Task 1:

def digits(n):

    count=0

    x=n

    while x:

        x=x//10

        count+=1

    return count

def findDisarium(n):

    count=digits(n)

    add=0

    x=n

    while x:

        r=x%10

        add=add+pow(r,count)

        count-=1

        x=x//10

    if add==n:

        print(True)

    else:

        print(False)

findDisarium(135)

**output:-**

True

Task 2:

def longest\_palindrome(s):

def is\_palindrome(text):

return text == text[::-1]

n = len(s)

if n == 0:

return "", 0

longest\_palindrome\_str = ""

max\_length = 0

count = 0

for i in range(n):

for j in range(i, n):

substring = s[i : j + 1]

if is\_palindrome(substring):

if len(substring) > max\_length:

max\_length = len(substring)

longest\_palindrome\_str = substring

count = 1

elif len(substring) == max\_length and substring == longest\_palindrome\_str:

count += 1

return longest\_palindrome\_str, count

print(longest\_palindrome("BBABCBCAB"))

**output:-**

('BAB', 1)

Task 3:

def compare\_versions(version1, version2):

v1 = list(map(int, version1.split('.')))

v2 = list(map(int, version2.split('.')))

max\_len = max(len(v1), len(v2))

v1.extend([0] \* (max\_len - len(v1)))

v2.extend([0] \* (max\_len - len(v2)))

for i in range(max\_len):

if v1[i] > v2[i]:

return 1

elif v1[i] < v2[i]:

return -1

return 0

print(compare\_versions("1.01", "1.001"))

**output:-**

0

Task 4:

def romanNum(num):

val = [

1000, 900, 500, 400,

100, 90, 50, 40,

10, 9, 5, 4, 1

]

syms = [

"M", "CM", "D", "CD",

"C", "XC", "L", "XL",

"X", "IX", "V", "IV", "I"

]

roman = ""

i = 0

while num > 0:

count = num // val[i]

roman += syms[i] \* count

num -= val[i] \* count

i += 1

return roman

print(romanNum(3))

**output:-**

III