Name: Nadeem Shaikh
Roll No: 30
MCA-1 Sem-2

1. WAP to display all types of pyramids of stars

Code:

```
def halfPyramid():
    for i in range(5):
        for j in range(i+1):
            print(end="* ")
        print()

def invertedHalfPyramid():
    for i in range(5):
        for j in range(i, 5):
            print(end="* ")
        print()

def fullPyramid():
    for i in range(5):
        for s in range(5, i+1, -1):
            print(end=" ")
        for j in range(i + 1):
            print(end="* ")
        print()

def invertedFullPyramid():
    for i in range(5):
        for s in range(i):
            print(end=" ")
        for j in range(i, 5):
            print(end="* ")
        print()

halfPyramid()
print("========================")
invertedHalfPyramid()
print("========================")
fullPyramid()
print("========================")
invertedFullPyramid()
```

Name: Nadeem Shaikh
Roll No: 30
MCA-1 Sem-2

Output:

```
*
* *
* * *
* * * *
* * * * *
========================
* * * * *
* * * *
* * *
* *
*
========================
   *
  * *
 * * *
* * * *
* * * * *
========================
* * * * *
 * * * *
  * * *
   * *
    *
```

Name: Nadeem Shaikh
Roll No: 30
MCA-1 Sem-2

2. **WAP to display multiplication table of all numbers from 1 to 10**

Code:

```
# Python Program to Print Multiplication Table of a Number

num = int(input("Enter the number: "))
print("Multiplication Table of", num)
for i in range(1, 11):
   print(num,"X",i,"=",num * i)
```

Output:

```
E:\Nadeem_MCA1\Python_Program>python table_multiplication.py
Enter the number: 5
Multiplication Table of 5
5 X 1 = 5
5 X 2 = 10
5 X 3 = 15
5 X 4 = 20
5 X 5 = 25
5 X 6 = 30
5 X 7 = 35
5 X 8 = 40
5 X 9 = 45
5 X 10 = 50
```

Name: Nadeem Shaikh
Roll No: 30
MCA-1 Sem-2

3. **WAP to implement tower of Hanoi**

Code:

```
# Python Program to implement tower of Hanoi

def hanoi(disks, source, auxiliary, target):
    if disks == 1:
        print('Move disk 1 from peg {} to peg {}.'.format(source, target))
        return

    hanoi(disks - 1, source, target, auxiliary)
    print('Move disk {} from peg {} to peg {}.'.format(disks, source, target))
    hanoi(disks - 1, auxiliary, source, target)


disks = int(input('Enter number of disks: '))
hanoi(disks, 'A', 'B', 'C')
```

Output:

```
E:\Nadeem_MCA1\Python_Program>python tower_of_hanoi.py
Enter number of disks: 4
Move disk 1 from peg A to peg B.
Move disk 2 from peg A to peg C.
Move disk 1 from peg B to peg C.
Move disk 3 from peg A to peg B.
Move disk 1 from peg C to peg A.
Move disk 2 from peg C to peg B.
Move disk 1 from peg A to peg B.
Move disk 4 from peg A to peg C.
Move disk 1 from peg B to peg C.
Move disk 2 from peg B to peg A.
Move disk 1 from peg C to peg A.
Move disk 3 from peg B to peg C.
Move disk 1 from peg A to peg B.
Move disk 2 from peg A to peg C.
Move disk 1 from peg B to peg C.
```

4. **WAP to calculate simple interest using a user defined function. Accept amount, duration from user. Set interest rate as default parameter**.

