT:

```
Enterthenumberoflines:5
1
01
101
0101
10101
```

iv.) 1

22

333

4444

55555

PROGRAM:

```
n=int(input("Enterthenumberoflines:")) for i
in range (1,n + 1):
        forjinrange(1,i+1):
            print(i,end="")
        i=i+1
        print()
```

OUTPUT:

```
Enterthenumberoflines:5 1
22
333
4444
55555
```

v.)54321

4321

321

21

1

PROGRAM:

```
n=int(input("Enterthenumberoflines:"))
```

```
foriinrange(5,0,-1):
        forjinrange(i,0,-1):
                print(j,end="")
        print()
```

OUTPUT:

```
Enterthenumberoflines:5
54321
4321
321
21
1
```

vi.)1

```
    23
    456
    78910
```

PROGRAM:

```
a=int(input("Enterthenumberoflines:")) num
= 1
for i in range(1, a+1):
   forjinrange(1,i+1):
      print(num,end="")
      num+=1
   print()
```

OUTPUT:

```
Enterthenumberoflines:4 1
23
456
78910
```

PRACTICAL:5

1. **Given the string:text='Hello, Python is awesome and versatile!',write a Python program to perform the following operations on the given string:**

   i) **Display the String: Show the content of the string text.**

   ii) **String Length: Display the length of the string.**

   iii) **Uppercase and Lowercase: Display the string in uppercase and lowercase formats.**

   iv) **Reverse the String: Display the string in reverse order.**

   v) **Extract a Substring: Extract and display a substring that includes only the word 'Python'.**

   vi) **Replace Text: Replace the word 'awesome' with 'great' in the string.**

   vii) **Strip White Spaces: Remove any leading or trailing white spaces from the string.**

PROGRAM:

```
text='Hello,Pythonisawesomeandversatile!'


#Display the String

print("StringContent:")

print(text)


#String Length

print("\nLengthoftheString:")

print(f"Thelengthofthestringis:{len(text)}")


#Uppercase and Lowercase

print("\n UppercaseandLowercase:")

print(f"Uppercase: {text.upper()}")

print(f"Lowercase:{text.lower()}")


#Reverse the String

print("\nReversetheString:")

print(f"Reversed string: {text[::-1]}")

# Extract a Substring with 'Python'

print("\n Substring with 'Python':")
```

```
substring = text[text.find('Python'):text.find('Python') + len('Python')]

print(f"Substring containing 'Python': {substring}")


#Replace Text

print("\nReplace'awesome'with'great':")

new_text  =text.replace('awesome',  'great')

print(new_text)


#Strip White Spaces

print("\nStripWhiteSpaces:")

stripped_text = text.strip()

print(f"Stringafterstrippingleadingandtrailingwhitespaces:'{stripped_text}'")
```

OUTPUT:

```
String Content:

Hello, Python is awesome and versatile!


 Length of the String:

The length of the string is: 39


 Uppercase and Lowercase:

Uppercase: HELLO, PYTHON IS AWESOME AND VERSATILE!

Lowercase: hello, python is awesome and versatile!


 Reverse the String:

Reversed string:!elitasrev dna emosewa si nohtyP ,olleH


 Substring with 'Python':

Substring containing 'Python': Python


 Replace 'awesome' with 'great':

Hello, Python is great and versatile!


 Strip White Spaces:
```

String after stripping leading and trailing white spaces: 'Hello, Python is awesome and versatile!'

2. Write a program to read username and display whether the entered name is valid or not.

PROGRAM:

```
importre

defis_valid_username(username):

    #Checkthelengthoftheusername if

    len(username) < 6:

        returnFalse

    #Checkifitcontainsonlylettersandnumbers if

    not re.match("[a-zA-Z0-9]", username):

        returnFalse

    #Checkifitdoesn'tstartwithanumber

    ifusername[0].isdigit():

        returnFalse

    #Ifallconditionsaremet,it'savalidusername

    return True

#Getinputfromtheuser

username=input("Enterausername:")

#Checkifit'svalidanddisplaytheresult     if

is_valid_username(username):

    print(f"Theusername'{username}'isvalid.")

else:

    print(f"Theusername'{username}'isnotvalid."
```

