==
===

1. Write a Python program that swaps the values of two variables without using a temporary variable, only using arithmetic operations.
Sample Input A=10 ,B= 61 Output : A=61,B= 10

```
a = 10
b = 61
a = a + b
b = a - b
a = a - b
print("A =", a)
print("B =", b)
```

==
===

2. Write a python program that takes two integers as input and performs bitwise AND, OR, XOR, left shift, and right shift operations on them. Print the results in binary format

```
# Take two integers as input
num1 = int(input("Enter the first integer: "))
num2 = int(input("Enter the second integer: "))
# Perform bitwise AND operation
result_and = num1 & num2
# Perform bitwise OR operation
result_or = num1 | num2
# Perform bitwise XOR operation
result_xor = num1 ^ num2
# Perform left shift operation
result_left_shift = num1 << num2
# Perform right shift operation
result_right_shift = num1 >> num2
# Print results in binary format
print("Bitwise AND:", bin(result_and))print("Bitwise OR:", bin(result_or))
print("Bitwise XOR:", bin(result_xor))
print("Left Shift:", bin(result_left_shift))
print("Right Shift:", bin(result_right_shift))
```

==
===

3. Write a Python program that takes two integers as input and performs the following operations:
1. Sum of the two integers
2. Difference of the two integers
3. Product of the two integers
4. Quotient (integer division) of the two integers
5. Remainder (modulus) of the two integers

```
def get_input():
num1 = int(input("Enter first integer: "))
num2 = int(input("Enter second integer: "))
return num1, num2
def perform_operations(num1, num2):
while True:
operation = input("Enter operation (+, -, *, /, %, q to quit): ")
```

```
if operation == "+":
print(num1 + num2)
elif operation == "-":
print(num1 - num2)
elif operation == "*":
print(num1 * num2)
elif operation == "/":
if num2 != 0:
print(num1 // num2)
else:
print("Error: Division by zero")
elif operation == "%":
if num2 != 0:
print(num1 % num2)
else:
print("Error: Division by zero")
elif operation == "q":
breakelse:
print("Invalid operation. Please try again.")
num1, num2 = get_input()
perform_operations(num1, num2)
```

===

4. Write a Python program that takes a temperature in Fahrenheit as input and converts it to Celsius using the formula: C=5/9(F−32). Display the converted temperature.

```
fahrenheit = float(input("Enter temperature in Fahrenheit: "))
celsius = (5/9) * (fahrenheit - 32)
print(f"{fahrenheit} Fahrenheit is equal to {celsius:.2f} Celsius")
```

===

5. Write a Python program for counting the digits in a number.

Input
Enter number:89589
Output: 5

```
def count_digits():
number = int(input("Enter a number: "))
count = 0
while number != 0:
number = number // 10
count += 1
print(f"The number of digits in {number} is {count}.")
count_digits()
```

===

6. Write a python program for the following Expressions

i) 24 // 6 % 3
ii) float(4 + int(2.39) % 2)
iii) 2 ** 2 ** 3

```
print(24 // 6 % 3)
print(float(4 + int(2.39) % 2))
print(2 ** 2 **
3)==================================================================
====
```

7. Write a python program to Print the following pattern

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

```
n = 5
for i in range(1, n+1):
for j in range(1, i+1):
print(j, end=" ")
print()
```

8. Write a Python program to find the maximum of three numbers entered by the user.

Test case
Input : 3 7 2
Output : 7

```
num1 = float(input("Enter the first number: "))
num2 = float(input("Enter the second number: "))
num3 = float(input("Enter the third number: "))
maximum_number = max(num1, num2, num3)
print("The maximum of the three numbers is:", maximum_number)
```

9. Write a Python program that takes input for marks obtained in different subjects from the user and calculates the total marks, percentage, and grade based on predefined grade criteria.

```
def calculate_grade(percentage):
if percentage >= 90:
return 'A'
elif 80 <= percentage < 90:
return 'B'
elif 70 <= percentage < 80:
return 'C'
elif 60 <= percentage < 70:
return 'D'else:
return 'F'
num_subjects = int(input("Enter the number of subjects: "))
marks = []
for i in range(num_subjects):
subject_marks = float(input(f"Enter marks obtained in subject {i+1}: "))
marks.append(subject_marks)
total_marks = sum(marks)
percentage = (total_marks / (num_subjects * 100)) * 100
grade = calculate_grade(percentage)
print(f"\nTotal Marks: {total_marks}")
print(f"Percentage: {percentage:.2f}%")
print(f"Grade: {grade}")
```

==

1. Write a Python program to find the Nth Fibonacci number. The program should take the value of N as input

n=8

Output 0 1 1 2 3 5 8 13

```
N = int(input("Enter the value of N to find the Nth Fibonacci number: "))
if N <= 0:
print("Invalid input. Please enter a positive integer.")
else:
fib1, fib2 = 0, 1
for i in range(2, N):
fib_sum = fib1 + fib2
fib1, fib2 = fib2, fib_sum
print(f"The {N}th Fibonacci number is:", fib2 if N > 1 else fib1)
```

==

2. Write a python program for sum of squares of first N natural numbers.Given a positive integer N, the task is to find 12 + 22 + 32 + ... + N2

```
def sum_of_squares(n):
return (n * (n + 1) * (2 * n + 1)) // 6
n = int(input("Enter a positive integer: "))
print("Sum of squares of the first", n, "natural numbers:", sum_of_squares(n))
```

==

3. Write a Python program that accepts a string and calculates the number of digits and letters.

Sample Data : Python 3.2

Expected Output :

Letters 6

Digits 2

```
def count_letters_digits(string):
num_letters = 0
num_digits = 0
for char in string:
if char.isalpha():
num_letters += 1elif char.isdigit():
num_digits += 1
return num_letters, num_digits
n = input("Enter a string: ")
letters, digits = count_letters_digits(n)
print("Letters:", letters)
print("Digits:", digits)
```

==

4. Write a Python program to reverse the order of the items in the array.

Sample Output

Original array: array('i', [1, 3, 5, 3, 7, 1, 9, 3])