Code:

```
# Python Program for calculating simple interest. Taking values from user.

print("Enter the Principle Amount: ")
P = int(input())
print("Enter Time Period: ")
N = float(input())
print("Enter Rate of Interest (%): ")
R = float(input())
si = (P*N*R)/100
print("\nSimple Interest Amount: ")
print(si)
```

Ouput:

```
E:\Nadeem_MCA1\Python_Program>python simple_interest.py
Enter the Principle Amount:
25000
Enter Time Period:
5
Enter Rate of Interest (%):
6.25

Simple Interest Amount:
7812.5
```

Name: Nadeem Shaikh
Roll No: 30
MCA-1 Sem-2

5. **WAP to count even and odd numbers in a list**

Code:

```
# Python Program to count odd & even numbers in a list

num_list=[]
n=int(input("Enter the Starting of the range:"))
k=int(input("Enter the Ending of the range:"))
for i in range(n,k):
  num_list.append(i)
print("Original Number List:", num_list)
even_list=[]
odd_list=[]
for i in range(len(num_list)):
  if(num_list[i]%2==0):
  even_list.append(num_list[i])
  else:
  odd_list.append(num_list[i])
print("Even Numbers List:", even_list)
print("Odd Numbers List:", odd_list)
```

Output:

```
E:\Nadeem_MCA1\Python_Program>python
odd_even_list.py Enter the Starting of the range:2
Enter the Ending of the range:20
Original Number List: [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19]
Even Numbers List: [2, 4, 6, 8, 10, 12, 14, 16, 18]
Odd Numbers List: [3, 5, 7, 9, 11, 13, 15, 17, 19]
```

Name: Nadeem Shaikh
Roll No: 30
MCA-1 Sem-2

6. WAP to find sum of all numbers, min,max, mean,median,mode of numbers in a list

Code:

```
# Calculate sum of numbers in a list
# for mode of elements
from collections import Counter

n = [1, 2, 3, 4, 5, 6, 7, 8]
a = len(n)
n.sort()

#Min number of list
minimum = min(n)
print("min number is - ",minimum)

#Max number of list
maximum = max(n)
print("max number is - ",maximum)

#Mean number of list
get_sum = sum(n)
mean = get_sum / a
print("Mean / Average is: " + str(mean))

#Median number of list
if a % 2 == 0:
   median1 = n[a//2]
   median2 = n[a//2 - 1]
   median = (median1 + median2)/2
else:
   median = n[a//2]
print("Median is: " + str(median))

#Mode number of list
data = Counter(n)
get_mode = dict(data)
mode = [k for k, v in get_mode.items() if v == max(list(data.values()))]

if len(mode) == a:
   get_mode = "No mode found"
else:
   get_mode = "Mode is / are: " + ', '.join(map(str, mode))

print(get_mode)
```

Output:

```
E:\Nadeem_MCA1\Python_Program>python
min_max_median.py min number is - 1
max number is - 8
Mean / Average is:
4.5 Median is: 4.5
No mode found
```

Name: Nadeem Shaikh
Roll No: 30
MCA-1 Sem-2

7. WAP to find sum and multiplication of two matrices implemented using list

Code:

```
# Python program to multiply two matrices using a list


A = [[12, 7, 3],
    [0, 5, 6],
    [8, 8, 9]]
B = [[5, 8, 1, 2, ],
    [0, 0, 0, 0],
    [4, 5, 9, 1]]

# multiply matrix
res = [[sum(a * b for a, b in zip(A_row, B_col))
       for B_col in zip(*B)] for A_row in A]
for r in res:
   print(r)
```

Output:

```
E:\Nadeem_MCA1\Python_Program>python
matrix_using_list.py [72, 111, 39, 27]
[24, 30, 54, 6]
[76, 109, 89, 25]
```

Name: Nadeem Shaikh
Roll No: 30
MCA-1 Sem-2

8. WAP to store student roll number and marks using dictionary.

Implements following functions
Add a record, delete, update marks, search a roll number and display marks, sort the records in ascending and descending order, display student information with highest marks. Implement a menu driven program

Output:

E:\Nadeem_MCA1\Python_Program>python

Student_Management.py Functions used in the program as,

1.Accept Student details
2.Display Student Details
3.Search Details of a
Student 4.Delete Details
of Student 5.Update
Student Details 6.Exit