OUTPUT:

```
Enterausername:sahil_07

Theusername'sahil_07'isvalid.
```

3. **Write a program to read mobile number and display whether the entered name is valid or not.**

PROGRAM:

```
num=input("Enterthemobilenumber:") flag =

True

if(len(num)!= 10):

  flag=False

else:

  foriinnum:

    if(noti.isdigit()):

      flag=False

      break

if(flag==True):

  print("Themobilenumberisvalid.")

else:

  print("Themobilenumberisnotvalid.")
```

OUTPUT:

```
Enterthemobilenumber:8976693662 The

mobile number is valid.
```

4. **Write a program to read email address and display whether the entered name is valid or not.**

PROGRAM:

```
importre
def is_valid_email(email):
  if"@"notinemail:
    returnFalse
  if"."notinemail.split("@")[1]:
    returnFalse
  if email.startswith(".") or email.endswith(".") or email.startswith("@") or email.endswith("@"):
    return False
  if not re.match(r"[^@]+@[^@]+\.[^@]+", email):
    return False
  returnTrue
email=input("Enteranemailaddress:") if
is_valid_email(email):
  print(f"Theemailaddress'{email}'isvalid.") else:
  print(f"Theemailaddress'{email}'isnotvalid.")
```

OUTPUT:

```
Enter an email address:sahil07rathore07@gmail.com
Theemailaddress'sahil07rathore07@gmail.com'isvalid.
```

PRACTICAL:6

Create a Python list to represent Indian Railway train names with at least 10 train names as its elements. Write code to perform the following operations directly on this list:

i) Display All Train Names: Print all the train names in the list.

ii) Add a New Train: Append a new train name to the list.

iii) Remove a Train: Delete a specific train name from the list.

iv) Search for a Train: Prompt the user to input a train name and check if it exists in the list. Display an appropriate message based on the existence of the train.

v) Sort Train Names: Sort the train names in alphabetical order.

vi) Display Train Count: Print the total count of train names in the list.

vii) Print First Three Train Names: Display the first three train names in the list.

viii) Print Last Three Train Names: Show the last three train names in the list.

ix) Print Train Names Reversed: Display all train names in reverse order.

x) Access and Display a Range of Train Names: Slice the list to display train names from the 4th to the 7th position.

xi) Clear the List: Remove all elements from the list.

PROGRAM:

```
#Create a list of Indian Railway train names

indian_trains = [

  "Rajdhani Express",

  "Shatabdi Express",

  "Duronto Express",

  "Garib Rath",

  "HumsafarExpress",

  "GatimaanExpress",

  "Tejas Express",

  "VandeBharatExpress",

  "Jan Shatabdi Express",
    "DoubleDeckerExpress"
]
#DisplayAllTrainNames
print("AllTrainNames:")
fortrain inindian_trains:
    print(train)


# Add a New Train
indian_trains.append("Ayodhya Express")
print("\nNew Train Added:", indian_trains)


# Remove a Train
train_to_remove="GaribRath"
if train_to_remove in indian_trains:
    indian_trains.remove(train_to_remove)
    print("\nTrain Removed:", indian_trains)
else:
    print(f"\nTrain'{train_to_remove}'notfoundinthelist.")


#Search for aTrain
search_train=input("\niv)Enteratrainnametosearch:") if
search_train in indian_trains:
    print(f"Train'{search_train}'existsinthelist.")
else:
    print(f"Train'{search_train}'doesnotexistinthelist."
#Sort Train Names
indian_trains.sort()
print("\nTrainNamesSorted:",indian_trains)

#Display Train Count
print("\nTotalTrainCount:",len(indian_trains))

#Print First Three Train Names
print("\nFirstThreeTrainNames:",indian_trains[:3])

#Print Last Three Train Names
print("\nLastThreeTrainNames:",indian_trains[-3:])

#Print Train Names Reversed
print("\nTrainNamesReversed:",indian_trains[::-1])

#Access and Display a Range of Train Names
print("\nTrainNamesfrom4thto7thposition:",indian_trains[3:7])

# Clear the List
indian_trains.clear()
print("\nListCleared.TrainNames:",indian_trains)
```

