1a. Develop a program to read the student details like Name, USN, and Marks in three subjects. Display the student details, total marks and percentage with suitable messages.

```
name = input("Enter the name: ")
usn = input("Enter USN : ")
subject1 = int(input("Enter the marks scored in subject 1:  "))
subject2 = int(input("Enter the marks scored in subject 2:  "))
subject3 = int(input("Enter the marks scored in subject 3:  "))
Total_Marks = subject1 + subject2 + subject3
percentage = Total_Marks/3
print("\nStudents Details: ")
print("Name of the student is", name)
print("USN of the student is", usn)
print("Marks scored in subject1 is: ",subject1)
print("Marks scored in subject2 is: ",subject2)
print("Marks scored in subject3 is: ",subject3)
print("The total marks are",Total_Marks)
print("The percentage is {:.2f}".format(percentage))
```

1 b. Develop a program to read the name and year of birth of a person. Display whether the person is a senior citizen or not.

```
import datetime
name = input("Enter the name of person: ")
Birth_year = int(input("Enter the year of birth of person:"))
today = datetime.date.today()
year = today.year
age = year-Birth_year
if age>=60:
    print(name,"is a senior citizen")
else:
    print(name,"is not a senior citizen")
```

2a. Develop a program to generate Fibonacci sequence of length (N). Read N from the console.

```
num = int(input("Enter a number: "))
first_number = 0
second_number = 1
if num<= 0:
   print("please enter positive number")
elif num == 1:
   print("Fibonacci Series: ",first_number)
else:
   print("Fibonacci Series: ",first_number,second_number,end='  ')
   for i in range(2,num):
      next_number = first_number+second_number
      first_number = second_number
      second_number = next_number
      print(next_number,end='  ')
```

2b. Write a function to calculate factorial of a number. Develop a program to compute binomial coefficient (Given N and R).

```
#To find the factorial of a number
def factorial(num):
   result = 1
   for i in range(num,0,-1):
      result = result*i # [ if num = 5 then i value will be 5,4,3,2,1]
   print("Factorial of ",num, "is ",result)
num = int(input("Enter a number to find factorial : "))
factorial(num) # Function call

# To find binomial coeffiecient of a number
from math import factorial as fact
n= int(input("\nEnter the value of n: "))
r = int(input("Enter the value of r (r<n) : "))
```

```
if(n<r):
    print("n value should be greater than r")
    print("The binomial Coefficient of",n,"and",r,"is : ", None)
else:
    result = fact(n)/(fact(r)*fact(n-r))
    print("The binomial Coefficient of",n,"and",r,"is : ", result)
```

3. Read N numbers from the console and create a list. Develop a program to print mean, variance and standard deviation with suitable messages

```
import math
List_1 = list()
number = int(input( "Enter the number of elements you want: "))
print ('Enter numbers in array: ')
for i in range(int(number)):
    n = int(input( "number  : "))
    List_1.append(n) # to add the number in the list
print("The List is ", List_1)
sum_series=0
sum_square=0
for x in List_1:
    sum_series = sum_series+x
    mean = sum_series/number
for x in List_1:
    sum_square =sum_square+(x-mean)**2
#print("sum_square=", sum_square)
variance=sum_square/number
std_dev = math.sqrt(variance)
#print("the mean=",mean,"variance=",variance, "and standard deviation=", std_dev)
print("\nmean = {:.3f} \nvariance = {:.2f} \nstandard deviation = {:.2f}".format(mean, variance, std_dev))
```

4. Read a multi-digit number (as chars) from the console. Develop a program to print the frequency of each digit with suitable message.

```
num=input("input multi-digit number:")
n=len(num)
n0,n1,n2,n3,n4,n5,n6,n7,n8,n9=0,0,0,0,0,0,0,0,0,0
for i in range(n):
    if num[i]=='0':
        n0+=1 # same as n0 = n0+1
    elif num[i]=='1':
        n1+=1
    elif num[i]=='2':
        n2+=1
    elif num[i]=='3':
        n3+=1
    elif num[i]=='4':
        n4+=1
    elif num[i]=='5':
        n5+=1
    elif num[i]=='6':
        n6+=1
    elif num[i]=='7':
        n7+=1
    elif num[i]=='8':
        n8+=1
    elif num[i]=='9':
        n9+=1
dfreq=[n0,n1,n2,n3,n4,n5,n6,n7,n8,n9]
print("the number", num, "has:")
for i in range(10):
    if dfreq[i]==0:
        continue
 print(i,"digit", dfreq[i],"times")
```

