



Laxmi Charitable Trust's
**Sheth L.U.J. & Sir M.V. College of
Arts, Science & Commerce**

Dr. S Radhakrishnan Marg, Andheri (E), Mumbai – 400 069.

Certificate

This is to certify that, Mr./Ms. NEERAJ APPARI
Seat No. F129 studying in F.Y.B.Sc. SEM-II Computer
Science has satisfactorily completed the Practicals in the
Subject of DATA STRUCTURE as prescribed by University of
Mumbai, during academic year 2019-2020.

Signature

Subject in charge

Date: -

Signature

Co-ordinator B.Sc. C.S

Date: -

Signature

External Examiner

Date: -

ALNRAO



**SHETH L.U.J. COLLEGE OF ARTS &
SIR M.V. COLLEGE OF SCIENCE & COMMERCE**

Department of Computer Science

ALNRAO

Date

Index

Topic

26/01/2020 Write a program to demonstrate linear search algorithm

6/01/2020 Write a program to demonstrate binary search algorithm

14/01/2020 Write a Python program to implement bubble sorting

14/01/2020 Write a Python program to implement selection sorting

20/01/2020 Write a python program to print factorial of number

20/01/2020 Write a python program to print fibonacci series

20/01/2020 Write a python program to implement selection insertion sorting

27/01/2020 Write a python program to print exponential series

3/02/2020 Write a Python program to demonstrate Set and set ADT



**SHETH L.U.J. COLLEGE OF ARTS &
SIR M.V. COLLEGE OF SCIENCE & COMMERCE**

Department of Computer Science

ALNIRAO

- | | | |
|------------|---|-----------------|
| 3/02/2020 | Write a python program to demonstrate map data structure | ✓
24/02/2020 |
| 24/02/2020 | Implement binary tree and its traversal | ✓
24/02/2020 |
| 24/02/2020 | Implement stack operations | ✓
24/02/2020 |
| 02/03/2020 | Write a python program to demonstrate queue in data structure | ✓
03/03/2020 |
| 02/03/2020 | Write a python program to demonstrate linked list in data structure | ✓
03/03/2020 |

DEAPRABO

Output

Enter target value: 7

True

Target found

Enter target value: 28

False

Target not found

Output

Enter target value = 7

True

Target found

Enter target value = 28

False

Target not found

W



**SHETH L.U.J. COLLEGE OF ARTS &
SIR M.V. COLLEGE OF SCIENCE & COMMERCE**

Department of Computer Science

ALN PAU

Practical - I

Aim: Write a program to demonstrate linear search

def LinearSearch(l1, target);

n = len(l1)

found = False

for i in range(n):

if l1[i] == target:

found = True

print(found)

print("Target not found")

print("

break

if found == False:

print("False")

print("Target not found")

l1 = [1, 2, 3, 4, 5, 7, 8, 9]

target = int(input("Enter target value:"))

~~linearSearch(l1, target)~~

DATA

Output :

5 - location

Enter target value: 7
True
Target found

Enter target value: 28
False
Target not found

Output

Enter target value: 7
True
Target found

Enter target value: 28
False
Target not found

X



**SHETH L.U.J. COLLEGE OF ARTS &
SIR M.V. COLLEGE OF SCIENCE & COMMERCE**
Department of Computer Science

ALNRAO

Practical 2.

Aim: Write a search binary program to demonstrate binary algorithm

def binarySearch(thevalues, target):

low = 0

high = len(thevalues) - 1

while low <= high:

mid = (high + low) // 2

if thevalues[mid] == target:

return True

print("Target found")

elif target < thevalues[mid]:

high = mid - 1

else:

low = mid + 1

print("Target not found")

return False

thevalues = [1, 2, 3, 4, 5, 6, 7]

target = int(input("Enter target value:"))

print(binarySearch(thevalues, target))