Reverse the order of the items:

array('i', [3, 9, 1, 7, 3, 5, 3, 1])

```
import array as arr
arr = arr.array('i', [1, 3, 5, 3, 7, 1, 9, 3])
print("Original array:", arr)
arr.reverse()
print("Reversed array:", arr)
```

===
===

5. Write a python program to merge 2 lists and also print the list in the Ascending order using built in function

input:
a=[1,2,7,4,5]
b=[11,10,13]
Output:
c=[1,2,7,4,5,11,10,13]
Z= [1,2,4,5,7,10,11,13]

```
# Initialize the lists
a = [1, 2, 7, 4, 5]
b = [11, 10, 13]
# Merge the lists
c = a + b
# Sort the merged list in ascending orderZ = sorted(c)
# Print the merged and sorted list
print("Merged list:", c)
print("Sorted list:", Z)
```

===
===

6. Write a Python program to count the number of even and odd numbers in a series of numbers.

Sample numbers : numbers = (1, 2, 3, 4, 5, 6, 7, 8, 9)
Expected Output :
Number of even numbers : 5
Number of odd numbers : 4

```
def count_even_odd(numbers):
even_count = sum(num % 2 == 0 for num in numbers)
odd_count = len(numbers) - even_count
return even_count, odd_count
numbers = (1, 2, 3, 4, 5, 6, 7, 8, 9)
even_numbers, odd_numbers = count_even_odd(numbers)
print("Number of even numbers:", even_numbers)
print("Number of odd numbers:", odd_numbers)
```

===
===

7. Write a Python program that checks if a specific substring is present in a given string. The program should print "Substring found" if the substring is present, otherwise print "Substring not found"

Sample Input: python is Integrated Language
Substring : Language
Output: Substring found

```
string = "python is Integrated Language"
substring = "language"
```

```
string = string.lower()
substring = substring.lower()
if substring in string:
print("Substring found")
else:
print("Substring not
found")============================================================
========
```

8. Write a program to find the Factorial of a number using recursion.

```
def factorial(n):
if n == 0 :
return 1
else:
return n * factorial(n-1)
num = int(input("Enter a number: "))
print("factorial=",factorial(num))
=================================================================
===
```

9. Write a program to check if a number is a prime number or not using recursion.

```
def is_prime_recursive(number, divisor=2):
if number < 2:
return False
if divisor * divisor > number:
return True
if number % divisor == 0:
return False
return is_prime_recursive(number, divisor + 1)
num = int(input("Enter a number: "))
if is_prime_recursive(num):
print(f"{num} is a prime number.")
else:
print(f"{num} is not a prime number.")
```

(OR)

```
def prime(n,i):
if i==1:
return 1
if n%i==0:
return 0
return prime(n,i-1)
n=int(input("enter a num "))
x=prime(n,n-1)
if x==1:print("prime number ")
if x==0:
print("not prime number ")
prime number
```

General category programs

1. Write a program to print all the Non-Prime numbers between A and B?

Sample Input: A = 12 B = 19

Sample Output:
14, 15, 16, 18

```
a = int(input())
b = int(input())
for x in range (a, b+1):
    if x > 1:
        for i in range (2, x):
            if (x%i)== 0:
                break
        else:
            print (x)
```

2. Find the year of the given Anniversary is leap year or not. If leap year then print the next Anniversary, if not leap year then print the previous Anniversary.

Sample Input:
Enter Date: 04/11/1947 Sample Output:
Given Anniversary Year: Non Leap Year. Anniversary Date: 04/11/1946

```
date = input("Enter the date to be checked: ")
c=date.split("/")
b = list(map(int,c))
input_year=(b[2])

if(input_year%4 == 0):
    if(input_year%100 == 0):
        if(input_year%400 == 0):
            print("%d is Leap Year" %input_year)
        else:
            print("%d is not the Leap Year" %input_year)
    else:
        print("%d is the Leap Year" %input_year)
else:
    print("%d is not the Leap Year" %input_year)

x=input_year%4
if x!=0:
    print("Previous Leap year:", input_year-x)
else:
    print("Next leap year:", input_year+4)
```

3. Write a program to print the given number is Perfect number or not?

Sample Input: Given Number: 6
Sample Output: Its a Perfect Number

```
Number = int(input(" Please Enter any Number: "))
Sum = 0
for i in range(1, Number):
    if(Number % i == 0):
        Sum = Sum + i
if (Sum == Number):
    print(" %d is a Perfect Number" %Number)
else:
    print(" %d is not a Perfect Number" %Number)
```

4. Write a program to generate Pythagorean Triplets for the given limit.

```
Enter upper limit: 10
3 4 5
8 6 10
```

```
A=input("Enter upper limit:")
c=0
m=2
if A.isnumeric():
    x=int(A)
    while(c<x):
        for n in range(1,m+1):
            a=m*m-n*n
            b=2*m*n
            c=m*m+n*n
            if(c>x):
                break
            if(a==0 or b==0 or c==0):
                break
            print(a,b,c)
        m=m+1
else:
    print("invalid input")
```

5. Write a program to find the sum of digits of N digit number (sum should be single digit)

Sample Input: Enter N value : 3 Enter 3 digit number: 143

Sample Output: Sum of 3 digit number: 8

```
num=int(input("Enter the number:"))
Sum=0
temp=num
while temp>0:
    digit=temp%10
    Sum+=digit
    temp=temp//10
print("Sum of Digits:", Sum)
```

6. Program to find whether the given number is Armstrong number or not

Sample Input: Enter number: 153
Sample Output: Given number is Armstrong number

```
num=int(input("Enter the number:"))
Sum=0
temp=num
while temp>0:
    digit=temp%10
    Sum+=digit**3
    temp=temp//10
if Sum==num:
    print("Armstrong Number")
else:
    print("Not a Armstrong Number")
```

7. Program to find whether the given number is Harshad number or not

Sample Input: Enter number: 21
Sample Output: Given number is Harshad number

```
num=int(input("Enter the number:"))
Sum=0
temp=num
while temp>0:
    digit=temp%10
    Sum+=digit
    temp=temp//10
if num%Sum==0:
    print("Harshad Number")
else:
    print("Not a Harshad Number")
```

