Practical - 1

Q) Write a program to design and develop a python program to implement various Control Statement using Suitable Example

A) Write a program to Check whether an alphabet is a Vowel or Consonant

```
letter = Input ("Input a letter of the alphabet")
    if letter in ('a', 'e', 'i', 'o', 'u'):
        print ("%s is a vowel" % letter)
elif Letter == 'y':
    print ("Sometimes letter y stands for Vowel and
        Sometimes stands for Consonant")
else:
    print ("%s is Consonant" % letter)
```

Output :-

```
Input a letter of the alphabet k
k is Consonant.
```

B) Write a python program to find those number which are divisble by 7 and multiple of 5 b/w 1500 and 2700

```
n=[]
for i in range (1500,2700):
    if (i%7==0) and (i%5==0):
        n.append (str(i))
print (',' .join (n))
```

Output:

1505, 1540, 1575, 1610, 1645, 1680, 1715, 1750, 1785, 1820, 1855, 1890, 1925, 1960, 1995, 2030, 2065, 2100, 2135, 2170, 2205, 2240, 2275, 2310, 2345, 2380, 2415, 2450, 2485, 2520, 2555, 2590, 2625, 2660, 2695.

practical-2

A) Write a program in python to define and call fonction for suitable problem

```
def outer_fun (a,b):
        Square = a**2
print ("Square is ", Square)
    def addition (a,b):
        return add+5
result = outer_fun (5,10)
print ("the result is: \n", result)
```

Output:-

Square is 25
The Result is :
20

Practical - 3

A) Write a python program of different types of function argument

```
# function defination
def information (Name, Age):
    print ("Name", Name)
    print ("Age", Age)
    return

# Calling out the function
information (Age = 32, Name = "Atul")
```

Output

```
Name Atul
Age 32
```

DATE:

3.b default argument

```
# function defination
def information (name, age =32):
    print ("Name", name)
    print ("Age", age)
    return

# Call function 'information
'information (age = 35, name = "Gayatri")
information (name = "Atul")
```

Output :

```
Name Gayatri
Age 35
Name Atul
Age 32
```

3.c Arbitary Argument

```
def greet (*names):
    for name in names:
        print ("Good Morning", name)

greet ("Atharv", "Daniel", "Yash", "Aayush")
```

Output

```
Good Morning Atharv
Good Morning Daniel
Good Morning Yash
Good Morning Aayush
```

Page
Date: / /

practical - 4

Aim: Write a python programme to demonstrate the presidence and associativity of operators.

4.A Demonstrate Operator presidenee

```
x = 10
y = 5
c = x * y * (x+y)
    print(c)
```

output : 750

4:B Demonstrate left-right associativity

```
print (8 * 2//3)
print (8 * (2//3))
```

output : 5
0

4:c Demonstrate right - left associativity

```
print ("2**4 * *2)
print ((2**4)**2)
```

output : 65536
256

practical - 5

Date: / /

Write Suitable python programme to implement recursion for problems such as factorial Fibonacci Series

5.A Factorial

Input :

```
def recur_factorial(n):
    if n==1:
        return n
    else:
        return n*recur_factorial(n-1)
num = int(input("Enter the No. to find factorial"))
if num<0:
    print("Sorry, factorial does not exist for negative No.")
elif num==0:
    print("Factorial of 0 is 1")
else:
    print("Factorial of ", num, "is", recur_factorial(num))
```

Output :

```
Enter the No. to find factorial 5
Factorial of 5 is 120
```

Page
Date: / /

S.B Fibonacci Series.

```
def recur_fibo(n):
        if n<=1:
        return n
    else:
    return (recur_fibo(n-1) + recur_fibo(n-2))
n = int(input("Enter the number:"))
 if n<=0
   print("please enter a positive number.")
  else:
   print("Fibonacci Sequence")
   for i in range(n):
        print(recur_fibo(i))
```

Output:

```
Enter the Number: 6
 Fibonacci Sequence
0
,
1
2
3
5
```

Practical-6

Date: / /

Write a python programme to implement and use lambda function in python

6.A programme for Square using lambda function

Input :

```
Square = lambda a : a * a
num = int (input ("Enter the number"))
result = square (num)
print ("square of number is", result)
```

