Question 1

Write a python program to sum of the first n positive integers.

```
n = int(input("Enter a Number: "))
i = 1
Sum = 0

if n >= 0:
    while i <= n:
        Sum += i
        i += 1
    print(f"The sum of {n} positive integers is {Sum}")
else:
    print("Sorry! The number you have entered is negative.")
```

Question 2

Write a Python program to count occurrences of a substring in a string

```
Main_Str = input("Enter the main String :  ")
Sub_Str =  input("Enter Sub String :")

count = 0
Sub_Str_len = len(Sub_Str)

for i in range(len(Main_Str)-Sub_Str_len+1):
    if Main_Str[i:i+Sub_Str_len]==Sub_Str:
        count+=1

print(f"The Substring {Sub_Str } appears {count} times in {Main_Str}")
```

```
Enter the main String :    Bhaumik
Enter Sub String : aum

The Substring aum appears 1 times in Bhaumik
```

Question 3

Write a Python program to count the occurrences of each word in a given sentence.

```python
sentence = input("Enter a sentence: ").lower()

word = ""
words = []

for char in sentence:
    if char != " ":
        word += char
    else:
        if word:
            words.append(word)
            word = ""
if word:
    words.append(word)

word_count = {}
for w in words:
    if w in word_count:
        word_count[w] += 1
    else:
        word_count[w] = 1

print("\nWord Occurrences:")
for word, count in word_count.items():
    print(f"{word}: {count}")
```

```
Enter a sentence:  mY NAME IS abc


Word Occurrences:
my: 1
name: 1
is: 1
abc: 1
```

Question 4

Write a Python program to get a single string from two given strings, separated by a space and swap the first two characters of each string.

```
String1 = input("Enter your 1st string : ")
String2 = input("Enter your 2nd string: ")

if len(String1)<2 or len(String2)<2:
    print("Atleast 2 characters should be used for both the strings")
else:
    new_string1 = String2[:2] + String1[2:]
    new_string2 = String1[:2] + String2[2:]

    final_Str =  new_string1 + " " + new_string2
    print("Final String : ", final_Str)
```

```
Enter your 1st string :  good
Enter your 2nd string:  morning

Final String :  mood  gorning
```

Question 5

Write a Python program to add 'ing' at the end of a given string (length should be at least 3). If the given string already ends with 'ing' then add 'ly' instead If the string length of the given string is less than 3, leave it unchanged

```
Str = input("Enter a word ")
if len(Str)<3:
    New_Str = Str
elif Str[-3:]=='ing':
    New_Str = Str[:-3] + 'ly'
else:
    New_Str = Str + 'ing'
print("New word", New_Str)
```

```
Enter a word  jogging

New word joggly
```

Question 6

Write a Python program to find the first appearance of the substring 'not' and'poor' from a given string, if 'not' follows the 'poor', replace the whole 'not'...'poor' substring with 'good'. Return the resulting string

```
sentence = input("Enter a sentence: ")


not_idx = sentence.find('not')
poor_idx = sentence.find('poor')
if not_idx != -1 and poor_idx != -1 and not_idx < poor_idx:
     sentence = sentence[:not_idx] + 'good' + sentence[poor_idx+4:]
     print(sentence)


Enter a sentence:  food is not poor

food is good
```

Question 7

Program to find Greatest Common Divisor of two numbers. For example, the GCD of 20 and 28 is 4 and the GCD of 98 and 56 is 14.

```
Num1 =  int(input("Enter the 1st number: "))
Num2 = int(input("Enter the 2nd nunber: "))
while Num2!=0:
    temp = Num2
    Num2 = Num1%Num2
    Num1=temp
print("Greatest Common Divisor is : ",Num1)

Enter the 1st number:  35
Enter the 2nd nunber:  34

Greatest Common Divisor is :  1
```

Question 8

Write a Python program to check whether a list contains a sublist.

```
l1 = [10,12,14,16,18,20,22,24]
Sub_l1 = [14,16,18,20]

for i in range(len(l1)-len(Sub_l1)+1):
    if l1[i:i+len(Sub_l1)]==Sub_l1:
        print("Sublist is present in main list")
        break
else:
    print("Sublist not available")
```

```
Sublist is present in main list
```

