**1.Find LSD and MSD of the given number**

**C0DE:**

def find\_lsd\_and\_msd(number):

number = abs(number)

lsd = number % 10

msd = int(str(number)[0])

return lsd, msd

user\_input = int(input("Enter an integer: "))

lsd, msd = find\_lsd\_and\_msd(user\_input)

print(f"The Least Significant Digit (LSD) is: {lsd}")

print(f"The Most Significant Digit (MSD) is: {msd}")

**OR**

number = input("Enter a number: ")

lsd = number[-1]

msd = number[0]

lsd = int(lsd)

msd = int(msd)

print(f"Least Significant Digit (LSD): {lsd}")

print(f"Most Significant Digit (MSD): {msd}")

**2.Square and Cube**

**CODE:**

n=int(input("Enter the value:"))

x=n\*\*2

y=n\*\*3

print(x)

print(y)

**3.Write the string in the upper case and count the number of spaces available in the string**

**CODE:**

user\_string = input("Enter a string: ")

uppercase\_string = user\_string.upper()

space\_count = user\_string.count(' ')

print(f"Uppercase String: {uppercase\_string}")

print(f"Number of Spaces: {space\_count}")

**4. Find out if the year of the given anniversary is a leap year or not. If leap year print the next anniversary, if not leap year then print the previous anniversary.**

**CODE:**

def is\_leap\_year(year):

if (year % 4 == 0 and year % 100 != 0) or (year % 400 == 0):

return True

else:

return False

year = int(input("Enter the year of the anniversary: "))

if is\_leap\_year(year):

print(f"{year} is a leap year.")

next\_anniversary = year + 1

print(f"The next anniversary year is {next\_anniversary}.")

else:

print(f"{year} is not a leap year.")

previous\_anniversary = year - 1

print(f"The previous anniversary year is {previous\_anniversary}.")

**5. Python program for operating on a tuple concentrating two tuples. find the index of a specified element in the tuple. counts the number of occurrences of a specified element in the tuple. display the modified tuple and the results of the index and count operations**

**CODE:**

tuple1 = tuple(input("Enter elements of the first tuple separated by space: ").split())

tuple2 = tuple(input("Enter elements of the second tuple separated by space: ").split())

concatenated\_tuple = tuple1 + tuple2

element = input("Enter the element to find the index and count: ")

try:

index\_of\_element = concatenated\_tuple.index(element)

except ValueError:

index\_of\_element = None

count\_of\_element = concatenated\_tuple.count(element)

print(f"Concatenated Tuple: {concatenated\_tuple}")

if index\_of\_element is not None:

print(f"The index of the element '{element}' is: {index\_of\_element}")

else:

print(f"The element '{element}' is not found in the tuple.")

print(f"The count of the element '{element}' is: {count\_of\_element}")

**6. Write a program to print all the non prime numbers between a and b.**

**CODE:**

a = int(input("Enter the starting number (a): "))

b = int(input("Enter the ending number (b): "))

def is\_prime(n):

if n <= 1:

return False

for i in range(2, int(n\*\*0.5) + 1):

if n % i == 0:

return False

return True

non\_prime\_numbers = []

for num in range(a, b + 1):

if not is\_prime(num):

non\_prime\_numbers.append(num)

print("Non-prime numbers between", a, "and", b, "are:", non\_prime\_numbers)

**INPUT:**2 to 9

**OUTPUT:** [4, 6, 8, 9]

**7. Write a program to sort a list according to length of the elements.**

**CODE:**

input\_list = input("Enter a list of words separated by spaces: ").split()

sorted\_list = sorted(input\_list, key=len)

print("Sorted list by length of elements:", sorted\_list)

**INPUT:** laptop phone tab computer

**OUTPUT:** ['tab', 'phone', 'laptop', 'computer']

**8. Write a program to sort words in alphabetical order.**

**CODE:**

words = input("Enter a list of words separated by spaces: ").split()

sorted\_words = sorted(words)

print("Sorted list of words in alphabetical order:", sorted\_words)

**INPUT:** ravi sujji akki

**OUTPUT:** ['akki', 'ravi', 'sujji']

**9. Write a program to read a character until a’\*’ is encountered. Also count the number of uppercase and lowercase numbers entered by the users.**