DATA

Output

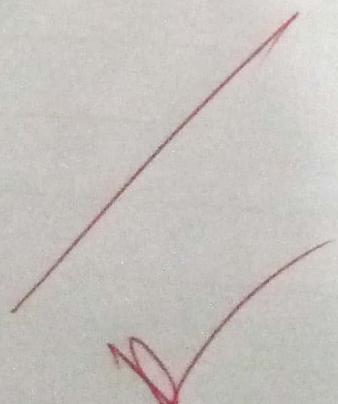
1, 54, 32, 34, 5, 43, 64

1, 32, 34, 5, 43, 54, 64

1, 32, 5, 34, 43, 54, 64

1, 5, 32, 34, 43, 54, 64

1, 5, 32,





**SHETH L.U.J. COLLEGE OF ARTS &
SIR M.V. COLLEGE OF SCIENCE & COMMERCE**
Department of Computer Science

ALNRAO

Practical - 3

Aim - Write a python program to implement bubble sorting algorithm.

def bubblesort(the seq):

n = len(the seq)

for i in range(n-1):

 for j in range(n-1):

 if the seq[j] > the seq[j+1]:

 tmp = the seq[j]

 the seq[j] = the seq[j+1]

 the seq[j+1] = tmp

 print(the seq)

bubblesort([1, 54, 64, 32, 34, 5, 43])

(45, 35, 24, 67)

Sorted array

24

35

45

67





**SHETH L.U.J. COLLEGE OF ARTS &
SIR M.V. COLLEGE OF SCIENCE & COMMERCE**

Department of Computer Science

ALNR AD

Practical - 4

`def selection sort (the seq):`

Aim: Write a Python program to implement selection sorting algorithm

`def Selectionsort (the seq):`

`n = len (the seq)`

`for i in range (n - 1)`

`small Ndx = i`

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

`if the seq [j] < the seq [small Ndx].`

`small Ndx = j`

`if small Ndx != i:`

`tmp = the seq [i]`

`the seq [i] = the seq [small Ndx]`

`the seq [small Ndx] = tmp`

`print (the seq)`

`the seq = [45, 35, 24, 67]`

`selection sort (the seq)`

`print ("The sorted array :")`

`for i in range (len (the seq)):`

~~`print ("%d" % the seq [i])`~~

Program

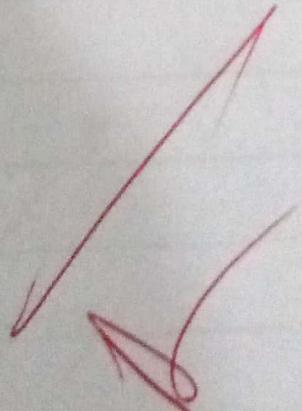
Output

factorial (2) : 2

factorial (3) : 6

factorial (4) : 24

factorial (5) : 120





**SHETH L.U.J. COLLEGE OF ARTS &
SIR M.V. COLLEGE OF SCIENCE & COMMERCE**
Department of Computer Science

ALNIRAT

Practical - 5(A)

Aim - Write a recursive function to print factorial of number

def recur(n):

if n == 1:

 return 1

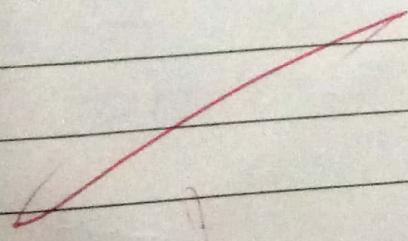
else:

 res = n * recur(n-1)

 print("Factorial (", n-1, ") : ", res)

 return res

print(recur(5))



ALURAU

Output
Fibonacci series

1
1
2
3
5

2



Practical - OS - CB

```
def recur_fibo(n):  
    if n <= 1:  
        return n  
    else:  
        return (recur_fibo (n-1)+recur_fibo(n-2))  
  
nterms = 10  
if nterms <= 0:  
    print ("Please enter a positive integer")  
else:  
    print ("Fibonacci sequence")  
    for i in range(nterms):  
        print (recur_fibo(i))
```



**SHETH L.U.J. COLLEGE OF ARTS &
SIR M.V. COLLEGE OF SCIENCE & COMMERCE**

Department of Computer Science

ALN102AO

Practical - 5-(C)

Aim - Write a recursion function to print exponential value

```
def power(a,b)
    if (b == 1):
        return(a)
    if (b != 1):
        return (a * power(a,b-1))
b = int(input ("Enter base"))
a = int(input ("Enter exponential value"))
print ("result:", power (a,b))
```

ALNRAO

Output:

Enter base : 2

Enter exponential value:3
result : 8

~~8~~



Practical - 06

Aim- Write a python program to implement insertion sorting

def insertion ():

```
    seg = input ("Enter the list of numbers :")
    seg = seg.split()
    seg = [int(x) for x in seg]
    for i in range (i, len(seg)):
        aim = seg[i]
        j = i - 1
        while (j >= 0) and (aim < seg[j]):
            seg[j+1] = seg[j]
            j = j - 1
        seg[j+1] = aim
    print ("Sorted array :")
    for i in range (len(seg)):
        print ("%d" % seg[i])
```

Insertion()

ALNIRAD

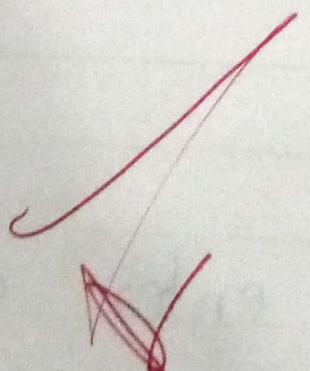
Output

Enter numbers 5 4 6 9 2

Sorted array: 2

3
4
5
6
9

4
5
6
9





Practical No 7

Write a python program to demonstrate set and
set ADT
if set1():

```
set1 = set(["Yash", "Shubham", "Saeel", "Neeraj", "Rajesh", "Prashant"])
print("Set 1:", set1)
set1.add("Om")
print("After Adding:", set1)
```

```
set2 = set(["Neel", "Sachin", "Rahul", "Prashant", "Raj", "Neeraj"])
print(set2)
print("Set 2:", set2)
```

```
set3 = set(["Neeraj", "Prashant"])
print("Set 3:", set3)
```

```
setunion = set1.union(set2)
print("Union:", setunion)
```

```
setintr = set1.intersection(set2)
print("Intersection:", setintr)
```

~~setdiff = set1.difference(set2)~~
print("Difference:", setdiff)

PLNRAD

Output:

Set1: {'Shubham', 'Neeraj', 'Saeel', 'Prashant', 'Yash'}
After Adding: {'Shubham', 'Neeraj', 'Saeel', 'Om', 'Prashant', 'Yash'}

Set2: {'Neeraj', 'Raj', 'Prashant', 'Sachin', 'Rahul', 'Neel'}
Set3: {'Neeraj', 'Prashant', 'Yash', 'Saeel'}

Union: {'Neel', 'Shubham', 'Rahul', 'Neeraj', 'Saeel', 'Om', 'Prashant', 'Sachin', 'Yash', 'Raj'}

Intersection: {'Neeraj', 'Prashant'}

Difference: {'Yash', 'Saeel', 'Om', 'Shubham'}

false

True

After removing {'Shubham', 'Neeraj', 'Saeel', 'Prashant', 'Yash'}

S

not equal

Shubham

Neeraj

Saeel

Prashant

Yash



**SHETH L.U.J. COLLEGE OF ARTS &
SIR M.V. COLLEGE OF SCIENCE & COMMERCE**

Department of Computer Science

ALNRAC

Q: print (set3) is subset

print (set3.issubset(set1))

print (set3.issubset(set2))

set1.remove("Om")

print ("After removing:", set1)

print (len (set1))

if set1 == set2 :

 print ("Both are equal")

else :

 print ("Both are not equal")

for iter1 in iter(set1):

 print (iter1)

Set1()

✓
P



**SHETH L.U.J. COLLEGE OF ARTS &
SIR M.V. COLLEGE OF SCIENCE & COMMERCE**

Department of Computer Science

DLNR AU

Practical No. 3

Aim - Write a Python program to demonstrate map data structure

```
class map():
    def __init__(self):
        self.entrylist = dict()
    def length(self):
        return len(self.entrylist)
    def contains(self, key):
        if key in self.entrylist:
            return True
        else:
            return False
    def value_of(self, key):
        if key in self.entrylist:
            return self.entrylist[key]
    def iterator(self):
        for i, v in self.entrylist:
            if i == "one":
                print(v)
```

```
bench = Map()
bench.add (1, "Saee1")
bench.add (2, "Shubham")
bench.add (3, "Sachin")
bench.length()
print(bench.length())
print(chonda(bench))
print(bench.contains(1))
print(bench.contains(2))
bench.remove(3)
print(bench.iterator())
print(bench.valueof(1))
```