8. Program to find whether the given number is Happy number or not

Sample Input: Enter number: 19
Sample Output: Given number is happy number

```
def happy(n):
    temp=n
    sum=0
    while temp>0:
        digit=temp%10
        sum=digit**2+sum
        temp=temp//10
    return sum

# Main Program

num=int(input("Enter the number:"))
result=num
while result!=1 and result!=4:
    result=(happy(result))
if result==1:
    print("True")
elif result==4:
    print("False")
```

9. Program to find whether the given number is Tech number or not
Sample Input: Enter number: 3025
Sample Output: Given number is Tech number

```
n = 3025
m = str(n)
a = m[:len(m)//2]
b = m[len(m)//2:]
c = int(a)+int(b)
d = c**2

if(d==n):
    print("Tech number")
else:
    print("Not a Tech number")
```

10. Write a program using function to calculate the simple interest. Suppose the customer is a senior citizen. She is being offered 15 percent rate of interest; he is being offered 12 percent rate of interest for all other customers, the ROI is 10 percent.
Sample Input:
Enter the principal amount: 200000 Enter the no of years: 3
Gender (m/f): m

Is customer senior citizen (y/n): n Sample Output:
Interest: 60000

```
p=int(input("Enter the Principle amount:"))
n=int(input("Enter the number of years:"))
SC=input("Senior Citizen Yes/No:")
G=input("Male/Female:")
if SC=='Y' and G=='M':
   print("SI=",(p*n*12)/100)
elif SC=='Y' and G=='F':
   print("SI=",(p*n*15)/100)
else:
   print("SI=",(p*n*10)/100)
```

11. Find the number of factors for the given number and print the 1st N factors of the given number.

Sample Input: Given number: 100
N: 4
Sample Output: Number of factors = 9
1st 4 factors are: 1, 2, 4, 5

```
x=int(input("Enter the number:"))
y=[]
print("The factors of",x,"are:")
for i in range(1, x):
   if x % i == 0:
      y.append(i)
print(y)
print("Number of factors:", len(y))
n=int(input("Enter N value:"))
if n>len(y):
   print("Invalid")
else:
   print("First", n, "factors:")
   for k in range(0,n):
      print(y[k], end=' ')
```

12. Write a program to print number of factors and to print nth factor of the given number.

Sample Input: Given Number: 100
N = 4
Sample Output:
Number of factors = 9 4th factor of 100 = 5

```
x=int(input("Enter the number:"))
y=[]
print("The factors of",x,"are:")
for i in range(1, x + 1):
    if x % i == 0:
        y.append(i)
print(y)
print("Number of factors:", len(y))
n=int(input("Enter N value:"))
print(n, "th factor is:",y[n-1])
```

13. Write a program to print unique permutations of a given number Sample Input:
Given Number: 143 Sample Output:
Permutations are:
134
143
314
341
413
431

```
import itertools
n=input("Enter the number")
P=list(itertools.permutations(n))
print(*[''.join(p) for p in P])
```

14. Write a program to find the square, cube of the given decimal number Sample Input:
Given Number: 0.6
Sample Output: Square Number: 0.36 Cube Number:0.216

```
import math
num=float(input("Enter the number:"))
print("Square number=",math.pow(num,2))
print("Cube number=",round(math.pow(num,3),3))
```

15. Write a program to convert the Binary to Decimal, Octal Sample Input:
Given Number: 1101 Sample Output:
Decimal Number: 13 Octal: 15

```
num=input("Enter the binary number:")
bin_num="01"
```

```
for i in range(len(num)):
    binary=True
    if num[i] not in bin_num:
        print("Invalid input")
        binary=False
        break
if binary:
    dec_number=int(num,2)
    oct_number=oct(dec_number)
    hexa=hex(dec_number)
    print("Decimal Equivalent=",dec_number)
    print("Octal Equivalent=",oct_number)
    print("Hexa Equivalent=",hexa)
```

16. Add Binary

Given two binary strings a and b, return their sum as a binary string.

- a and b consist only of '0' or '1' characters.
- Each string does not contain leading zeros except for the zero itself.

Test cases:

1.Input: a = "11", b = "1"

1. Output: "100"

```
num1=input("Enter the binary number 1=")
num2=input("Enter the binary number 2=")
sum=bin(int(num1,2)+int(num2,2))
print("Sum of binary numbers: ",sum[2:])
```

17.Python program to find the greatest of three binary numbers

```
a='1101'
b='1110'
c='1111'

bina=int(a,2)
binb=int(b,2)
binc=int(c,2)

if bina>binb and bina>binc:
    print("Greatest is", a)
elif binb>bina and binb>binc:
    print("Greatest is", b)
else:
    print("Greatest is", c)
```

18.Write a program for matrix multiplication?

Sample Input:

Mat1 = $\begin{pmatrix} 1 & 2 \\ 5 & 3 \end{pmatrix}$

Mat2 = $\begin{pmatrix} 2 & 3 \\ 4 & 1 \end{pmatrix}$

Sample Output:

Mat Sum = $\begin{pmatrix} 10 & 5 \\ 22 & 18 \end{pmatrix}$

```
X=[[1,2],
  [5,3]]
Y=[[2,3],
  [4,1]]
result=[[0,0],
      [0,0]]

# iterate through rows of X
for i in range(len(X)):
  # iterate through columns of Y
  for j in range(len(Y[0])):
     # iterate through rows of Y
     for k in range(len(Y)):
        result[i][j] += X[i][k] * Y[k][j]

for r in result:
  print(r)
```

17. Write a program for matrix addition?

Sample Input:

Mat1 = $\begin{pmatrix} 1 & 2 \\ 5 & 3 \end{pmatrix}$

Mat2 = $\begin{pmatrix} 2 & 3 \\ 4 & 1 \end{pmatrix}$

Sample Output:

Mat Sum = $\begin{pmatrix} 3 & 5 \\ 9 & 4 \end{pmatrix}$

```
a=[[1,2],
   [5,3]]
b=[[2,3],
   [4,1]]
c=[[0,0],
   [0,0]]
for i in range(len(a)):
    for j in range(len(b)):
        c[i][j]=a[i][j]+b[i][j]
for i in c:
    print(i)
```