Question 9

Write a Python program to find the second smallest number in a list.

```
input_string = input("Enter numbers separated by commas (e.g.,
4,2,7,1): ")

temp = ''
numbers = []
for char in input_string:
    if char != ',':
        temp += char
    else:
        numbers.append(int(temp))
        temp = ''

if temp:
    numbers.append(int(temp))


unique_numbers = []
for num in numbers:
    if num not in unique_numbers:
        unique_numbers.append(num)

if len(unique_numbers) < 2:
    print("There is no second smallest number.")
else:

    smallest = second_smallest = float('inf')
    for num in unique_numbers:
        if num < smallest:
            second_smallest = smallest
            smallest = num
        elif num < second_smallest:
            second_smallest = num

    print("The second smallest number is:", second_smallest)
```

```
Enter numbers separated by commas (e.g., 4,2,7,1):  1,2,3,1,2,4,5,3

The second smallest number is: 2
```

Question 10

Write a Python program to get unique values from a list.

```
List = [3,3,5,7,1,3,3,4,5,6,6]
unique = []

for num in List:
    if num not in unique:
        unique.append(num)
print(unique)
```

```
[3, 5, 7, 1, 4, 6]
```

Question 11

Write a Python program to unzip a list of tuples into individual lists

```
list_of_tuples = [(1,2),(3,4),(5,6),(7,8),(9,10)]

LST1 = []
LST2 = []

for i in range (len(list_of_tuples)):
    LST1.append(list_of_tuples[i][0])
    LST2.append(list_of_tuples[i][1])

print("List no:1(odd)= ", LST1)
print("List no:2(Even)= ", LST2)
```

```
List no:1(odd)=  [1, 3, 5, 7, 9]
List no:2(Even)=  [2, 4, 6, 8, 10]
```

Question 12

Write a Python program to convert a list of tuples into a dictionary

```
tup_list = [('Name', 'Shiv'),('Age' , 20)]

dictionary = dict(tup_list)

print(dictionary)
```

```
{'Name': 'Shiv', 'Age': 20}
```

Question 13

Write a Python program to sort a dictionary (ascending /descending) by value

```
my_dict = {'Ram': 25, 'Krishna': 20, 'shiv': 30, 'Raj': 13}

pairs = []
for key in my_dict:
    pairs.append([key, my_dict[key]])

for i in range(len(pairs)):
    min_idx = i
    for j in range(i+1, len(pairs)):
        if pairs[j][1] < pairs[min_idx][1]:
            min_idx = j
    pairs[i], pairs[min_idx] = pairs[min_idx], pairs[i]

asc_dict = {}
for item in pairs:
    asc_dict[item[0]] = item[1]

print("Ascending order:", asc_dict)

for i in range(len(pairs)):
    max_idx = i
    for j in range(i+1, len(pairs)):
        if pairs[j][1] > pairs[max_idx][1]:
            max_idx = j
    pairs[i], pairs[max_idx] = pairs[max_idx], pairs[i]

desc_dict = {}
for item in pairs:
    desc_dict[item[0]] = item[1]
```

```
print("Descending order:", desc_dict)

Ascending order: {'Raj': 13, 'Krishna': 20, 'Ram': 25, 'shiv': 30}
Descending order: {'shiv': 30, 'Ram': 25, 'Krishna': 20, 'Raj': 13}
```

Question 14

Write a Python program to find the highest 3 values in a dictionary

```
my_dict = {'ten': 10, 'fourteen': 14, 'Eight': 8, 'fourty': 40, 'Six':
6}

pairs = []
for key in my_dict:
    pairs.append([key, my_dict[key]])

top_values = []
for _ in range(3):
    max_pair = ['', float('-inf')]
    for pair in pairs:
        if pair[1] > max_pair[1]:
            max_pair = pair
    top_values.append(max_pair)
    pairs.remove(max_pair)

for key, value in top_values:
    print(f"{key}: {value}")
```

Question 15

Given a number n, write a python program to make and print the list of Fibonacci

series up to n. Input : n=7 Hint : first 7 numbers in the series Expected output :

First few Fibonacci numbers are 0, 1, 1, 2, 3, 5, 8, 13

```
n1 = int(input("Enter number for the series  : "))

a = 0
b = 1
print("The fibonacci series : ")
```

```
i = 0
while i<n1:
    print(a)
    c = a+b
    a=b
    b=c
    i+=1
```

```
Enter number for the series  :  7

The fibonacci series :
0
1
1
2
3
5
8
```

Question 16

Counting the frequencies in a list using a dictionary in Python. Input : [1, 1, 1, 5, 5, 3, 1, 3, 3, 1,4, 4, 4, 2, 2, 2, 2]