OUTPUT:

```
AllTrainNames:
RajdhaniExpress
```

```
ShatabdiExpress

Duronto Express

GaribRath

HumsafarExpress

GatimaanExpress

TejasExpress

Vande Bharat Express

Jan Shatabdi Express

DoubleDeckerExpress


NewTrainAdded:['RajdhaniExpress','ShatabdiExpress','DurontoExpress','GaribRath','HumsafarExpress',
'Gatimaan Express', 'Tejas Express','VandeBharatExpress','JanShatabdiExpress','DoubleDeckerExpress',
'AyodhyaExpress']


TrainRemoved:['RajdhaniExpress','ShatabdiExpress','DurontoExpress','HumsafarExpress','Gatimaan
Express', 'Tejas Express', 'Vande Bharat Express', 'Jan Shatabdi Express', 'Double Decker Express', 'Ayodhya
Express']


iv) Enteratrainnametosearch:VandeBharatExpress

Train 'Vande Bharat Express' exists in the list.


TrainNames Sorted:['Ayodhya Express', 'Double Decker Express', 'Duronto Express', 'Gatimaan Express',
'HumsafarExpress','JanShatabdiExpress','RajdhaniExpress','ShatabdiExpress','TejasExpress','Vande
Bharat Express']


TotalTrainCount:10


FirstThreeTrainNames:['AyodhyaExpress','DoubleDeckerExpress','DurontoExpress']


LastThreeTrainNames:['ShatabdiExpress', 'TejasExpress','VandeBharatExpress']


TrainNames Reversed:['Vande Bharat Express', 'Tejas Express', 'Shatabdi Express', 'Rajdhani Express', 'Jan
ShatabdiExpress','HumsafarExpress', 'Gatimaan Express','Duronto Express','Double Decker Express',
'AyodhyaExpress']
```

```
TrainNamesfrom4thto7thposition:['GatimaanExpress','HumsafarExpress','JanShatabdiExpress', 'Rajdhani
Express']


ListCleared.TrainNames:[]
```

PRACTICAL:7

Create a Python tuple representing names of at least 10 cities, in a tuple. Write code to perform the following operations directly on this tuple:

i) Display All City Information: Print names of all the cities in the tuple.

ii) Add New City Information: Add a new city to the tuple.

iii) Remove City Information: Remove a specific city from the tuple.

iv) Search for City Information: Prompt the user to input a city name and check if its details exist in the tuple.

v) Sort City Information: Sort the city details based on its name.

vi) Display City Count: Print the total count of cities in the tuple.

vii) Print First Three City Information: Display the names of the first three cities in the tuple.

viii) Print Last Three City Information: Show the names of the last three cities in the tuple.

ix) Print City Information Reversed: Display all city names in reverse order.

x) Access and Display a Range of City Information: Slice the tuple to display names from the 4th to the 7th position.

xi) Find Index of a City: Find the index of a specific city's name in the tuple.

PROGRAM:

```
#Create a tuple with names of cities

city_names = (

   "Mumbai", "Delhi", "Bengaluru", "Hyderabad", "Chennai",

   "Kolkata", "Ahmedabad", "Pune", "Jaipur", "Lucknow"

)

#Display All City Information

print("AllCityInformation:")

for city in city_names:

   print(city)


#Add New City Information

new_city = "Chandigarh"

city_names += (new_city,)

print("\nNewCityAdded:",city_names)
```

```
#Remove City Information
city_to_remove = "Pune"
ifcity_to_removeincity_names:
    city_list=list(city_names)
    city_list.remove(city_to_remove)
    city_names = tuple(city_list)
    print("\n City Removed:", city_names)
else:
    print(f"\nCity'{city_to_remove}'notfoundinthetuple.")

#Search for City Information
search_city=input("\nEnteracitynametosearch:") if
search_city in city_names:
    print(f"City'{search_city}'existsinthelist.")
else:
    print(f"City'{search_city}'doesnotexistinthelist.")

#Sort City Information
sorted_cities=sorted(city_names)
print("\nCityInformationSorted:",sorted_cities)

#Display City Count
print("\nTotal City Count:",len(city_names))

#Print First Three City Information
print("\nFirstThreeCityInformation:",city_names[:3])

#Print Last Three City Information
print("\nLastThreeCityInformation:",city_names[-3:])

#Print City Information Reversed
print("\nCityInformationReversed:",city_names[::-1])
```

```
#Access and Display a Range of City Information
print("\nCityInformationfrom4thto7thposition:",city_names[3:7])


#Find Index of a City
search_city_index=input("\nEnteracitynametofinditsindex:") if
search_city_index in city_names:
    index=city_names.index(search_city_index)
    print(f"Index of '{search_city_index}' in the tuple: {index}")
else:
    print(f"City'{search_city_index}'notfoundinthetuple.")
```