18. Find the LCM and GCD of n numbers

Sample Input:
N value = 2
Number 1 = 16
Number 2 = 20

Sample Output: LCM = 80 GCD = 4

```
n1 = int(input("Enter First number :"))
n2 = int(input("Enter Second number :"))
x = n1
y = n2
while(n2!=0):
   t = n2
   n2 = n1 % n2
   n1 = t
gcd = n1
print("GCD of {0} and {1} = {2}".format(x,y,gcd))
lcm = (x*y)/gcd
print("LCM of {0} and {1} = {2}".format(x,y,lcm))
```

19. Transpose of a matrix

```
matrix = [[4, 6, 7, 8],
          [3, 7, 2, 7],
          [7, 3, 7, 5]]
```

```
a=[[1,2],
   [3,2]]
c=[[0,0],
   [0,0]]
```

```
for i in range(len(a)):
    for j in range(len(a)):
        c[i][j]=a[j][i]
for i in c:
    print(i)
```

20. Program to find row, column and diagonal sum in Matrix

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

o/p:

```
Sum of 1 row: 6
Sum of 2 row: 15
Sum of 3 row: 24
Sum of 1 column: 12
Sum of 2 column: 15
Sum of 3 column: 18
Diagonal sum 15
```

```
#Initialize matrix a
a = [[1, 2, 3],
  [4, 5, 6],
  [7, 8, 9]]

#Calculates number of rows and columns present in given matrix
rows = len(a);
cols = len(a[0]);

#Calculates sum of each row of given matrix
for i in range(0, rows):
  sumRow = 0;
  for j in range(0, cols):
    sumRow = sumRow + a[i][j];
  print("Sum of " + str(i+1) +" row: " + str(sumRow));

#Calculates sum of each column of given matrix
for i in range(0, rows):
  sumCol = 0;
  for j in range(0, cols):
    sumCol = sumCol + a[j][i];
  print("Sum of " + str(i+1) +" column: " + str(sumCol));

#Calculates sum of diagonal
diagonal=0
for k in range(0,rows):
    diagonal=diagonal+a[k][k]
print("Diagonal sum",diagonal)
```

21. Given three integers **M, N** and **K**. Consider a grid of **M * N**, where **mat[i][j] = i * j** (1 based index). The task is to return the **Kth** smallest element in the **M * N** multiplication table.

```
def findKthNumber(m, n, k):

    low = 1
    high = n*m

    while low < high:
        mid = (low + high) // 2
        count = 0
        for i in range(1, m+1):
            count += min(n, mid//i)
        if count < k:
            low = mid + 1
        else:
            high = mid
    return low

#Driver Program
m=3
n=3
k=5
print(findKthNumber(m,n,k))
```

22. Print the sum of boundary elements of a matrix

```
def printBoundary(a, m, n):
    for i in range(m):
        for j in range(n):
            if (i == 0):
                print a[i][j],
            elif (i == m-1):
                print a[i][j],
            elif (j == 0):
                print a[i][j],
            elif (j == n-1):
                print a[i][j],
            else:
                print " ",
        print
```

```
# Driver code
if _name_ == "_main_":
    a = [[1, 2, 3, 4], [5, 6, 7, 8],
         [1, 2, 3, 4], [5, 6, 7, 8]]
```

23. Print the given matrix in spiral order

```
a=[[2,5,3],
   [6,4,1],
   [9,7,8]]
l=[]
for i in range(len(a[0])):
    l.append(a[0][i])
for j in range(1,len(a)-1):
    l.append(a[j][-1])
for k in range(1,len(a[-1])+1):
    l.append(a[-1][-k])
for m in range(len(a[0])-1):
    l.append(a[1][m])
print(l)
```

24. Write a python program to find the sum of N numbers

Sample input: N=10

Sample output: Sum=55

```
N=int(input("Enter the limit:"))
count=0
for i in range(1,N+1):
    count+=i
print("Sum of N natural numbers",count)
```

25.Write a python program to find the sum of 1^2+2^2+.......N^2 numbers

Sample input: N=6

Sample output: Sum=91

```
N=int(input("Enter the limit:"))
count=0
for i in range(1,N+1):
    count+=i*i
print("Sum of square of N natural numbers",count)
```

26.Find the factorial of the number.

Sample input: N=5

Sample output: Sum=120

```
def fact(n):
   if n==0 or n==1:
      return 1
   if n>1:
      return n*fact(n-1)

# Main program
num=int(input("Enter the number: "))
print(fact(num))
```

27.Write a python program to find the sum of 1!+2!+.......N! numbers

Sample input: N=4

Sample output: Sum=33

```
def fact(n):
   if n==0 or n==1:
      return 1
   if n>1:
      return n*fact(n-1)

# Main program
num=int(input("Enter the number: "))
sum=0
for i in range(1,num+1):
   sum+=fact(i)
print(sum)
```

28.Write a python program to find the sum of 1!/1+2!/2+.......N!/N numbers

Sample input: N=5

Sample output: Sum=34

```
def fact(n):
   if n==0 or n==1:
      return 1
   if n>1:
      return n*fact(n-1)

# Main program
num=int(input("Enter the number: "))
sum=0
for i in range(1,num+1):
   sum+=fact(i)/i
print(sum)
```

29.Write a python program to find the difference between sum of square and square of sum N numbers

Sample input: N=5

Sample output: Diff=170

```
n=20
x=(n*(n+1)*(2*n+1))/6
y=((n*(n+1))/2)**2
print("Difference:",y-x)
```

30.Write a python program to find the sum of all digits in a triangle

```
def digits_sum():
        for i in reversed(range(len(triangle_num) - 1)):
                for j in range(len(triangle_num[i])):
                        triangle_num[i][j] += max(triangle_num[i + 1][j],
triangle_num[i + 1][j + 1])
        return str(triangle_num[0][0])

#Main Program
triangle_num =
        [[3],
      [4,6],
     [2,7,6],
   [8,5,9,3]]
print(digits_sum())
```

31. Fibonacci series

```
def Fibonacci(n):

  if n < 0:
    print("Incorrect input")

  elif n == 0:
    return 0

  elif n == 1 or n == 2:
    return 1

  else:
        return Fibonacci(n-1) + Fibonacci(n-2)

# Driver Program
num=int(input("Enter the number of terms="))
for i in range(num):
    print(Fibonacci(i))
```

32.You are climbing a staircase. It takes n steps to reach the top. Each time you can either climb 1 or 2 steps. In how many distinct ways can you climb to the top?