List of Students

Name : Andre
RollNo : 1
Marks1 : 100
Marks2 : 100

Name : Bob
RollNo : 2
Marks1 : 90
Marks2 : 90

Name : Clara
RollNo : 3
Marks1 : 80
Marks2 : 80

Student
Found, Name
: Bob RollNo :
2
Marks1 : 90
Marks2 : 90

2
List after
deletion Name :
Andre RollNo :
1
Marks1 : 100
Marks2 : 100

Name : Clara
RollNo : 3
Marks1 : 80
Marks2 : 80

2
List after
updation Name :
Andre RollNo : 1
Marks1 : 100
Marks2 : 100

Name : Clara
RollNo : 2
Marks1 : 80
Marks2 : 80

Name: Nadeem Shaikh
Roll No: 30
MCA-1 Sem-2

9. WAP to implement function decorator to display cube of a number

Code:

```
# Implement function decorator to display cube of a number

def decor(func):
    def inner():
    n = int(input(" Enter the number : "))
    return n * n * n
    return inner

@decor
   def num():
  return inner

 print(num())
```

Output:

```
Enter the number : 3

27
```

Name: Nadeem Shaikh
Roll No: 30
MCA-1 Sem-2

10. Write a Program to implement generator function to display square of numbers from 1 to 10

Code:

```
import math

def square():
    for x in range(1,11):
        yield (x*x)

for val in square():
    print(val)
```

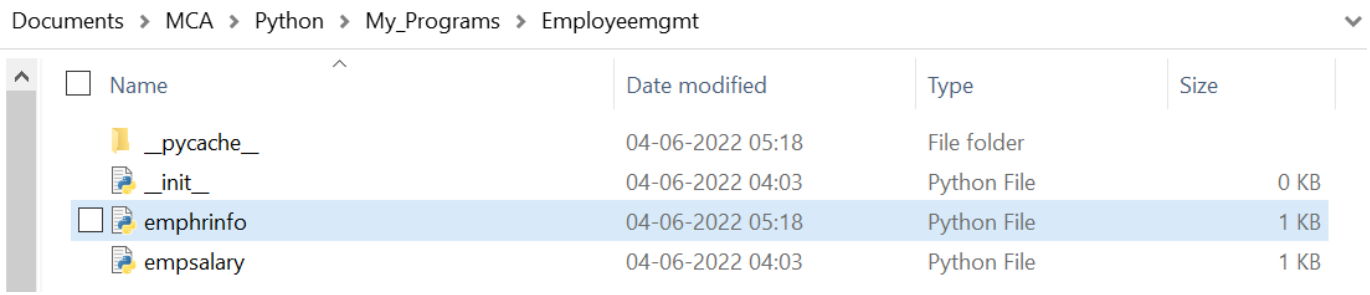
Output:

```
E:\Nadeem_MCA1\Python_Program>python Generator_function.py
1
4
9
16
25
36
49
64
81
100
```

Name: Nadeem Shaikh
Roll No: 30
MCA-1 Sem-2

11. WAP to implement a package and module

Package- Employeemgmt
Module empsalary - function to calculate gross and net salary
Module emphrinfo- function to display employee information i.e. name, designation, dept, qualification and experience



Code: emphrinfo.py

```
class Employee:
    count=0
    def_init_(self, name, desig, dept, quali, exp):
        self.name=name
        self.desig=desig
        self.dept=dept
        self.quali=quali
        self.exp=exp
        Employee.count+=1
    def displayCount(self):
        print("There are %d employees" % Employee.count)
    def displayDetails(self):
        print("Name:", self.name, ", Designation:", self.desig, ", Deptartment:", self.dept, ", Qualification:",
            self.quali, ", Experience:", self.exp)
e1=Employee("John", "Manager", "Production", "BE. Mech", "7 yrs")

print("Details of Employee:")
e1.displayDetails()
```

Name: Nadeem Shaikh
Roll No: 30
MCA-1 Sem-2

Code: empsalary.py

```
def da(basic):
  da = basic*80/100
  return da

def hra(basic):
  hra = basic*15/100
  return hra

def pf(basic):
  pf = basic*12/100
  return pf

def itax(gross):
  tax = gross*0.1
  return tax

basic = float(input("Enter basic salary: "))

gross = basic+da(basic)+hra(basic)
print("Your gross salary: {:10.2f}".format(gross))

net = gross-pf(basic)-itax(gross)
print("Your net salary: {:10.2f}".format(net))
```