**CODE:**

def count\_characters():

uppercase\_count = 0

lowercase\_count = 0

number\_count = 0

print("Enter characters one by one. Enter '\*' to stop:")

while True:

ch = input()

if ch == '\*':

break

elif ch.isupper():

uppercase\_count += 1

elif ch.islower():

lowercase\_count += 1

elif ch.isdigit():

number\_count += 1

print("Number of uppercase letters:", uppercase\_count)

print("Number of lowercase letters:", lowercase\_count)

print("Number of digits:", number\_count)

count\_characters()

**INPUT:** A

b

D

4

7

x

\*

**OUTPUT:** Number of uppercase letters: 2

Number of lowercase letters: 2

Number of digits: 2

**10. Write a Python program to remove duplicates from a list**

**CODE:**

def remove\_duplicates(numbers):

return list(set(numbers))

def remove():

try:

user\_input = input("Enter a list of numbers separated by spaces: ")

numbers = list(map(int, user\_input.split()))

unique\_numbers = remove\_duplicates(numbers)

print("List after removing duplicates:", unique\_numbers)

except ValueError:

print("Please enter a valid list of numbers.")

remove()

**INPUT:** 2 2 3 4 5 2 3 4 5

**OUTPUT:** [2, 3, 4, 5]

**11. Write a python program to find the sum of N numbers entered by the user.**

**CODE:**

n=int(input("Enter the value:"))

sum=0

for i in range(1,n+1):

sum=sum+i

print(sum)

**INPUT:** 10

**OUTPUT:** 55

**12. Write a python program to print number of vowels and consonants in given string**

**CODE:**

def count\_vowels\_and\_consonants(input\_string):

vowels = "aeiouAEIOU"

vowel\_count = 0

consonant\_count = 0

for char in input\_string:

if char in vowels:

vowel\_count += 1

elif char.isalpha():

consonant\_count += 1

return vowel\_count, consonant\_count

input\_string=input('enter the string:')

vowels, consonants = count\_vowels\_and\_consonants(input\_string)

print(f"Number of vowels: {vowels}")

print(f"Number of consonants: {consonants}")

**INPUT:** Hello World

**OUTPUT:** Number of vowels: 3

Number of consonants: 7

**13. Program to find whether two strings have same index and returns the number of matches**

**CODE:**

string1 = input("Enter the first string: ")

string2 = input("Enter the second string: ")

if len(string1) != len(string2):

print("The strings have different lengths.")

else:

match\_count = 0

for i in range(len(string1)):

if string1[i] == string2[i]:

match\_count += 1

print(f"Number of matches at the same index: {match\_count}")

**INPUT:** Enter the first string: afsin

Enter the second string: afsin

**OUTPUT:** Number of matches at the same index: 5

**14. Write a program to count the words starting “T” In the string**

**CODE:**

def count\_words\_starting\_with\_t(input\_string):

words = input\_string.split()

count = 0

for word in words:

if word.lower().startswith('t'):

count += 1

return count

input\_string = input('enter the string:')

count = count\_words\_starting\_with\_t(input\_string)

print(f"Number of words starting with 'T': {count}")

**INPUT:** The monkeys drank the tea

**OUTPUT:** 3

**15. Write a program to convert to the case of the given string**

**CODE:**

def convert\_case(input\_string):

uppercase\_string = input\_string.upper()

lowercase\_string = input\_string.lower()

titlecase\_string = input\_string.title()

return uppercase\_string, lowercase\_string, titlecase\_string

input\_string = input('enter the string:')

uppercase, lowercase, titlecase = convert\_case(input\_string)

print(f"Uppercase: {uppercase}")

print(f"Lowercase: {lowercase}")

print(f"Titlecase: {titlecase}")1

**INPUT:** APPLE

**OUTPUT:** Uppercase: APPLE

Lowercase: apple

Title case: Apple

**16. Write a program to convert the Decimal to Binary**

**CODE:**

def decimal\_to\_binary(decimal\_number):

if decimal\_number == 0:

return "0"

binary\_number = ""

while decimal\_number > 0:

remainder = decimal\_number % 2

binary\_number = str(remainder) + binary\_number

decimal\_number = decimal\_number // 2

return binary\_number

decimal\_number = 57

binary\_representation = decimal\_to\_binary(decimal\_number)

print(f"Binary representation of {decimal\_number} is {binary\_representation}")

**INPUT:** 57

**OUTPUT:** Binary representation of 57 is 111001