Expected output : 1 : 5 , 2 : 4 , 3 : 3 , 4 : 3 , 5 : 2

```
lst1 = [2,2,2,1,3,4,5,6,7,7,7,7,8,9,10,10]

freq = {}

for num in lst1:
    if num in freq:
        freq[num]+=1
    else:
        freq[num] = 1
for key in freq:
    print(f"{key} : {freq[key]}")
```

```
2 : 3
1 : 1
3 : 1
4 : 1
5 : 1
6 : 1
7 : 4
8 : 1
9 : 1
10 : 2
```

Question 17

Write a python program using function to find the sum of odd series and even series Odd series: 12/ 1! +32/ 3! + 52/ 5! +……n Even series: 22/ 2! + 42/ 4! + 62/ 6!+……n

```
def Factorial(num):
    fact = 1
    for i in range(1, num+1):
        fact*=i
    return fact

def Odd_series(n):
    Sum_odd = 0
    for i in range(1, 2*n,2):
        Sum_odd+= (i**2)/Factorial(i)
    return Sum_odd

def Even_series(n):
    Sum_even = 0
    for i in range(2, 2*n+1, 2):
        Sum_even+= (i**2)/Factorial(i)
    return Sum_even

N1 = int(input("Enter a value: "))

odd = Odd_series(N1)
even = Even_series(N1)

print("The sum of Odd series: ", odd)
print("The sum of Even series", even)
```

```
Enter a value:  100

The sum of Odd series:  2.7182818284590455
The sum of Even series 2.7182818284590446
```

Question 18

Python Program to Find Factorial of Number Using Recursion

```
def factorial(n):
    if n==0 or n==1:
        return 1
    else:
        return n * factorial(n-1)
```

```
NUM1 = int(input("Enter a value : "))

if NUM<0:
    print("Factorial does not apply for the negative numbers!")
else:
    print(f"The factorial of {NUM1} is {factorial(NUM1)}")

Enter a value :  6

The factorial of 6 is 720
```

Question 19

Write a Python function that takes a list and returns a new list with unique elements of the first list.

```
def Unique_list(List):
    unique = []
    for item in List:
        if item not in unique:
            unique.append(item)
    return unique

my_list = [1,1,'Good','Good',3,4,5,5,7,7,4]
final = Unique_list(my_list)
print("Unique elements", final)


Unique elements [1, 'Good', 3, 4, 5, 7]
```

Question 20 (Mini project)

Mini project : Problem Statement : Password Generator

Make a program to generate a strong password using the input given by the user.

To generate a password, randomly take some words from the user input and then

include numbers, special characters and capital letters to generate the password.

Also, keep a check that password length is more than 8 characters.

Note: Include Exception handling wherever required. Also, make a ‘User’ class

and store the details like user id, name and password of each user as a tuple.

```
class User:
    def __init__(self,user_id, name, password):
        self.details = (user_id, name, password)
    def display(self):
        print(f"User ID : {self.details[0]}")
        print(f"Name : {self.details[1]}")
        print(f"Password: {self.details[2]}")

def generate_password(user_input, numbers, special_Char):
    try:
        if not user_input.strip():
            raise ValueError("Word input cannot be empty.")
        if not numbers.strip():
            raise ValueError("Number input cannot be empty.")
        if not special_Char.strip():
            raise ValueError("Special characters input cannot be
empty.")
        words = user_input.strip().split()
        if len(words)<2:
            raise ValueError("Please enter atleast two words.")
```

```python
        word1 = words[0].capitalize()
        word2 = words[1].capitalize()
        ## We combine everything
        password = word1 + word2 + numbers + special_Char
        ## Checking for mininum criteria of 8 characters
        if len(password) < 8:
            raise Exception("The password is too short. Should be 
atleast of 8 characters")
        return password
    except Exception as e:
        print("Error:" , e)
        return None

try:
    user_id = input("Enter user ID: ")
    name = input("Enter your name: ")
    user_input = input("Enter a few words (space): ")
    numbers = input("Enter a few numbers to include in your password: 
")
    special_Char = input("Enter special characters to include in your 
password: ")

    password = generate_password(user_input, numbers, special_Char)
    if password:
        new_user = User(user_id, name, password)
        print("\nUser Details:")
        new_user.display()
    else:
        print("Password generation failed.")

except Exception as e:
    print("Unexpected error:", e)
```

```
Enter user ID:  abc\
Enter your name:  xyz
Enter a few words (space):  ABC XYZ
Enter a few numbers to include in your password:  123123
Enter special characters to include in your password:  !


User Details:
User ID : abc\
Name : xyz
Password: AbcXyz123123!
```