

COMPUTER PRACTICALS

a) Python Program to Generate all the Divisors of an Integer.

```
n = int(input("enter the integer:"))
```

```
print("The divisor of the number are:")
```

```
for i in range(1, n+1):
```

```
    if (n % i == 0):
```

```
        print(i)
```

output

```
enter the number of integer: 42
```

```
The divisors of the number are:
```

```
1  
2  
3  
6  
7  
14  
21  
42
```

b) Python Program to Find Those Numbers which are Divisible by 7 and Multiple of 5 in a Given Range of Numbers.

```
lower = int(input("enter the lower range:"))
```

```
upper = int(input("enter the upper range:"))
```

```
for i in range(lower, upper + 1):
```

```
    if (i % 7 == 0 and i % 5 == 0):
```

```
        print(i)
```

output

```
enter the lower range: 1
```

```
enter the upper range: 200
```

35

70

105

140

175.

Q) Python Program to Check if a Number is a Strong Number.

8

sum1 = 0

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

temp = num

while (num):

i = 1

f = 1

s = num % 10

while (i <= s):

f = f * i

i = i + 1

sum1 = sum1 + f

num = num // 10

if (sum1 == temp):

print ("the number is a strong number").

else:

print ("the number is not a strong number")

Output

enter a number : 2120

The number is not a strong number.

enter a number : 145

The number is a strong number.

Q5) Python Program to check if Two numbers are Amicable Numbers.

```

x = int(input('Enter number 1 : '))
y = int(input('Enter number 2 : '))
sum1 = 0
sum2 = 0
for i in range(1, x):
    if x % i == 0:
        sum1 += i
for j in range(1, y):
    if y % j == 0:
        sum2 += j
if (sum1 == y and sum2 == x):
    print('Amicable!')
else:
    print('Not Amicable!')
```

Output:

enter number 1 : 220
 enter number 2 : 280
 amicable!

enter number 1 : 210
 enter number 2 : 324
 not amicable!

3a) Python Program to Find the Sum of Sin Series.

```

import math
def sin(x,n):
    sinc = 0
    for i in range(n):
        sign = (-1)**i
        pi = 22/7
        y = x * (pi / 180)
        sinc = sinc + ((y**i * (2.0 * i + 1)) / math.factorial(2 * i + 1)) * sign
    return sinc

```

$x = \text{int}(\text{input}(\text{"Enter the value of } x \text{ in degrees: "}))$
 $n = \text{int}(\text{input}(\text{"Enter the number of terms: "}))$
 $\text{print}(\text{round}(\text{sin}(x, n), 2))$

output

Enter the values of x in degrees: 180

Enter the number of terms: 30

0.52

36. Python Program to Find the sum of Cosine Series

import math

def cosine(x,n):

$\cos x = 1$

$\text{sign} = -1$

for i in range(2, n, 2):

$\pi = 22/7$

$y = x * (\pi / 180)$

$\cos x = \cos x + (\text{sign} * (y**i) / \text{math.factorial}(i))$

(i)

$\text{sign} = -\text{sign}$

return $\cos x$

$x = \text{int}(\text{input}(\text{"Enter the value of } x \text{ in degrees: "}))$

$n = \text{int}(\text{input}(\text{"Enter the value of terms: "}))$

print (round (cosine (x, n), 2))

output

Enter the values of x in degrees : 30

Enter the number of terms : 5

0.87

4a Python Program of Search the Number of Times a Particular Number Occurs in a list.

```
a = []
n = int(input("Enter number of elements :"))
for i in range(1, n+1):
    b = int(input("Enter element :"))
    a.append(b)
```

k = 0
num = int(input("Enter the number to be counted :"))

```
for j in a:
    if (j == num):
        k = k + 1
print("Number of times ", num, " appears is ", k)
```

output

enter number of elements : 4

enter element : 2

enter element : 5

enter element : 6

enter element : 2

enter the number to be counted : 2

number of times 2 appears is 2

Qa. Python Program to Search the Number of Times a Particular Number Occurs in a List.

```
a = []
n = int(input("Enter number of elements:"))
for i in range(1, n+1):
    b = int(input("Enter element:"))
    a.append(b)
K = 0
num = int(input("Enter the number to be counted:"))
for j in a:
    if(j == num):
        K = K + 1
print("Number of times", num, "appears is", K)
```