Code: package.py

```
from Employeemgmt import emphrinfo
from Employeemgmt import empsalary
```

Output:

```
E:\Nadeem_MCA1\Python_Program>python package.py
Details of Employee:

Name: John , Designation: Manager , Deptartment: Production , Qualification: BE. Mech , Experience: 7 yrs
Enter basic salary: 100000
Your gross salary:     195000.00
Your net salary:     163500.00
```

Name: Nadeem Shaikh
Roll No: 30
MCA-1 Sem-2

12. WAP to implement a class to store student information as id, name, marks. Implement all class, instance, public, private attributes. Implement instance, class, constructor, destructor, getter and setter methods

Code:

```
class Student:
counter = 0
classname = "MCA1"

def __init__(self, r, n):
self.rollno = r
self.name=n
Student.counter += 1
def display(self):
print("Roll number :",self.rollno)
print("Name ",self.name)
#setter method

def set_name(self,name):
self.name = name
#getter method

def get_name(self):
return self.name
@classmethod

def displaytotal(cls):
print("Total Students :",Student.counter)
@staticmethod

def dispclass():
print("Student class name is: ",Student.classname)
s1 = Student(1, "Alex")
s1.display()

Student.displaytotal()
s2 = Student(2, "Bob")
s2.display()

Student.displaytotal()
s3 = Student(3,"Mima")
s3.display()
s3.set_name("Andre")
print(s3.get_name())
```

Output:

Roll number : 1
Name Alex Total
Students : 1 Roll
number : 2 Name
Bob
Total Students : 2
Roll number : 3
Name Mima Andre

13. WAP to validate email id, password, url and mobile using regular

Code:

```
import re

email_regex = re.compile(r"[^@]+@[^@]+\.[^@]+")
a = "nadeem.iist@gmail.com"
if email_regex.match(a):
   print("Mentioned Email address",a,"is valid")
else:
   print("Mentioned Email address",a,"is not valid")


def main():
  passwd = 'NS#supp0rt#123@'
  reg = "^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)(?=.*[@$!%*#?&])[A-Za-z\d@$!#%*?&]{6,20}$"

  # compiling regex
  pat = re.compile(reg)

  # searching regex
  mat = re.search(pat, passwd)

  # validating conditions
  if mat:
            print("Mentioned Password",passwd,"is valid")
  else:
            print("Mentioned Password",passwd,"is invalid !!")

# Driver Code
if __name__ == '__main__':
  main()


def isValidURL(str):

   # Regex to check valid URL
   regex = ("((http|https)://)(www.)?" +
         "[a-zA-Z0-9@:%._\\+~#?&//=]" +
         "{2,256}\\.[a-z]" +
         "{2,6}\\b([-a-zA-Z0-9@:%" +
         "._\\+~#?&//=]*)")

   # Compile the ReGex
   p = re.compile(regex)

   # If the string is empty
   # return false
   if (str == None):
      return False

   # Return if the string
   # matched the ReGex
   if(re.search(p, str)):
      return True
   else:
```

```
        return False


# Test Case 1:
url = "https://moonjeinstitute.com"

if(isValidURL(url) == True):
   print("Yes, mentioned URL",url,"is Valid")
else:
   print("No, mentioned URL",url,"is not Valid")



def isValid(s):

  # 1) Begins with 0 or 91
  # 2) Then contains 7 or 8 or 9.
  # 3) Then contains 9 digits
  Pattern = re.compile("(0|91)?[7-9][0-9]{9}")
  return Pattern.match(s)

# Driver Code
s = "347873923408"
if (isValid(s)):
  print ("Mentioned Mobile Number: ",s,"is Valid")
else :
  print ("Mentioned Mobile Number: ",s,"is not Valid")
```

Output:

```
E:\Nadeem_MCA1\Python_Program>python regex.py

        Mentioned Email address nadeem.iist@gmail.com is valid
        Mentioned Password NS#supp0rt#123@ is valid
        Yes, mentioned URL https://www.moonjeinstitute.com is Valid
        Mentioned Mobile Number:  347873923408 is not Valid
```