1.a)Develop a python program to read 2 numbers from the keyboard and perform the basic arithmetic operations based on the choice. (1-Add, 2-Subtract, 3-Multiply, 4-Divide).

```
num1 = float(input("Enter the first number: "))
num2 = float(input("Enter the second number: "))
print("\nSelect an operation:")
print("1 - Add")
print("2 - Subtract")
print("3 - Multiply")
print("4 - Divide")
choice = int(input("Enter your choice (1-4): "))
if choice == 1:
    print("Result:", num1 + num2)
elif choice == 2:
    print("Result:", num1 - num2)
elif choice == 3:
    print("Result:", num1 * num2)
elif choice == 4:
    if num2 != 0:
        print("Result:", num1 / num2)
    else:
        print("Error: Division by zero is not allowed.")
else:
    print("Invalid choice! Please enter a number between 1 and 4.")
```

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.

```
from datetime import datetime
name = input("Enter your name: ")
year_of_birth = int(input("Enter your year of birth: "))
current_year = datetime.now().year
age = current_year - year_of_birth
print(f"\nHello {name}, you are {age} years old.")
if age >= 60:
    print("You are a senior citizen.")
else:
    print("You are not a senior citizen.")
```

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

```
N = int(input("Enter the length of Fibonacci sequence (N): "))
print(f"\nFibonacci sequence of length {N}:")
for i in range(N):
    print(a, end=" ")
    a, b = b, a + b
```

2b) Write a python program to create a list and perform the following operations

- **Inserting an element**
- **Removing an element**
- **Appending an element**
- **Displaying the length of the list**
- **Popping an element**
- **Clearing the list**

```
my_list = [10, 20, 30, 40]
print("Initial List:", my_list)
my_list.insert(2, 25)
print("After Inserting 25 at index 2:", my_list)
my_list.remove(40)
print("After Removing 40:", my_list)
my_list.append(50)
print("After Appending 50:", my_list)
print("Length of the list:", len(my_list))
popped_element = my_list.pop()
print("Popped Element:", popped_element)
print("List after Popping:", my_list)
my_list.clear()
print("After Clearing the list:", my_list)
```

3. a. 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
L = []
n = int(input("Enter the number of elements: "))
for i in range(n):
    val = int(input("Enter the element : "))
    L.append(val)
print('The length of list1 is', len(L))
print('List Contents', L)
mean = sum(L) / n
total = 0
for elem in L:
    total += (elem - mean) ** 2
variance = total / n
stddev = math.sqrt(variance)
print("Mean =", mean)
print("Variance =", variance)
print("Standard Deviation =", stddev)
```

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

```
n = input("Enter a number : ")
freq = {}
for i in n:
    if i in freq:
        freq[i] += 1
    else:
        freq[i] = 1
 # printing result
print("Count of all characters is :", freq)
```

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

```
import operator
fname = input('Enter the file name: ')
try:
    fhand = open(fname)
    counts = dict()
    for line in fhand:
        words = line.split()
        for word in words:
            if word in counts:
                counts[word] += 1
            else:
                counts[word] = 1
     counts = sorted(counts.items(), key=operator.itemgetter(1), reverse=True)
    for i in range(10):
        print(counts[i])
except:
    print('File cannot be opened:', fname)
```

5. Develop a program to read 6 subject marks from the keyboard for a student. Generate a report that displays the marks from the highest to the lowest score attained by the student. [Read the marks into a 1-Dimesional array and sort using the Bubble Sort technique].# Program: Sort 6 subject marks using Bubble Sort (Descending Order)

```
marks = []
print("Enter marks for 6 subjects:")
for i in range(6):
    mark = int(input(f"Enter mark for subject {i+1}: "))
    marks.append(mark)
```

```
n = len(marks)
for i in range(n - 1):
    for j in range(n - i - 1):
        if marks[j] < marks[j + 1]:  # Swap if next mark is higher
            marks[j], marks[j + 1] = marks[j + 1], marks[j]

print("\n--- Student Report ---")
print("Marks in descending order:")
for mark in marks:
    print(mark)
```

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()].

```
f = open("input.txt")
words = []
for line in f:
    temp = line.split()
    for i in temp:
      words.append(i)
f.close()
words.sort()
outfile = open("output.txt", "w")
for i in words:
    outfile.writelines(i)
    outfile.writelines(" ")
outfile.close()
```

INPUT: input.txt

```
Bapuji Educational Association (BEA) is a conglomerate of over 50
educational institutions across the city of Davangere. The Association was
established in the year 1958 with the inception of a first grade college in
Davangere. Two medical colleges, two dental colleges, an engineering college
- Bapuji Institute of Engineering & Technology (BIET) and numerous other
colleges are associated with association. The Bapuji Educational Association
is one of the oldest and most prestigious educational associations in
Karnataka. Today the association has grown to be a big tree, like the banyan,
with all its twigs and branches. It runs schools and colleges right from
Nursery to Post Graduate courses, from Diploma to Engineering, Nursing to
Medical.
```

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

```
def DivExp(a, b):
    try:
        assert a > 0, "a should be greater than 0."
         # Raise an exception if b is 0
        if b == 0:
            raise ZeroDivisionError("Division by zero is  not allowed.")

