In [ ]:

```
#********************************************************************
#
#                Big Data Assignment
#
#********************************************************************
#Task1
# A Fibonacci series (starting from 1) written in order without any spaces in between,
# thus producing a sequence of digits.
# Write a Python application to find the Nth digit in the sequence.
# o Write the function using standard for Loop
# o Write the function using recursion


# Function to Create a Fibonacci Series using For Loop
# o Write the function using standard for Loop
cnt = 1
def Create_Fibseries():
    a = 1
    b = 1
    ser = str(a) +str(b)
#    print(ser)
    for i in range(20):
        b,a = a+b,b
        ser  = ser + str(b)
    return ser

Fibseries = Create_Fibseries()

print(" The Fibonacci Series - %s " %Fibseries + " ....\n",'-'*80)

cnt_Num = int(input(" Which Digit of Fibonacci Series needed : "))

for i in Fibseries:
#    print(cnt_Num)
    if (cnt_Num == cnt):
        print(" The Digit is (by Using For Loop)  - %s" % i)
        break
    cnt +=1
```

In [ ]:

```
# ○ Write the function using recursion

def Create_FibSeries(ndigit):
    cnt = ndigit
    if (ndigit < 0):
        print("wrong number")
    elif (ndigit == 0):
        return 0
    elif (ndigit == 1):
        return 1
    else:
        return (Create_FibSeries(ndigit-1)+ Create_FibSeries(ndigit-2))

digit_num = int(input("Please Enter Which Digit of Fibonacci to see : "))
# Create_FibSeries(digit_num)

FibSeq = "1"
for i in range(2,(digit_num +1)):
    FibSeq = FibSeq + str(Create_FibSeries(i))

print(" The Fibonacci Series is \n",'-'*80)
print(FibSeq)

i=0
cnt = 1
for i in FibSeq:
    if (digit_num == cnt):
        print("The Digit is(by using recursion) -  %s" % i)
        break
    cnt +=1
```

In [ ]:

```
# Create a calculator to work with rational numbers.
# Requirements:
# o It should provide capability to add, subtract, divide and multiply rational numbers

# o Create a method to compute GCD (this will come in handy during operations on rational)
# Add option to work with whole numbers which are also rational numbers i.e. (n/1)
#- achieve the above using auxiliary constructors - enable method overloading to
# enable each function to work with numbers and rational.

import fractions,math

print('''Select Option (1/2/3/4)
          1. Add
          2. Substract
          3. Multiply
          4. Divide
          5. GCD \n''')
inp = int(input("Enter Option (1/2/3/4/5)"))
numType = input("Operation on Rational(R) or Whole number(W)")



def addnum(f1,f2):
    return(" %s + %s =  %s" % (f1,f2,(f1+f2)))

def subnum(f1,f2):
    return(" %s - %s =  %s" % (f1,f2,(f1-f2)))

def mulnum(f1,f2):
    return(" %s * %s =  %s" % (f1,f2,(f1*f2)))

def divnum(f1,f2):
    return(" %s / %s =  %s" % (f1,f2,(f1/f2)))

if(numType in ('W','w','R','r')):
    val1 = input("Please Enter Number1 - ")
    val2 = input("Please Enter Number2 - ")

    f1 = fractions.Fraction(val1)
    f2 = fractions.Fraction(val2)

    if (inp == 1):
        print(addnum(f1,f2))
    elif(inp == 2):
        print(subnum(f1,f2))
    elif(inp == 3):
        print(mulnum(f1,f2))
    elif(inp == 4):
        print(divnum(f1,f2))
    elif(inp == 5):
        print(fractions.gcd(f1,f2))
    else:
        print('-'*80,"\nWrong Option")
else:
    print('-'*80,"\nWrong Option")


```

In [ ]:

```python
# Task3
# 1. Write a Simple Program to Show inheritance



class Person:

    def __init__(self, first, last):
        self.firstname = first
        self.lastname = last

    def Name(self):
        return self.firstname + " " + self.lastname

class Student(Person):

    def __init__(self, first, last, rollnum):
        Person.__init__(self,first, last)
        self.rollnum = rollnum

    def GetStudent(self):
        return self.Name() + ", " +  self.rollnum

x = Person("Vivek", "Khare")
y = Student("Vivek", "Khare", "1731")
print("-"*80,"\n Program to show simple inheritance , Base Class-Person, Child Class -Student\n")

print(x.Name())
print(y.GetStudent())
```

In [ ]:

```
# 2. Write a Simple Program to Show Multiple Inheritance
class Person:

    def __init__(self, first, last):
        self.firstname = first
        self.lastname = last

    def Name(self):
        return self.firstname + " " + self.lastname

class School:

    def __init__(self,SchName,ClsName):
        self.schoolName = SchName
        self.className = ClsName

    def schName(self):
        return self.schoolName

    def clsName(self):
        return self.className



class Student(Person,School):

    def __init__(self, first, last, SchName,ClsName):
        Person.__init__(self,first, last)
        School.__init__(self,SchName,ClsName)


    def GetStudentDet(self):
        return self.Name() + " Studying in " +  self.clsName() +" of School   " +self.schName()

#x = Person("Vivek", "Khare")
print("-"*80,"\n Program to show multiple inheritance , Base Class-Person, Child Class -Student, School\n")

y = Student("Vivek", "Khare", "APS","Class 11")

#print(x.Name())
print(y.GetStudentDet())
```

In [ ]:

```
#### 3.Write a partial function to add three numbers in which one number is constant and
# two numbers can be passed as inputs and define another method which can take the
# partial function as input and squares the result.

from functools import partial

# partial function to add 3 numbers
def add(a,b,c):
    return a+b+c


add_part = partial(add,1)

# partial function to add 3 numbers , one being constant and other two being inputs
print(" Partial Function created add_part, that takes two inputs , but adds 3 numbers with one constant(1)\n",'-
num1 = int(input("Please Enter number 1 to add - "))
num2 = int(input("Please Enter number 2 to add - "))

print(add_part(num1,num2))

def square(x=add_part(num1,num2)):
    return (x**2)

# method that takes partial function as input and returns square of the number
print(" Method defined Square, that takes partial function as input and returns square\n",'-'*80)
print(square())



```

In [ ]:

```
#.Write a program to print the prices of 4 courses of Acadgild:
# Android-12999,Big Data Development-17999,Big Data Development-17999,Spark-19999
# using match and add a default condition if the user enters any other course


val = int(input('''\n Enter option to know Prices of the Courses offered \n
       1 - For Android
       2 - For Big Data Development
       3 - For Big Data Developement
       4 - For Spark \n'''))

if(val == 1):
    print(" Price for Android Course - 12999")
elif(val == 2 or val == 3):
    print(" Price for Big Data development Course - 17999")
elif(val == 4):
    print(" Price for Spark Course - 19999")
else:
    print(" No Such Course")
```