Sol:

```
def fib(n):
    if n <= 1:
        return n
    return fib(n-1) + fib(n-2)

# Driver program
s = int(input("Enter the value of n: "))
print ("Number of ways = ", end="")
print (fib(s+1))
```

Output:

```
Enter the value of n: 5
Number of ways = 8
```

33. Vehicles and children program

```
M=int(input("Enter the number of vehicles:"))
N=int(input("Enter number of children: "))
x=M%N
if x==0:
    print("You are so lucky")
elif x!=0 and x%2!=0:
    print("Mr.Peter gets", x, "Vehicles")
elif x!=0 and x%2==0:
    print("Mr.Peter gets", x, "Vehicles. He is lucky")
```

34. Find the difference between two dates.

```
from datetime import datetime
from dateutil import relativedelta

# get two dates
d1 = '17/7/1980'
d2 = '16/3/2007'
```

```
# convert string to date object
start_date = datetime.strptime(d1, "%d/%m/%Y")
end_date = datetime.strptime(d2, "%d/%m/%Y")

# Get the relativedelta between two dates
delta = relativedelta.relativedelta(end_date, start_date)
print('Years, Months, Days between two dates is')
print(delta.years, 'Years,', delta.months, 'months,', delta.days, 'days')

delta.years
d3=d1.split('/')
d4=d2.split('/')
BY=int(d3[2])
JY=int(d4[2])

if(delta.years>=19 and BY%4==0):
    print("I m a lucky adult")
elif delta.years<19:
    print("I m aspiring to become adult")

if BY%4==0:
    print("Birth year is leap Year")
else:
    print("Birth year is not a leap Year")

if JY%4==0:
    print("Joining year is leap Year")
else:
    print("Joining year is not a leap Year")
```

35. Calendar Programs

```
# Current time
from datetime import datetime
now=datetime.now()
print(now)

# Current date
from datetime import datetime
now=datetime.today()
print(now)

# Entire month in a year
import calendar
y = int(input("Enter the Year :"))
print(calendar.prcal(y))

# Particular month in a year
```

```
import calendar
y = int(input("Enter the Year :"))
m=int(input("Enter the month :"))
print(calendar.month(y,m))

#Program to find number of weekdays in(mm/yyyy)
import numpy as np
# Number of weekdays in March 2017
print("Number of weekdays in March 2017:",
     np.busday_count('2017-03', '2017-04'))


# Number of sundays in Nov 2020
print("Number of Sunday in november 2020:",
     np.busday_count('2020-11', '2020-12',weekmask='Sun'))
# input year and month
yearMonth = '2017-05'

# getting date of first monday
date = np.busday_offset(yearMonth, 0, roll='forward',weekmask='Mon')
# display date
print(date)
```

I. STRING OPERATIONS AND METHODS

1.Write a program to find the number of special characters in the given statement
Sample Input:
Given statement: Modi Birthday @ September 17, #&$% is the wishes code for him.
Sample Output:
Number of special Characters: 5

```
#Python code to Count Alphabets, Special character Numeric values and space
string=input("Please enter a string: ")#take input from the user
alphabets,num,special,space=0,0,0,0;#variable declaration and initilization
a=[]
d=[]
spl=[]
for i in range(len(string)):
  if(string[i].isalpha()): #check Alphabets letters
      #print(string[i],end="")
      alphabets+=1
      a.append(string[i])
  elif(string[i].isdigit()):#check numeric value
```

```
    #print(string[i],end="")
    num+=1
    d.append(string[i])
  elif(string[i].isspace()):#check space
    space+=1
  else:
    #print(string[i],end="")
    special+=1
    spl.append(string[i])
print("Alpabets letters: ",alphabets, a)
print("\nnumbers: ",num, d)
print("\nSpace: ",space)
print("\nSpecial characters: ",special, spl)
```

2. Write a program to print the number of vowels and number of consonants in the given statement and which is maximum?

Sample Input:
Saveetha School of Engineering Sample Output:
Number of vowels = 12 Number of Consonants = 15

```
str = input("Enter the string:")
vcount, ccount= 0,0
Vowels = "AaEeIiOoUu"
c=[]
v=[]
#Converting entire string to lower case to reduce the comparisons
#str = str.lower()
for i in range(0,len(str)):
    #Checks whether a character is a vowel
    if str[i] in (Vowels):
        vcount = vcount + 1
        v.append(str[i])
        #count = [each for each in str if each in Vowels]

    elif (str[i] !=" " and str[i] not in (Vowels)):
        ccount = ccount + 1
        c.append(str[i])
print("Total number of vowel and consonant are" );
print(vcount,v)
print(ccount,c)
```

3. Program to find whether two strings have same character in same index and returns the number of matches

Sample input:
S1=”what”

S2=”watch”

Sample output:

```
1
```

```
def match(s1,s2):
    count=0

    for i in range(min(len(s1),len(s2))):
        if s1[i].lower()==s2[i].lower():
            count=count+1
    return count

#Driver Program
S1="What"
S2="watch"
print("Total number of matches:")
print(match(S1,S2))
```

4. Program to print number of words in a line and number of lines in a para

Sample input:

'''This is the most straightforward way to count the number
of lines in a text file in Python. The readlines() method reads all
lines from a file and stores it in a list. Next, use the len() function
to find the length of the list which is nothing but total lines present in a file.'''

