1(a).py > ...

```python
'''# 1(a) 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.'''

#CODE:-
Name = input("Enter your Name: ")
USN = input("Enter your USN: ")
Marks1 = int(input("Enter your Marks 1: "))
Marks2 = int(input("Enter your Marks 2: "))
Marks3 = int(input("Enter your Marks 3: "))

Total = Marks1 + Marks2 + Marks3
Percentage = (Total / 300) * 100

print("Your Name is:", Name)
print("Your USN is:", USN)
print("Your Marks 1:", Marks1)
print("Your Marks 2:", Marks2)
print("Your Marks 3:", Marks3)
print("Your Total Marks are:", Total)
print("Your Percentage is:", Percentage)

if Percentage > 35:
    print("Congrats, you have passed the exam!")
else:
    print("You failed in the exam!")

```

1(b).py > ...

```python
'# 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.'

#CODE:-
import datetime

def calculate_age(birth_year):
    current_year = datetime.datetime.now().year
    return current_year - birth_year

name = input("Enter the name of the person: ")
birth_year = int(input("Enter the year of birth: "))

age = calculate_age(birth_year)

print(f"The age of {name} is {age} years.")
if age >= 60:
    print(f"{name} is a senior citizen.")
else:
    print(f"{name} is not a senior citizen.")

```

2(a).py > ...

```python
'# 2(a) Develop a program to generate Fibonacci sequences of length (N). Read N from the console.'

#CODE:-
n = int(input("Enter the number of Fibonacci series terms required: "))

a, b = 0, 1
count = 0

if n <= 0:
    print("Please enter a positive number")
elif n == 1:
    print("Fibonacci sequence up to", n, "is", a)
else:
    print("Fibonacci sequence:")
    while count < n:
        print(a)
        c = a + b
        a = b
        b = c
        count += 1
        print()
```

2(b).py > ...

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

#CODE:-
def fact(n):
    f = 1
    for i in range(1, n+1):
        f = f * i
    return f

def binomial_coefficient(n, r):
    return fact(n) / (fact(r) * fact(n - r))

n = int(input("Enter N: "))
r = int(input("Enter R: "))

binom_coeff = binomial_coefficient(n, r)

print(f"The binomial coefficient ({n} choose {r}) is: {binom_coeff}")

```

3.py > ...

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

#CODE:-
def calculate_mean(numbers):
    return sum(numbers) / len(numbers)

def calculate_variance(numbers):
    mean = calculate_mean(numbers)
    return sum((x - mean) ** 2 for x in numbers) / len(numbers)

def calculate_std_dev(variance):
    return variance ** 0.5

N = int(input("Enter the value of N: "))
numbers = []
for i in range(N):
    num = float(input(f"Enter number {i + 1}: "))
    numbers.append(num)

mean = calculate_mean(numbers)
variance = calculate_variance(numbers)
std_dev = calculate_std_dev(variance)

print(f"Mean: {mean}")
print(f"Variance: {variance}")
print(f"Standard Deviation: {std_dev}")

```

4.py > ...

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

#CODE:-
print ('Enter a Multidigit number: ')
num = input()
dict = {}
for x in num:
    if x in dict:
        dict[x]+= 1
    else:
        dict[x] = 1
print(str(dict))
```

5.py > count_word_frequency

```python
'''# 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]]'''

#CODE:-
import operator
def count_word_frequency(file_name):
    with open(file_name, "r") as file:
        word_count = {}
        for line in file:
            words = line.split()
            for word in words:
                if word in word_count:
                    word_count[word] += 1
                else:
                    word_count[word] = 1
    return word_count

# Function to get the top N most frequent words
def get_top_words(word_count, top_n):
    sorted_word_count = sorted(word_count.items(), key=operator.itemgetter(1), reverse=True)
    return dict(sorted_word_count[:top_n])

# Main code
file_name = input("Enter the file name: ")

# Count word frequencies
word_count = count_word_frequency(file_name)
print("All word occurrences before sorting:")
print(word_count)

# Get top 10 most frequent words
top_words = get_top_words(word_count, 10)
print("Top 10 words sorted by frequency:")
print(top_words)

```

6.py > ...