Output:

```
enter number of element : 4
enter element : 2
enter element : 5
enter element : 6
enter element : 2
enter the number to be counted : 2
number of times 2 appears is 2
```

4b. Python Program to Merge Two List and Sort it

```
a = []
c = []
n1 = int(input("Enter number of elements:"))
for i in range(1, n1+1):
    b = int(input("Enter element:"))
    a.append(b)
n2 = int(input("Enter number of elements:"))
for i in range(1, n2+1):
    d = int(input("Enter element:"))
    c.append(d)
```

```

new = a + c
new.sort()
print("Sorted list is:", new)
output
enter number of elements: 3
enter element: 2
enter element: 5
enter element: 8
enter the number elements: 4
enter element: 1
enter element: 2
enter element: 3
enter element: 4
sorted list is: [1, 2, 2, 3, 4, 5, 8]

```

5a Python Program to Find the Second largest Number
in a list using Bubble sort

```

a = []
n = int(input("Enter number of elements:"))
for i in range(1, n+1):
    b = int(input("Enter element:"))
    a.append(b)
for i in range(0, len(a)):
    for j in range(0, len(a)-i-1):
        if (a[j] > a[j+1]):
            temp = a[j]
            a[j] = a[j+1]
            a[j+1] = temp
print('Second largest number is:', a[n-2])

```

output

```

Enter number of elements: 5
Enter element: 2
Enter element: 5
Enter element: 8
Enter element: 1

```

Enter element : 3 .

Second largest number is : 5 .

Python Program to find the union of two lists .

l1 = []

num1 = int(input('Enter the size of list 1 : '))

for n in range(num1):

 numbers1 = int(input('Enter any number : '))

 l1.append(numbers1).

l2 = []

num2 = int(input('Enter size of list 2 : '))

for n in range(num2):

 numbers2 = int(input('Enter any number : '))

 l2.append(numbers2)

union = list(set(l1).union(l2))

print('The union of two lists is : ', union)

Output :

Enter the size of list 1 : 4

Enter any number : 2 .

Enter any number : 3 .

Enter any number : 6 .

Enter any number : 4 .

Enter size of list 2 : 2

Enter any number : 1 .

Enter any number : 5 .

The union of two lists are : [1, 2, 3, 4, 5, 6]

Python Program to find the Intersection of lists

```
def intersection(a,b):
    return list(set(a) & set(b))

def main():
    aList = []
    bList = []
    n1 = int(input("Enter number of elements for list 1 :"))
    n2 = int(input("Enter number of elements for list 2 :"))
    print("For list 1 :")
    for x in range(0,n1):
        element = int(input("Enter element "+str(x+1)+" :"))
        aList.append(element)
    print("For list 2 :")
    for x in range(0,n2):
        element = int(input("Enter element "+str(x+1)+" :"))
        bList.append(element)
    print("The intersection is :")
    print(intersection(aList,bList))
```

main()
output

Enter number of elements for list 1 : 3
Enter number of elements for list 2 : 3.

For list 1 :

Enter element 1 : 5

Enter element 2 : 6

Enter element 3 : 7

List 2:

Enter element 1: 6

Enter element 2: 8

Enter element 3: 7

The intersection is :

[6, 7].

Python Program to create a list of Tuples with the first Element as the number and second element as the square of the Number.

```
l_range = int(input("Enter the lower range :"))
u_range = int(input("Enter the upper range :"))
a = [(x, x**2) for x in range(l_range, u_range+1)]
print(a)
```

Output :

Enter the lower range : 2 .

Enter the upper range : 6 .