Output :

{1: "Saee1", 2: "Shubham", 3: "Sachin"}

True

True

1: Saee1

2: Shubham

Saee1



Practical No - 9

Aim: Implement binary tree and its traversal

Class Node:

```
def __init__(self, key):  
    self.left = None  
    self.right = None  
    self.val = key
```

```
def printInorder(root):
```

```
    if root:  
        printInorder(root.left)  
        print(root.val),  
        printInorder(root.right)
```

```
def printPostorder(root):
```

```
    if root:  
        printPostorder(root.left)  
        printPostorder(root.right)  
        print(root.val),
```

```
def printPreorder(root):
```

```
    if root:  
        printInorder  
        printPreorder  
        print(root.val),  
        printPreorder(root.left)  
        printPreorder(root.right)
```

ACN RAO

Output

Preorder traversal of binary tree is:

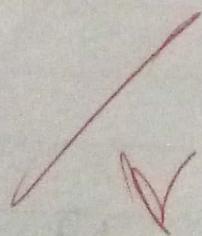
1
2
5
3

Inorder traversal of binary tree is:

2
5
1
3

Postorder traversal of binary tree is

5
2
3
1



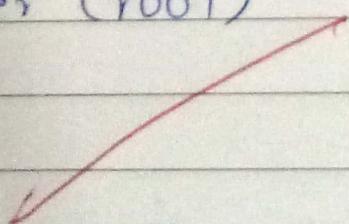


**SHETH L.U.J. COLLEGE OF ARTS &
SIR M.V. COLLEGE OF SCIENCE & COMMERCE**
Department of Computer Science

ALNIRAO

root = Node(1)
root.left = Node(2)
root.right = Node(3)
root.left.right = Node(4)
root.left.right = Node(5)

print ("Preorder traversal of binary tree is:")
print Preorder(root)
print ("\nInorder traversal of binary tree is:")
print Inorder(root)
print ("\nPostorder traversal of binary tree is:")
print Postorder(root)



ALMOST

Output:

Initial stack
['a', 'b', 'c']

Elements popped from stacks

c

b

a

Stack after elements are popped:

~~stack~~ now



Practical No - 10

stack = []

Aim: implement stack operations
SI

stack = []

stack.append('a')

stack.append('b')

stack.append('c')

print ('Initial stack')

print (stack)

print ('\nElements popped from stacks')

print (stack.pop())

print (stack.pop())

print (stack.pop())

print ('\nStack after elements are popped:')

print (stack)

Ans: Ravi

Output -

Initial queue
['Yash', 'Saee', 'Ravi']

Elements dequeued from queue

Yash

Saee

Ravi

Queue after removing element
[]



**SHETH L.U.J. COLLEGE OF ARTS &
SIR M.V. COLLEGE OF SCIENCE & COMMERCE**
Department of Computer Science

AUNRI

Practical + 1

Aim- Write a python program to demonstrate queue in data structures

queue = []

```
queue.append('Yash')
queue.append('Saeei')
queue.append('Ravi')
```

```
print("Initial queue")
print(queue)
```

```
print("In Elements dequeued from queue")
print(queue.pop(0))
print(queue.pop(0))
print(queue.pop(0))
```

```
print("In Queue after removing elements")
print(queue)
```

ALNIRAO

Output -

Mon

Tue

Wed

Thur

Fri



**SHETH L.U.J. COLLEGE OF ARTS &
SIR M.V. COLLEGE OF SCIENCE & COMMERCE**

Department of Computer Science

PINRDO

Practical - No - 12

Aim - Write a Python program to demonstrate linked list
in data structure

Class Node:

```
def __init__(self, dataval=None):
    self.dataval = dataval
    self.nextval = None
```

Class Slinked list:

```
def __init__(self):
    self.headval = None
```

```
def listprint(self):
    printval = self.headval
    while printval is not None:
        print(printval.dataval)
        printval = printval.nextval
```

```
list = Slinked list()
```

```
list.headval = Node("Mon")
```

```
e2 = Node("Tue")
```

```
e3 = Node("Wed")
```

```
e4 = Node("Thu")
```

list.headval.nextval = e2

list.e2.nextval = e3

list.listprint()