Sample output:

```
Number of lines:  3
Number of words in each line:
Line 1 18
Line 2 15
Line 3 22
```

```
#Program to print number of words in a line and number of lines in a para

string='''This is the most straightforward way to count the number
of lines in a text file in Python. The readlines() method reads all
lines from a file and stores it in a list. Next, use the len() function
to find the length of the list which is nothing but total lines present in a file.'''
str1=string.split(".")
str1.pop()
print("Number of lines: ",len(str1))
print("Number of words in each line:")
for i in range(len(str1)):
    words=str1[i].split()
    #print(words)
    print("Line",i+1,len(words))
```

5. Program to find number of sentences starts with "B"

Sample input:

```
'''The apple doesn't fall. ...
All that glitters are not gold. ...
A picture is worth a thousand words. ...
Beggers can't be choosers. ...
A bird in the hand. ...
Better safe than sorry. ...
An apple a day keeps doctor away. ...
Blood is thicker than water. ...'''
```

Sample output:

```
Total number of lines: 8
Number of Sentences that start with letter B : 3
```

```
# Program to find number of sentences starts with "B"

string=input(“Enter the Para: ”)
str1=string.split(" ...")
str1.pop()
print("Total number of lines:",len(str1))
count=0
for i in str1:
    str2=i.split()
    #print(str2)
    for j in str2:
        if j[0]=="B":
            count=count+1
print("Number of Sentences that start with letter B :",count)
```

6. Write a program that finds whether a given character is present in a string or not. In case it is present it prints the index at which it is present. Do not use built-in find functions to search the character.

Sample Input:

```
Enter the string: I am a programmer Enter the character to be searched: p
```

Sample Output:

```
P is found in string at index: 8
```

Note: Check for non-available Character in the given statement as Hidden Test case.

```
str = input("Enter the String:")

# Character to find
c = input("Enter the character to find:")
```

```
# Using Naive Method
res = None
j=0
while j<len(str):
   for i in range(0,len(str),1):
      if str[i] == c:
         res = True
         print(str[i], "Index:",i)
      j=j+1
if res==None:
   print("Character not found")
```

7. Write a program to arrange the letters of the word alphabetically in Normal order and reverse order

Sample Input:
Enter the word: MOSQUE Sample Output:
Alphabetical Order Normal: E M O Q S U Alphabetical Order Reverse: U S Q O M E

```
str=input("Enter the string:")
str=str.upper()
sort_str=sorted(str)
print(sort_str)
join_str="".join(sort_str)
rev_str=join_str[::-1]
print(join_str)
print(rev_str)
```

8. Write a program to find the number of letters repeatedly present in the given word and print the Repeated letters.

Sample Input:
Enter the word: TEMPLE Sample Output:
Number of repeated letters = 1 Repeated letter = E

```
string = input("Enter the string:")
string=string.lower()
repeat=[]
print("Duplicate characters in a given string: ");
#Counts each character present in the string
for i in range(0, len(string)):
   count = 1

   for j in range(i+1, len(string)):

      if(string[i] == string[j] and string[i] != ' '):
         count = count + 1;
```

```
            #Set string[j] to 0 to avoid printing visited character
            string = string[:j] + '0' + string[j+1:]

    #A character is considered as duplicate if count is greater than 1
    if(count > 1 and string[i] != '0'):
        repeat.append(string[i])
        print(string[i],count)

print("Number of repeated characters:", len(repeat),repeat)
```

9. Write functions to perform the following String operations and identify the vowels count in string S3.

Sample input: Index: 1
S1=’welcome’ S2=’Homely’

Sample output: wHeolmceolmye

```
s1 = "welcome"
s2 = "homely"
n = int(input("n="))
output = ""
i = 0
j = 0
while i < len(s1) and j < len(s2):
    output += s1[i:i+n] + s2[j:j+n]
    i += n
    j += n
output += s1[i:] + s2[j:]
print(output)
```

10. Write a program that accepts a string from user and re displays the same string after removing vowels from it.
Sample Input & Output:
Enter a string: we can play the game The string without vowels is: w cn ply th gm

Sol:

```
text = input("Enter the String: ")

vowels = ['a', 'e', 'i', 'o', 'u', 'A', 'E', 'I', 'O', 'U']
newtext = ""
for i in range(len(text)):
    if text[i] not in vowels:
        newtext = newtext + text[i]
```

```
print("\nString after removing Vowels: ")
text = newtext
print(text)
```

11. Given two strings “s” and “t”, determine if they are isomorphic.
Input: s = "egg", t = "add"
Output: true

```
def isisomorphic(str1, str2):
    if len(str1) != len(str2):
        return False
    else:
        map1, map2 = {}, {}
        for i in range(len(str1)):
            ch1, ch2 = str1[i], str2[i]
            if ch1 not in map1:
                map1[ch1] = ch2
            print(map1)
            if ch2 not in map2:
                map2[ch2] = ch1
            print(map2)
            if map1[ch1] != ch2 or map2[ch2] != ch1:
                return False
    return True

str1 = input("String 1=")
str2 = input("String 2=")
print(isisomorphic(str1, str2))
```

12. Given an integer n, return the number of strings of length n that consist only of vowels (a, e, i, o, u) and are lexicographically sorted.
Input: n = 2
Output: 15

```
def countstrings(n, start):
    if n == 0:
        return 1
    cnt = 0

    for i in range(start, 5):

        # decrease the length of string
        cnt += countstrings(n - 1, i)
    return cnt

def countVowelStrings(n):
    return countstrings(n, 0)

n = int(input("n="))
print(countVowelStrings(n))
```

13. Given a string S consisting of N lowercase alphabets, the task is to modify the string S by replacing each character with the alphabet whose circular distance from the character is equal to the frequency of the character in S.
Input: S="ghee"
Output: higg

```
def modify_string(S):
    frequency = {}

    # Count the frequency of each character
    for char in S:
        frequency[char] = frequency.get(char, 0) + 1

    result = ""

    # Replace characters with the corresponding circular distance
    for char in S:
        circular_distance = ord(char) + frequency[char]

        if circular_distance > 122:
            circular_distance -= 26

        result += chr(circular_distance)

    return result

# Example usage:
S = "ghee"
modified_string = modify_string(S)
print(modified_string)  # Output: higg
```