Output :

```
Enter the number 3
Square of Number is 9
```

Date: / /

G.B Programme to find Square, Cube and Square root using lambda function.

```
import math
    def myfun(n):
        return lambda a: math.pow(a,n)
Square = myfun(2)
Cube = myfun(3)
Square root = myfun(0.5)
num = int(input("Enter the Number"))
print(Cube(num))
print(Square root(num))
```

output:

```
Enter the Number 9
 81.0
 729.0
 3.0
```

Date: / /

practical-7

Write a python program to Create and manipluate array in python also demonstate the use of Silicing and indexing for accessing elements from the array

```
Import numpy as np
x=np.reshape (np.arange (16), (4,4))
print ("original arrays")
print (x)
print ("Sliced element")
result = x[[0,1,2], [0,1,3]]
print (result)
```

Output

Original arrays

```
[[ 0 1 2  3]
 [ 4 5 6  7]
 [ 8 9 10 11]
 [12 13 14 15]
 Sliced element
[0 5 11]
```

Date: / /

practical - 8

(a) Write a program to implement list in python for suitable program demostrate various operation on it

```
list1=['red','green','blue','black','white']
list2= [14,12,13]
print("Elements of list1", list1)
print("Elements of list2", list2)
print("The length of list1 is", len(list1))
list1.insert(2,'pink')
print("list after adding element :", list1)
list1.append('yellow')
print("list after append :", list1)
list2.sort()
print("Sorted list2 :", list2)
list1.remove('blue')
print("list after removing element :", list1)
print("print from begining to end index", list1[:4])
print("print from begining to end index", list1[2:4])
list1.reverse()
print("reverse list is :", list1)
list1.extend(list2)
print("print after Extend operation :", list1)
```

output

Element of list1 ['red', 'green', 'blue', 'black', 'white']
Element of list2: [14, 12, 13]
Length of list1 is 5
length after adding elements: ['red', 'green', 'pink', 'blue' 'black', 'white']
list1 after append: ['red', 'green', 'pink', 'blue', 'black' 'white', 'yellow']
Sorted list2: [12, 13, 14]
List1 after removing elements: ['red', 'green', 'pink' 'black']
print from begining to end index: ['red', 'green', 'pink', 'black']
print from begining to end index: ['pink', 'black']
Reverse list1 is: ['yellow', 'white', 'black', 'pink', 'green', 'red']
print after extend operation: ['yellow', 'white', 'black', 'pink', 'green', 'red', 12, 13, 14]

Page
Date: / /

practical - 9

Q) write a program to implement tuple in python demostrate various operation on it

```
t= (1,2,3,4,5)
print("The value in tuple are", t)
  t = t + (7)
print("The value Tuple after adding elements : ", t)
print("Tuple after slice : ", t[2:4])
print(" multiplication : ", (t*2))
print("The length of tuple is : ", len(t))
print("The maximum value of tuple is : ", max(t))
print("The minimum value of tuple is : ", min(t))
 del t
 print ("Tuple deleted")
```

Output

```
The Value in Tuple are (1,2,3,4,5)
Tuple after adding elements : (1,2,3,4,5,7
Tuple after Slice : 3,4
Multiplication : (1,2,3,4,5,7,1,2,3,4,5,7)
The length of tuple is : 6
The maximum value of tuple is : 7
The minimum value of tuple is : 1
Tuple deleted
```

Date: / /

practical - 10

Write a program to implement dictionary in python demostrate various operations on it.

```
my dictionary = {
                'a': '65'
                'b' : '66'
                'c' : '67'
               }
print (type (my dictionary))
print (my dictionary)
my dictionary ['d'] = '68'
my dictionary ['e'] = '69'
my dictionary ['f'] = '70'
print ("Dictionary after adding items :", mydictionary)
length = len (mydictionary)
print ("length of dictionary is :", length)
for key in my dictionary:
    print ("key:", key)
for key, value in my dictionary.items():
    print ("values :", value)
my dictionary.clear()
print (mydictionary)
```

Output

```
<class 'dict'>
{'a': '65', 'b': '66', 'c': '67'}
Dictionary after adding items : {'a': '65', 'b': '66',
'c': '67', 'd': '68', 'e': '69', 'f': '70'}
length of dictionary : 6
keys : a
keys : b
keys : c
keys : d
keys : e
keys : f
Values : 65
Values : 66
Values : 67
Values : 68
Values : 69
Values : 70
{}
```