OUTPUT:

```
AllCityInformation:
Mumbai
Delhi
Bengaluru
Hyderabad
Chennai
Kolkata
Ahmedabad
Pune
Jaipur
Lucknow

NewCityAdded:('Mumbai','Delhi','Bengaluru','Hyderabad','Chennai','Kolkata','Ahmedabad','Pune',
'Jaipur', 'Lucknow', 'Chandigarh')

 City Removed: ('Mumbai', 'Delhi', 'Bengaluru', 'Hyderabad', 'Chennai', 'Kolkata', 'Ahmedabad', 'Jaipur',
'Lucknow','Chandigarh')

 Enteracitynametosearch:Mumbai
City 'Mumbai' exists in the list.

 City Information Sorted: ['Ahmedabad', 'Bengaluru', 'Chandigarh', 'Chennai', 'Delhi', 'Hyderabad', 'Jaipur',
'Kolkata', 'Lucknow', 'Mumbai']
```

```
Total CityCount: 10

FirstThreeCityInformation:('Mumbai','Delhi','Bengaluru')

LastThreeCityInformation:('Jaipur','Lucknow','Chandigarh')

City Information Reversed: ('Chandigarh', 'Lucknow', 'Jaipur', 'Ahmedabad', 'Kolkata', 'Chennai',
'Hyderabad', 'Bengaluru', 'Delhi', 'Mumbai')

CityInformationfrom4th to7thposition:('Hyderabad','Chennai','Kolkata','Ahmedabad')

Enteracitynameofinditsindex:Lucknow Index
of 'Lucknow' in the tuple: 8
```

PRACTICAL:8

Create a Python dictionary representing the capitals of various countries, with a minimum of 10 country-capital pairs. Write code to perform the following operations directly on this dictionary:

i) Display All Country-Capital Pairs: Print all country-capital pairs in the dictionary.

ii) Add New Country-Capital Pair: Add information about a new country- capital pair to the dictionary.

iii) Remove Country-Capital Pair: Remove a specific country's capital from the dictionary.

iv) Search for Country-Capital Pair: Prompt the user to input a country name and check if its capital exists in the dictionary. Display the capital if the country is found.

v) Update Country-Capital Pair: Update the capital of a specific country within the dictionary.

vi) Display Country Count: Print the total count of country-capital pairs in the dictionary.

vii) Access and Print Specific Country-Capital Pair: Display the capital of a specific country from the dictionary.

viii) Access and Display a List of Capitals: Extract and print a list of all capitals from the dictionary.

ix) Extract Country Names: Extract and display all country names from the dictionary.

x) Display Key-Value Pairs: Display all key-value pairs of country-capital pairs in the dictionary.

xi) Check for Key Existence: Verify if a particular country exists as a key in the dictionary.

xii) Remove Specific Country-Capital Pair: Remove a specific country-capital pair from the dictionary.