14. Given two strings S1 and S2, representing sentences, the task is to print both sentences after removing all words which are present in both sentences
Input: S1 = "sky is blue in color", S2 ="Raj likes sky blue color "
Output: is in
         Raj likes

```
def removeCommonWords(s1,s2):
  com=[]
  sent1=list(s1.split())
  sent2=list(s2.split())
  for i in sent1:
    if i in sent2:
      sent1.remove(i)
      sent2.remove(i)
      com.append(i)
      continue
  print(*sent1)
  print(*sent2)
  print("common words",*com)
sentence1 = input("Enter string1: ")
sentence2 = input("Enter string2: ")
removeCommonWords(sentence1,sentence2)
```

15. Given a string s consisting of words and spaces, return *the length of the **last** word in the string.* A **word** is a maximal substring consisting of non-space characters only.
Test Case:
Input: s = "Hello World"
Output: 5

```
s=input("Enter the string:")
s1=s.split()
n=len(s1)
print("Number of words: ",n)
print("Last word: ",s1[n-1], len(s1[n-1]))
```

16. Given a string s and an integer k, return the length of the longest substring of s such that the frequency of each character in this substring is greater than or equal to k.
s consists of only lowercase English letters.
Test cases:
1.Input: s = "aaabb", k = 3
Output: 3

```
def Substring(s):
    ans, temp = 1, 1
    for i in range(1, len(s)):
        if (s[i] == s[i - 1]):
            temp += 1
        else:
            ans = max(ans, temp)
            temp = 1

    ans = max(ans, temp)
    return ans

s = input("Enter the string: ")
print(Substring(s))
```

17. Reverse Words in a String
Given an input string s, reverse the order of the words.
Input: s = "the sky is blue"
Output: "blue is sky the"

```
str1=input("Enter the string: ")
str2=str1.split()[::-1]
print(*str2)
```

18. Raju, has again started troubling people in your city. The people have turned on to you for getting rid of Raju. Raju presents to you a number consisting of numbers from 0 to 9 characters. He wants you to reverse it from the final answer such that the number becomes Mirror number. A Mirror is a number which equals its reverse. The hope of people are on you so you have to solve the riddle. You have to tell if some number exists which you would reverse to convert the number into Mirror
Sample input:
Enter the number: 123456
Sample output:
Mirror image: 654321

```
num= int(input("Enter the integer: "))
num1=str(num)
num2=num1[::-1]
print(num2)
```

19. Given an array of strings strs, group **the anagrams** together. You can return the answer in **any order**.
Input: strs = ["eat","tea","tan","ate","nat","bat"]

Output: [["bat"],["nat","tan"],["ate","eat","tea"]]

```
def Anagrams(li ):
   dictionary = {}
   for word in li:
      sortedWord = ''.join(sorted(word))
      print(sortedWord)
      if sortedWord not in dictionary:
         dictionary[sortedWord] = [word]
      else:
         dictionary[sortedWord] += [word]
   return [dictionary[i] for i in dictionary]

li = ['pop','bat','tab','opp','cat']
print(Anagrams(li))
```

20. Program to print first letters of the word in a sentence separated by dot.

Sample input: "The cat on the wall"
Sample output: T.C.O.T.W.

```
string="The cat on the wall"
l1=list(string.split())
print(l1)

for i in range(len(l1)):
   print(l1[i][0].upper(),end=".")
   continue
```

21. Valid Palindrome
A phrase is a palindrome if, after converting all uppercase letters into lowercase letters and removing all non-alphanumeric characters, it reads the same forward and backward. Alphanumeric characters include letters and numbers.
Given a string s, return true if it is a palindrome, or false otherwise.
Test Cases:
1.Input: s = "A man, a plan, a canal: Panama"
Output: true

```
n="A man, a plan, a canal: Panam"
s=n.lower()
text=''
for i in s:
   if i.isalpha():
      text+=i
x=text[::-1]
if(x==text):
   print("Valid Palindrome")
else:
   print("Invalid Palindrome")
```

22. Write a function delchar(s,c) that takes as input strings s and c, where c has length 1 (i.e., a single character), and returns the string obtained by deleting all occurrences of c in s. If c has a length other than 1, the function should return s.
Sample Input:
Enter the string: Hello world
Enter a character to be deleted: l
Sample output:
String after the character is removed: Heo Word

```
#Display String after removing the given character

text = input("Enter the String: ")
char= input("Enter the char: ")
```

```
newtext = ""
for i in range(len(text)):
    if text[i]!=char:
        newtext = newtext + text[i]

print("\nString after removing the char: ")
text = newtext
print(text)
```

23. Given two strings haystack and needle, return the index of needle in haystack, if not return -1.
Sample input:
Haystack=’sadbutsad’
Needle=’sad’

Sample output:
[0,6]

```
def strStr(haystack,needle):
    l=[]
    if needle==" ":
        return 0
    else:
        for i in range(len(haystack)):
            if haystack[i]==needle[0]:
                if haystack[i:i+len(needle)]==needle:
                    l.append(i)
                    continue
                else:
                    return -1
        return l

# Driver Program

haystack="sadsad"
needle="sad"
print(strStr(haystack,needle))
```

24. Write a python program to evaluate math expression w/o eval().

```
def evaluate(string):
    string = string.replace(" ", "")

    def splitby(string, separators):
```

```
    lis = []
    current = ""
    for ch in string:
        if ch in separators:
            lis.append(current)
            lis.append(ch)
            current = ""
        else:
            current += ch
    lis.append(current)
    return lis

lis = splitby(string, "+-")

def evaluate_mul_div(string):
    lis = splitby(string, "x/")
    if len(lis) == 1:
        return lis[0]

    output = float(lis[0])
    lis = lis[1:]

    while len(lis) > 0:
        operator = lis[0]
        number = float(lis[1])
        lis = lis[2:]

        if operator == "x":
            output *= number

        elif operator == "/":
            output /= number

    return output


for i in range(len(lis)):
    lis[i] = evaluate_mul_div(lis[i])

output = float(lis[0])
lis = lis[1:]

while len(lis) > 0:
    operator = lis[0]
    number = float(lis[1])
    lis = lis[2:]
```

```
        if operator == "+":
            output += number
        elif operator == "-":
            output -= number

    return output

# Main Program
testcases = "1+2x3-4"
print(evaluate(testcases))
```