```python
'''# 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()].'''

#CODE:-
input_file = input("Enter the file name: ")

with open(input_file, "r") as file:
    lines = file.readlines()

words = []

for line in lines:
    for word in line.split():
        word = word.strip(",. ")
        words.append(word)

words.sort()

print("Sorted list:")
for word in words:
    print(word)

output_file = "final_sorted_list.txt"
with open(output_file, "w") as file:
    for word in words:
        file.write(word + "\n")

```

7.py > ...

```python
'''# 7) Create a program multiplicationTable.py that takes a number N from the command line and
creates an NxN multiplication table in an Excel spreadsheet.'''

#CODE:-
import openpyxl
import sys
N = int(sys.argv[1])
wb = openpyxl.Workbook()
sheet = wb.active
sheet.title = "Multiplication Table"
row = 1
for i in range(1, N + 1):
    for j in range(1, 11):
        res = f"{i}*{j} = {i*j}"
        sheet.cell(row=row, column=1 ,value = res)
        row+=1

wb.save(f"Nishan.xlsx")
print("Multiplication table saved successfully!")
```

8.py > ...

```python
'''# 8) Consider a studData.csv file. File has the USN, Name and CGPA of the students in the class.
Develop a program to find the first topper of the class.'''

#CODE:-
import csv
with open("Book1.csv") as file:
    csvReader = csv.reader(file)
    headerRow = next(csvReader)
    highest = 0
    topper = None
    for row in csvReader:
        USN,Name,CGPA = row[0],row[1],float(row[2])
        if CGPA>highest:
            highest = CGPA
            topper = (USN,Name,CGPA)
if topper:
    print(f"Topper of the class:\n USN:{topper[0]}\n Name:{topper[1]}\n CGPA:{topper[2]}")
else:
    print("No data found.")
```

9.py > ...

```
'''# 9) Define a function which takes TWO objects representing complex numbers and returns a new complex number with an
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.'''

#CODE:-
class Complex():
    def initComplex(self):
        self.realPart= int(input("Enter the Real Part: "))
        self.imgPart = int(input("Enter the Imaginary Part: "))
    def display(self):
        print(self.realPart, "+", self.imgPart,"i", sep="")
    def sum(self, c1, c2):
        self.realPart = c1.realPart + c2.realPart
        self.imgPart = c1.imgPart + c2.imgPart

c1= Complex()
c2= Complex()
c3= Complex()

print("Enter the first complex number")
c1.initComplex()
print('First Complex Number: ', end="")
c1.display()

print("Enter the second complex number")
c2.initComplex()
print('Second Complex Number: ', end="")
c2.display()

print("Sum of two complex number is " , end="")
c3.sum(c1,c2)
c3.display()
```

10.py > ...

```
'''# 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. [Hint: Use list to store the marks in three subjects and total marks. Use __init__()
method to initialize name, USN and the lists to store marks and total, Use getMarks() method to read marks into the list, and display()
method to display the score card details.]'''

# CODE:-

class Student:
    def __init__(self, Name, USN):
        self.name = Name
        self.usn = USN
        self.marks = [0, 0, 0]
        self.total = 0

    def getMarks(self):
        print("Enter the marks of three subjects: ")
        for i in range(3):
            self.marks[i] = int(input(f"Subject {i + 1}: "))

    def calcTotal(self):
        self.total = sum(self.marks)

    def calcPercentage(self):
        return self.total / 3

    def display(self):
        print("\nScore Card")
        print("Name: ", self.name)
        print("USN: ", self.usn)
        print("Marks in Subjects: ", self.marks)
        print("Total Marks: ", self.total)
        print("Percentage: ", f"{self.calcPercentage():.2f}%")

# Main execution
name = input("Enter student Name: ")
usn = input("Enter student USN: ")
student = Student(name, usn)
student.getMarks()
student.calcTotal()
student.display()
```