PROGRAM:

```
#Update Country-Capital Pair

update_country = "USA"
```

```
new_capital="Washington"
ifupdate_countryincountry_capitals:
    country_capitals[update_country]=new_capital
    print("\n Country-Capital Pair Updated:", country_capitals)
else:
    print(f"\nCountry'{update_country}'notfoundinthedictionary.")

#Display Country Count
print("\nTotalCountryCount:",len(country_capitals))

#Access and Print Specific Country-Capital Pair
specific_country = "India"
ifspecific_countryincountry_capitals:
    print(f"\nCapitalof'{specific_country}'is'{country_capitals[specific_country]}'")

# Access and Display a List of Capitals
capitals_list = list(country_capitals.values())
print("\n List of Capitals:", capitals_list)

#Extract Country Names
country_names=list(country_capitals.keys())
print("\nListofCountryNames:",country_names)

# Display Key-Value Pairs
print("\nKey-ValuePairs:")
forkey,valueincountry_capitals.items():
    print(f"{key}-{value}")

#Check for Key Existence
search_key = "Japan"
ifsearch_keyincountry_capitals:
    print(f"\n '{search_key}' exists as a key in thedictionary.")
else:
    print(f"\n'{search_key}'doesnotexistasakeyinthedictionary.")


#Remove Specific Country-Capital Pair

specific_country_remove="Germany"

ifspecific_country_removeincountry_capitals:

    delcountry_capitals[specific_country_remove]

    print(f"\nSpecific Country-Capital Pair '{specific_country_remove}' Removed:", country_capitals)

else:

    print(f"\nSpecificCountry-CapitalPair'{specific_country_remove}'notfoundinthedictionary.")
```

OUTPUT:

```
AllCountry-CapitalPairs:

India-NewDelhi

USA-WashingtonD.C. UK

- London

France - Paris

Germany-Berlin

Japan - Tokyo

Australia-Canberra

Canada - Ottawa

Brazil - Brasília

SouthAfrica-Pretoria


 NewCountry-CapitalPairAdded:{'India':'NewDelhi','USA':'WashingtonD.C.','UK':'London','France':
'Paris','Germany':'Berlin','Japan':'Tokyo','Australia':'Canberra','Canada':'Ottawa','Brazil':'Brasília', 'South
Africa': 'Pretoria', 'Russia': 'Moscow'}


 Country-Capital Pair Removed: {'India': 'New Delhi', 'USA': 'WashingtonD.C.', 'UK': 'London', 'France': 'Paris',
'Germany':'Berlin','Japan':'Tokyo','Australia':'Canberra','Canada':'Ottawa','South Africa':'Pretoria', 'Russia':
'Moscow'}


)Enteracountrynametosearch:India The

capital of 'India' is 'New Delhi'
```

```
 Country-Capital Pair Updated: {'India': 'New Delhi', 'USA': 'Washington', 'UK': 'London', 'France': 'Paris',
'Germany':'Berlin','Japan':'Tokyo','Australia':'Canberra','Canada':'Ottawa','SouthAfrica':'Pretoria',
'Russia': 'Moscow'}


 Total CountryCount:10


 Capitalof'India'is'NewDelhi'


 List of Capitals: ['New Delhi', 'Washington', 'London', 'Paris', 'Berlin', 'Tokyo', 'Canberra', 'Ottawa', 'Pretoria',
'Moscow']


 ListofCountryNames:['India','USA','UK','France','Germany','Japan','Australia','Canada','SouthAfrica',
'Russia']


 Key-ValuePairs:
India - New Delhi
USA-Washington
UK - London
France - Paris
Germany-Berlin
Japan - Tokyo
Australia - Canberra
Canada - Ottawa
SouthAfrica-Pretoria
Russia - Moscow


 'Japan'existsasakeyinthedictionary.


SpecificCountry-CapitalPair'Germany'Removed:{'India':'NewDelhi','USA':'Washington','UK':'London',
'France': 'Paris', 'Japan': 'Tokyo', 'Australia': 'Canberra', 'Canada': 'Ottawa', 'South Africa': 'Pretoria', 'Russia':
'Moscow'}
```

PRACTICAL:9

1. Given the array: arr=array('i',[3,7,1,14,5,8]), write a Python program using the array module to perform the following operations on the given array:

- **i) Display the Array: Show the content of the array arr.**
- **ii) Reverse the Array: Reverse the order of elements in the array.**
- **iii) Extract a Slice: Display a slice of the array containing elements from the 2nd to the 4th position.**
- **iv) Search for Element: Search for the element '14' in the array and display its index if found, else indicate if it's not present.**
- **v) Add an Element: Append the number '10' to the end of the array.**
- **vi) Remove an Element: Remove the element '5' from the array.**
- **vii) Sort the Array: Sort the elements of the array in ascending order.**
- **viii) Clear the Array: Remove all elements from the array.**

PROGRAM:

```
From array import array
# Create the array
arr=array('i',[3,7,1,14,5, 8])


#Display the Array
print("ArrayContent:",arr)


#Reverse the Array
arr.reverse()
print("\nReversedArray:", arr)


#Extract a Slice
slice_array=arr[1:4]
print("\nSliceoftheArray(2ndto4thposition):",slice_array)


#Search for Element
search_element=14
ifsearch_elementinarr:
   index=arr.index(search_element)
   print(f"\nElement'{search_element}'foundatindex:{index}")
```

```
else:
    print(f"\nElement'{search_element}'notfoundinthearray.")

#Add an Element
arr.append(10)
print("\nArrayafterappending'10':",arr)

#Remove an Element
element_to_remove=5
if element_to_remove in arr:
    arr.remove(element_to_remove)
    print("\n Arrayafterremoving'5':",arr) else:
    print(f"\nElement'{element_to_remove}'notfoundinthearray.")

#Sort the Array
arr=array('i',[3,7,1,14,5,8])#Restoringtheoriginalarray arr =
array('i', sorted(arr))
print("\nSortedArray:",arr)

#Clear the Array
arr=array('i',[3,7,1,14,5,8])#Restoringtheoriginalarray arr =
array('i')# Clearing the array
print("\nArrayCleared:",arr)
```

OUTPUT:

```
ArrayContent:array('i',[3,7,1,14,5,8])

 ReversedArray:array('i',[8,5,14,1,7,3])

 SliceoftheArray(2ndto4thposition):array('i',[5,14,1])

 Element'14'foundatindex:2
```

```
Arrayafterappending'10':array('i',[8,5,14,1,7,3,10])


Arrayafterremoving'5':array('i',[8,14,1,7,3,10])


SortedArray:array('i',[1,3,5,7,8,14])


Array Cleared:array('i')
```

2. Given the array: arr=array('i',[3,7,1,14,5,8]), write a Python program using the array module to:

- **i)** **Display the elements of the array**
- **ii)** **Find the sum of the elements of the array**
- **iii)** **Find the maximum and minimum elements in the array.**

PROGRAM:

```
From array import array


#Create the array

arr=array('i',[3,7,1,14,5, 8])


#Display the elements of the array

print("ElementsoftheArray:",list(arr))


#Find the sum of the elements of the array

array_sum = sum(arr)

print("\nSumofArrayElements:",array_sum)


#Find the maximum and minimum elements in the array

max_element = max(arr)

min_element=min(arr)

print("\n Maximum Element:", max_element)

print("\nMinimumElement:",min_element)
```

OUTPUT:

```
ElementsoftheArray:[3,7,1,14,5,8]


 SumofArrayElements:38


 MaximumElement:14


 MinimumElement: 1
```

PRACTICAL:10

1. **Define a function sum such that it accept any number of parameters and return their sum.**

PROGRAM:

```
defsum(*args):

   total= 0

   for num in args:

      total+=num

   return total


#Calling the function with 3 arguments

result = sum(5, 10, 15)

print("Sum:",result)


#Callingthefunctionwith2arguments

result = sum(5, 10)

print("Sum:",result)
```

OUTPUT:

```
Sum:30

Sum:15
```

2. **Write a program to create an Anonymous function such that it accept three parameters and return their sum.**

PROGRAM:

```
#Creating an anonymous function to sum three parameters

sum_three = lambda a, b, c: a + b + c


#Calling the anonymous function

result = sum_three(5, 10, 15)

print("Sum:",result)
```

