1.

def isFlip(n):

c1 = 0

c0 = 0

print(n)

for i in n:

if i == "0":

c0 +=1

else:

c1 +=1

if c0==1 or c1==1 :

return "Yes"

return "No"

num = input("Enter the binary number:\n")

print(isFlip(num))

2.

def lowup(s):

d = {"No. of Upper Case Characters":0,"No. of Lower Case Characters":0}

for i in s:

if i.isupper():

d["No. of Upper Case Characters"] += 1

elif i.islower():

d["No. of Lower Case Characters"] += 1

for key in d:

print(key," : ",d[key])

lowup(input("Enter a String:\n"))

3.

def perfect(n):

sum\_divisor = 0

for i in range(1,n):

if n%i ==0:

sum\_divisor += i

if n == sum\_divisor:

print("Yes")

else:

print("No")

perfect(int(input("Enter a Number:\n")))

4.

def sort\_hyphen(s):

l = s.split("-")

l.sort()

print('-'.join(l))

sort\_hyphen(input("Enter a sequence seperated by '-'\n"))

5.

#This Program is not generic as I could not figure out

#how to take a nested list and work with it

def recListSum(arr):

total = 0

for i in arr:

if type(i)== type([]):

total += recListSum(i)

else:

total += int(i)

return total

l = [1,2,[3,4],[5,6]]

sum = recListSum(l)

print(sum)

6.

def digitSum(n):

if n<10:

return n

else:

return digitSum(n%10 + digitSum(n//10))

sum = digitSum(int(input("Enter a Number:\n")))

print(sum)

7.

def posSum(n):

if n<1:

return 0

else:

return n + posSum(n-2)

sum = posSum(int(input("Enter a Number:\n")))

print(sum)

8.

def geoSum(a,r,n):

if n==0:

return 0

else:

return a\*(pow(r,n-1)) + geoSum(a,r,n-1)

l = (input("Enter a, r and n seperated by spaces:\n")).split()

a = int(l[0])

r = int(l[1])

n = int(l[2])

sum = geoSum(a,r,n)

print(sum)

9.

def power(a,b):

if b==0:

return 1

else:

return a\*power(a,b-1)

l = (input("Enter the no. and its power seperated by a comma:\n")).split(",")

a = int(l[0])

b = int(l[1])

sum = power(a,b)

print(sum)

10.

def primeFactList(n,i=2):

if i<=n:

if(n%i==0):

print(i,end=" ")

primeFactList(n//i,i)

else:

primeFactList(n,i+1)

return ""

n = int(input("Enter a Number:\n"))

print(primeFactList(n))