26. Largest 3 digit Palindrome

```
# Largest Palindrome
n = 0
for a in range(999, 100, -1):
    for b in range(a, 100, -1):
        x = a * b
        if x > n:
            s = str(a * b)
            if s == s[::-1]:
                n = a * b
print(n)
```

27. Given string num representing a non-negative integer num, and an integer k, return the smallest possible integer after removing k digits from num.

```
Input: num = "1432219", k = 3
Output: "1219"
```

```
def removeKdigits(num,k):
    stack = []
    for digit in num:
        while k > 0 and len(stack) > 0 and stack[-1] > digit:
            k -= 1
            stack.pop()
        stack.append(digit)
    if k > 0:
        stack = stack[:-k]
    return "".join(stack).lstrip("0") or "0"

num="143219"
k=2
print(removeKdigits(num,k))
```

28. Return the Unicode of Uppercase letters

```
import string
import re
alphabets = list(string.ascii_uppercase)
for i in alphabets:
    print(i,"=",ord(i))
print(chr(65))
```

29. Given two strings s1 and s2, write a function that will convert s1 to s2(if possible) by using min conversion.

```
def editDistance(str1, str2, m, n):

    if m == 0:
        return n

    if n == 0:
        return m

    if str1[m-1] == str2[n-1]:
        return editDistance(str1, str2, m-1, n-1)

    return 1 + min(editDistance(str1, str2, m, n-1),    # Insert
                   editDistance(str1, str2, m-1, n),    # Remove
                   editDistance(str1, str2, m-1, n-1)    # Replace
                   )

# Driver code
str1 = "sunday"
str2 = "saturday"
print (editDistance(str1, str2, len(str1), len(str2)))
```

30.

===
===

1. Sort a list

```
l=[1,2,74,34,23,78,66]
l.sort()
print(l)
```

===
===

2. Sum 1 to n using a While loop

```
n=10
i=1
```

```
s=0
while i<=n:
s=s+i
i+=1
print("sum ",s)
======================================================================
```

3. Concatenate 2 tuples. Find the index of a given element. Count occuurence of a given element.

```
a=(1,2,3)
b=(4,2,6)
c=a+b
print(c)
element=4
print("index of given element is ",c.index(element))
for i in c:
print(f"count of {i} is {c.count(i)} ")
======================================================================
```

4. Count special characters in a given string.

```
s="ajsd #$6 ^ dakf@#"
cc,dc,sp,spc=0,0,0,0
for i in range(len(s)):
if s[i].isalpha():
cc+=1
elif s[i].isdigit():dc+=1
elif s[i].isspace():
sp+=1
else:
spc+=1
print("char count ",cc)
print("digit count ",dc)
print("space count ",sp)
print("special char count ",spc)
======================================================================
```

5. Convert given string to uppercase. Also to lowercase.

```
s="Hello World"
print("upper case :",s.upper())
print("lower case :",s.lower())
======================================================================
```

6. Aged-based ticket price: 0-3: free; 4-12: Rs 10; >12: Rs 20

```
a=int(input("enter age "))
if a>0 and a<=3:
print("ticket is free of cost ")
elif a>3 and a<=10:
print("ticket cost is 10 rupees")
else:
print("ticket cost 20 rupees")
```

```
================================================================
===
```

7. Frequency of characters in a given string

```
s="helloworld"
d={}
for c in s:
if c not in d:
d[c]=1
else:
d[c]+=1
print(d)===========================================================
========
```

8. Mean, Median and Mode of a list of numbers

```
l=[1,2,3,6,45,1,5,9,21,67,10]
from statistics import mean,median,mode
print("mean is ",mean(l))
print("median is ",median(l))
print("mode is ",mode(l))
================================================================
===
```

9. Binary to Decimal and Octal

```
a="10010"
decimal=int(a,2)
octal=oct(int(a,2))
print("decimal num ",decimal)
print("octal num ",octal)
```

```
================================================================
===
```

1. Find Mth maximum number and Nth minimum in a list

```
m=3
n=2
l=[2,7,3,5,6,9,1]
l.sort()
print(l)
print("max",l[-m])
print("min",l[n-1])
================================================================
===
```

2. Sum of 1!+2!+.......N!

```
n=5
def fact(n):
f=1
for i in range(1,n+1):
f*=i
return f
s=0
for i in range(1,n+1):
s+=fact(i)
print("sum of factorials is ",s)
```

```
=====================================================================
===
3. Add two binary strings
a="1000"
b="0100"
a=int(a,2)
b=int(b,2)
print(a,b)
c=a+b
print("sum of binary nums is ",bin(c))
=====================================================================
===
4. Print a right triangle for a given n.E.g. n=3
1
21
321
n=5
for i in range(1,n+1):
for j in range(i,0,-1):
print(j,end=" ")
print("")
=====================================================================
===
5. LSD & MSD of a given number
n="12345"
print("LSD ",n[0])
print("MSD ",n[len(n)-1])
=====================================================================
===
6. Add two matrices
a=[[1,2],[3,4]]
b=[[4,3],[2,1]]
c=[[0,0],[0,0]]
r1=len(a)
c1=len(a[0])
r2=len(b)
c2=len(b[0])
if r1==r2 and c1==c2:
for i in range(r1):
for j in range(c1):
c[i][j]=a[i][j]+b[i][j]
else:
print("addition not possible ")
for r in c:
print(r)
=====================================================================
===7. Sum of digits of a number
n=12345
s=0
while n>0:
```

```
s+=n%10
n=n//10
print("sum of digits ",s)
==================================================================
===
```

8. Sum of squares of all odd numbers in a list

```
l=[1,2,3,4,5,6,7,8,9,10]
esq,osq=0,0
for i in l:
if i%2==0:
esq+=(i**2)
else:
osq+=(i**2)
print("odd square sum ",osq)
print("even square sun ",esq)
==================================================================
===
```

9. Arrange letters of a word alphabetically and then reverse it.

```
s="mosque"
sl=sorted(s)
print(sl)
res="".join(sl[::-1])
print(res)
==================================================================
===
```