$[(2, 4), (3, 9), (4, 16), (5, 25), (6, 36)]$

Python Program to Detect if Two Strings are Anagrams

```
s1 = input("enter first string :")
```

```
s2 = input("enter second string :")
```

```
if(sorted(s1) == sorted(s2)):
```

 print("the strings are anagrams.")

```
else
```

 print("the strings are not anagrams.")

Output :

enter first string : 2

enter second string : 5 .

the strings are not anagrams

Python Program to form a new string where the first character and the last character have been exchanged

def change(string):

 return string[-1:] + string[1:-1] + string[:1]

string = input("Enter string: ")

print("Modified string: ")

print(change(string))

Output

Enter string: shaleka

Modified string:

ataliks

Python Program to check if a string is a Pangram or

Not

from string import ascii_lowercase as asc_lower

def check(s):

 return set(asc_lower) - set(s.lower()) == set([])

string = input("Enter string: ")

if (check(string) == True):

 print("The string is a pangram")

else:

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

Output

Enter string: the five boxing wizards jump quickly

The string is a pangram

~~Python Program to Print all Permutations of a String in Lexicographic Order using Recursion~~

~~from math import factorial~~

~~def print_permutations_lexicographic_order(s):~~

~~""" Print all permutations of strings in lexicographic order. """~~

~~seq = list(s)~~

~~for i in range(factorial(len(seq))):~~ ~~print("".join(seq))~~

~~Python Program to print all Permutation of a string in Lexicographic order using recursion~~

~~from math import factorial~~

~~def print_permutations_lexicographic_order(s):~~

~~""" Print all permutations of string in lexicographic order. """~~

~~seq = list(s)~~

~~for i in range(factorial(len(seq))):~~
 ~~print("".join(seq))~~

~~nxt = get_next_permutation(seq)~~

~~# if seq is the highest permutation~~

~~if nxt is None:~~

~~# then reverse it.~~

~~seq.reverse()~~

~~else:~~

~~seq = nxt~~

~~def get_next_permutation(seq):~~

"" Return next greater lexicographic permutation. Return None if cannot.

This will return the next greater permutations of seq in lexicographic order. If seq is the highest permutation then this will return None. seq is a list.

""

if (len(seq) == 0):

return None.

next = get-next-permutation (seq[1:]).

if seq[1:] is the highest permutation
if next is None:

reverse seq[1:], so that seq[1:]
now is in ascending order.

seq[1:] = reversed (seq[1:])

find q, seq such that seq[q] is
the smallest element in seq[1:] such
that.

$seq[q] > seq[0]$

$q = 1$

while $q < \text{len}(seq)$ and $seq[0] >$
 $seq[q]:$

$q += 1$

if cannot find q, then seq is
the high permutation

if $q == \text{len}(seq):$

return None

swap seq[0] and seq[q].

$seq[0], seq[q] = seq[q], seq[0]$

return seq

else :

return [seq[0]] + next

s = input('Enter the string : ')

print(permute_lexicographic_order(s))

Output :

Enter the string : bag .

bag

bga

gab .

gba .

bag

bga

Python Program to Sum all the items in a Dictionary.

d = { 'A' : 100, 'B' : 540, 'C' : 239 }

print("total sum of values in the dictionary: ")
print(sum(d.values()))

The total sum of values in the dictionary:

879 .

Python Program to Multiply all the items in a Dictionary .

$d = \{ 'A': 10, 'B': 10, 'C': 239 \}$

$\text{tot} = 1$

for i in d:

$\text{tot} = \text{tot} * d[i]$

print (tot).

23900

Python Program to map two lists into a Dictionary

keys = []

values = []

$n = \text{int}(\text{input}(" \text{enter number of elements of dictionary : } "))$.

print ("for keys:").

for x in range (0, n):

 element = int (input ("enter element " +
 str (x+1) + ":"))

 keys.append (element).

print ("for values").

for x in range (0, n):

 element = int (input ("enter element " +
 str (x+1) + ":"))

 values.append (element).