OUTPUT:

```
Sum:30
```

3. **Define a function such that it accepts principal, rate of interest and period and returns the simple interest.**

PROGRAM:

```
defcalculate_simple_interest(principal,rate,time):
   simple_interest=(principal*rate*time)/100
   return simple_interest


principal_amount=1000#Exampleprincipalamount
interest_rate = 5      # Example rate of interest
time_period = 2        # Example time period


# Calling the function
result = calculate_simple_interest(principal_amount, interest_rate, time_period)
print("Simple Interest:", result)
```

OUTPUT:

```
SimpleInterest:100.0
```

4. **Create a Python program using the math module to demonstrate the following functionalities:**

- i) **Calculate Exponential Function: Calculate and display the value of eraised to the power of x (e^x), where x = 2.5.**
- ii) **Calculate Logarithmic Values: Find the logarithmic values of a number in different bases, for instance, log base10 and log base2 of a number (e.g., num = 100).**
- iii) **Calculate Trigonometric Function: Compute and display the sine/cosine/ tangent value of an angle (in radians, e.g., angle = math.pi/4).**
- iv) **Round a Number to the Nearest Integer: Round a floating-point number (e.g., val = 4.75) to the nearest integer.**
- v) **Round a floating-point number(e.g., val=3.25) up and down to the nearest integer**
- vi) **Calculate Absolute Value: Find the absolute value of a number (e.g., num=-8).**
- vii) **Calculate Power and Square Root: Calculate the power of a number (e.g., base=3, exponent=4) and find the square root of a number (e.g., num= 81).**

PROGRAM:

```
Import math

#Calculate Exponential Function (e^x)
x = 2.5
exp_result=math.exp(x)
print(f"ExponentialFunction(e^{x}):{exp_result}")

#Calculate Logarithmic Values
num = 100
log_base_10 =math.log10(num)
log_base_2 = math.log2(num)
print(f"\n Logarithmic Values of {num}:")
print(f"Logbase10of{num}:{log_base_10}")
print(f"Log base 2 of {num}: {log_base_2}")

#Calculate Trigonometric Function
angle = math.pi / 4
sin_value = math.sin(angle)
cos_value=math.cos(angle)
tan_value=math.tan(angle)
print(f"\nTrigonometricFunctions(Angle=π/4radians):")
print(f"Sine: {sin_value}")
print(f"Cosine: {cos_value}")
print(f"Tangent: {tan_value}")

#Round a Number to the Nearest Integer
val = 4.75
rounded_val=round(val)
print(f"\nRoundtoNearestInteger(4.75):{rounded_val}")

#Round a floating-point number up and down to the nearest integer
val_up = 3.25
val_down=3.25
rounded_val_up=math.ceil(val_up)
```

```
rounded_val_down=math.floor(val_down)
print(f"\nRoundUpandDowntoNearestInteger(3.25):")
print(f"Rounded up: {rounded_val_up}")
print(f"Roundeddown:{rounded_val_down}")

#Calculate Absolute Value
num = -8
abs_value=abs(num)
print(f"\nAbsoluteValueof{num}:{abs_value}")

#Calculate Power and Square Root
base = 3
exponent=4
power_result = math.pow(base, exponent)
sqrt_result = math.sqrt(81)
print(f"\nPowerandSquareRoot:")
print(f"Powerof{base}raisedto{exponent}:{power_result}")
print(f"Square Root of 81: {sqrt_result}")
```

OUTPUT:

```
ExponentialFunction(e^2.5):12.182493960703473

 LogarithmicValuesof100:
Logbase10of100:2.0
Logbase2of100:6.643856189774724

 TrigonometricFunctions(Angle=π/4radians):
Sine: 0.7071067811865476
Cosine:0.7071067811865476
Tangent:0.9999999999999999

 RoundtoNearestInteger(4.75):5

 RoundUpandDowntoNearestInteger(3.25):
Rounded up: 4
```

```
Roundeddown:3


 AbsoluteValueof-8:8


 Power and Square Root:
Powerof3raisedto4:81.0
SquareRootof81:9.0
```