5. Develop a program to print 10 most frequently appearing words in a text file. [Hint: Use dictionary with distinct words and their frequency of occurrences. Sort the dictionary in the reverse order of frequency and display dictionary slice of first 10 items]

```
ifile=open("abc.txt")
dict_words={}

for line in ifile:
    words=line.split() # converting into List
    #print("\n",words)
    for word in words: # converting into dictionary
        dict_words[word]=dict_words.get(word,0)+1
        #print(dict_words)

list_words=[]
for key, val in dict_words.items():
    #print(key,val)
    list_words.append((val,key))
    list_words.sort(reverse=True)
print("The slice of first 10 items of sorted dictionary are: \n ")
print(list_words[0:10])
```

6. Develop a program to sort the contents of a text file and write the sorted contents into a separate text file. [Hint: Use string methods strip(), len(), list methods sort(), append(), and file methods open(), readlines(), and write()]

```
ifile=open("file1.txt")
ofile=open("file2.txt", mode='w')
word_list=[]
for line in ifile:
    words=line.split()
    for word in words:
        word_list.append(word)
```

```
word_list.sort()
print(word_list)
for word in word_list:
    ofile.write(word+" ")
ofile.close()
```

7. Develop a program to backing Up a given Folder (Folder in a current working directory) into a ZIP File by using relevant modules and suitable methods.

```
import os
os.getcwd()
import os
import sys
import pathlib
import zipfile
print(os.getcwd())
dirName = input("Enter Folder name that you want to backup : ")
if not os.path.isdir(dirName):
    print("Directory", dirName, "doesn't exists")
    sys.exit(0)

curDirectory = pathlib.Path(dirName)
with zipfile.ZipFile("myZip_Hithesh.zip", mode="w") as archive:
    for file_path in curDirectory.rglob("*"): #adds all the files in the folder
        archive.write(file_path,arcname=file_path.relative_to(curDirectory))
if os.path.isfile("myZip_Hithesh.zip"):
    print("Archive", "myZip_Hithesh.zip", "created successfully")
else:

    print("Error in creating zip archive")
```

8. Write a function named DivExp which takes TWO parameters a, b and returns a value c (c=a/b). Write suitable assertion for a>0 in function DivExp and raise an exception for when b=0. Develop a suitable program which reads two values from the console and calls a function DivExp.

```
def DivExp(a,b):
    try:
        assert a>=0,"number a is negative."
        if b==0:
            raise ZeroDivisionError("Zero Division Error change denominator")
        c=a/b
        return c
    except ZeroDivisionError as x:
        print(x)
    except AssertionError as x:
        print("Assertion failure: "+str(x))
x=int(input("enter first number:"))
y=int(input("enter second number:"))
print(x,"  ",y)
print(DivExp(x,y))
```

9. Define a function which takes two objects representing complex numbers and returns new complex number with a addition of two complex numbers. define a suitable class ‘complex’ to represent the complex number. develop a program to read n (n >=2) complex numbers and to compute the addition of n complex numbers.

```
class Complex:
    def __init__(self, real=0, img=0):
        self.real=real
        self.img=img
    def __add__(c1, c2):
        return Complex(c1.real+c2.real, c1.img+c2.img)
    def __str__(self):
        return ("{}+i({})".format(self.real, self.img))
```

```
complex_list=[]
n=int(input("how many complex numbers do you want to add?"))
for i in range(n):
    m=float(input("enter the real part: "))
    n=float(input("enter the img part:  "))
    complex_list.append(Complex(m,n))

sum_series=Complex()
for x in complex_list:
    sum_series+=x
print("The sum of given complex numbers is", sum_series)
```

Exp.10 Develop a program that uses class student which prompts the user to enter marks in three subjects and calculates total marks, percentage and displays the score card details using init method.

```
class Student:
    def __init__(self, name=" ", usn=" ", marks=[]):
        self.name=name
        self.usn=usn
        self.marks=list()

    def getMarks(self):
        x=map(int, input("enter marks scored in 3 Subjects  separated by space: ").split())
        self.marks += x

    def getDetails(self):
        self.name=input("enter name: ")
        self.usn=input("enter usn: ")

    def display(self):
        print("name: ", self.name)
```

```
        print("usn: ", self.usn)
        print("marks: ", self.marks)

        total=0
        for x in self.marks:
            total+=x
        print("Total Marks: ", total, "\tPercentage: ", round(total/3,4), "%")

x=Student()
x.getDetails()
x.getMarks()
x.display()
```