$d = \text{dict}(\text{zip}(\text{keys}, \text{values}))$.

print ("the dictionary is :").

print (d).

Enter the number of elements for dictionary

For keys :-

Enter element :- 20.

The Dictionary

Enter elements :- 10.

{ 20 : 5, 10 : 6, 3 }

Enter element :- 30.

For values :-

Enter element :- 5.

Enter element :- 6.

Enter element :- 7.

Python Program to count the number of vowels present in a string using sets.

```
S = input ("enter string : ")
```

```
count = 0
```

```
vowels = set ("aeiou").
```

```
for letter in S:
```

```
    if letter in vowels:
```

```
        count += 1
```

```
print ("count of the vowels are : ")
```

```
print (count).
```

Enter string :- Computer Science

Count of the Vowels is 15 : 6

Python Program to Find Area of a Rectangle using Classes.

```
class rectangle():
```

```
    def __init__(self, breadth, length):
```

```
        self.breadth = breadth
```

```
        self.length = length
```

```
    def area(self):
```

```
        return self.breadth * self.length
```

```
a = int(input("Enter length of rectangle :"))
```

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

```
obj = rectangle(a, b)
```

```
print("Area of rectangle : ", obj.area())
```

```
print()
```

```
Enter length of Rectangle = 4
```

```
Enter Breadth of Rectangle = 5
```

```
Area of Rectangle : 30
```

Python Program to create a class and compute the area and the Perimeter of the circle.

```
import math
```

```
class circle():
```

```
    def __init__(self, radius):
```

```
        self.radius = radius
```

```
    def area(self):
```

```
        return math.pi * (self.radius ** 2)
```

```
    def perimeter(self):
```

```
        return 2 * math.pi * self.radius
```

```
r = int(input("Enter radius of circle :"))
obj = circle(r)
print("Area of circle : ", round(obj.area(), 2))
print("Perimeter of circle : ", round(obj.perimeter(), 2))
```

Enter radius of circle : 8

Area of the circle : 201.06.

Perimeter of the Circle : 50.27

Python Program to create a class which performs basic calculator operators

```
class cal():
    def __init__(self, a, b):
```

```
        self.a = a
```

```
        self.b = b
```

```
    def add(self):
```

```
        return self.a + self.b
```

```
    def mul(self):
```

```
        return self.a * self.b
```

```
    def div(self):
```

```
        return self.a / self.b
```

```
    def sub(self):
```

```
        return self.a - self.b
```

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

```
b = int(input("Enter second number :"))
```

```
obj = cal(a, b)
```

```
choice = 1
while choice != 0:
    print("0. Exit")
    print("1. Add")
    print("2. Subtraction")
    print("3. Multiplication")
    print("4. Division")
    choice = int(input("Enter choice:"))
    if choice == 1:
        print("Result: ", obj.add())
    elif choice == 2:
        print("Result: ", obj.sub())
    elif choice == 3:
        print("Result: ", obj.mul())
    elif choice == 4:
        print("Result: ", obj.div(2))
    elif choice == 0:
        print("exiting!")
    else:
        print("Invalid choice!!")
```

print()

Enter first number :- 5
Enter second number :- 4

- 0. Exit
- 1. Add
- 2. Subtraction
- 3. Multiplication
- 4. Division

Enter :- 1.

Result : 9.

- 0. Exit
- 1. Add
- 2. Subtraction
- 3. Multiplication
- 4. Division

Enter choice : 2

Result : 1.

- 0. Exit
- 1. Add
- 2. Subtraction
- 3. Multiplication
- 4. Division

Enter choice : 3.

Result : 10

Python Program to create a class in which one method accepts a string from the user and another prints it

```
class print1():
    def __init__(self):
        self.string = ""
    def get(self):
        self.string = input("Enter string : ")
    def put(self):
        print("String is : ")
        print(self.string)
```

```
obj = print1()
```

```
obj.get()
```

```
obj.put()
```

Output

Enter String :- Shalika

String is :- Shaleka