        # Perform the division
        c = a / b
        print("The result of division is:", c)

    except Exception as e:
        print(e)
a = int(input('Enter the value of a:'))
b = int(input('Enter the value of b:'))
DivExp(a,b)
```

8.Define a function that takes TWO objects representing complex numbers and returns a new complex number with the sum 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, imaginary):
        self.real = real
        self.imaginary = imaginary

    def _ _add_ _(self, other):
        return Complex(self.real + other.real, self.imaginary + other.imaginary)

    def __str__(self):
        return f"{self.real} + {self.imaginary}i"

def read(n):
    complex_numbers = []
    for i in range(n):
          real = float(input(f"Enter the real part of  complex number {i + 1}: "))
```

```
            imaginary = float(input(f"Enter the imaginary part of complex number {i + 1}: "))
            complex_numbers.append(Complex(real, imaginary))
    return complex_numbers

def sum(complex_numbers):
    total = Complex(0, 0)
    for cn in complex_numbers:
        total = total + cn
    return total
n = int(input("Enter the number of complex numbers (N >= 2): "))
if n < 2:
    print("N must be at least 2.")
    exit()
complex_numbers = read(n)
total = sum(complex_numbers)
print("The sum of the complex numbers is:", total)
```

9. Text Analysis Tool: Build a tool that analyses a paragraph: frequency of each word, longest word,number of sentences, etc.

```
import string
def text_analysis(paragraph):
    # --- Preprocessing ---
    # Remove punctuation for word analysis
    clean_text = paragraph.translate(str.maketrans('', '', string.punctuation))
    words = clean_text.split()

    # --- Word frequency ---
    word_freq = {}
    for word in words:
        word_lower = word.lower()  # ignore case
        word_freq[word_lower] = word_freq.get(word_lower, 0) + 1

    # --- Longest word ---
    longest_word = max(words, key=len)

    # --- Number of sentences ---
    sentences = paragraph.split('.')
    # Remove empty strings caused by trailing '.'
    num_sentences = len([s for s in sentences if s.strip() != ""])
```

```
# --- Report ---
    print("----- TEXT ANALYSIS REPORT -----")
    print(f"Total words: {len(words)}")
    print(f"Unique words: {len(word_freq)}")
    print(f"Number of sentences: {num_sentences}")
    print(f"Longest word: '{longest_word}' ({len(longest_word)} letters)")
    print("\nWord Frequencies:")
    for word, freq in sorted(word_freq.items(), key=lambda x: x[1], reverse=True):
        print(f"{word} : {freq}")


# --- Example usage ---
paragraph = """Artificial intelligence is the simulation of human intelligence processes by machines.
AI applications include natural language processing, speech recognition, and machine vision.
The field of AI is growing rapidly."""

text_analysis(paragraph)
```

11.Develop Student Grade Tracker: Accept multiple students' names and marks. Store them in a list of tuples or dictionaries. Display summary reports (average, topper, etc.).

```
def input_students():
    students = []
    while True:
        name = input("Enter student name (or type 'done' to finish): ")
        if name.lower() == "done":
            break
        try:
            marks = float(input(f"Enter marks for {name}: "))
            students.append({"name": name, "marks": marks})
        except ValueError:
            print("□  Please enter a valid number for marks.")
    return students

def summary_report(students):
    if not students:
        print("\nNo student data available.")
        return

    total_students = len(students)
    total_marks = sum(s["marks"] for s in students)
    avg_marks = total_marks / total_students
    topper = max(students, key=lambda x: x["marks"])
    lowest = min(students, key=lambda x: x["marks"])

    print("\n----- STUDENT GRADE REPORT -----")
    print(f"Total Students   : {total_students}")
    print(f"Average Marks    : {avg_marks:.2f}")
    print(f"Topper           : {topper['name']} ({topper['marks']})")
    print(f"Lowest Scorer    : {lowest['name']} ({lowest['marks']})")
    print("\nIndividual Records:")
    for s in students:
        print(f"{s['name']} → {s['marks']}")

def main():
    print("=== Student Grade Tracker ===")
    students = input_students()
    summary_report(students)

if __name__ == "__main__":
    main()
```

12. Develop a program to display contents of a folder recursively (Directory) having sub-folders and files (name and type).

```
import os

def display_directory_contents(path, indent=0):
    try:
        items = os.listdir(path)  # List all files and subfolders
    except PermissionError:
        print(" " * indent + f"[ACCESS DENIED] {path}")
        return

    for item in items:
        full_path = os.path.join(path, item)
        if os.path.isdir(full_path):
            print(" " * indent + f"[Folder] {item}")
            display_directory_contents(full_path, indent + 4)  # Recursive call
        else:
            print(" " * indent + f"[File]   {item}")

def main():
    folder = input("Enter the directory path to explore: ")
    if os.path.exists(folder):
        print(f"\n--- Contents of '{folder}' ---")
        display_directory_contents(folder)
    else:
        print("□  The specified path does not exist.")

if __name__ == "__main__":
    main()
```