#Napisz algorytm dodawania dwóch liczb binarnych o tej samej długości

def DodBin1Opcja(a,b):

w = ""

r = 0

n = len(a)

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

s = int(a[-i]) + int(b[-i]) + r

m = s % 2

r = s // 2

w = str(m) + w

if r>0:

w = str(r) + w

return w

# Dodawanie binarne gdy rozna dlugosc

def DodBin2Opcja(a,b):

if(len(a)<len(b)):

for i in range(len(a)-len(b)):

a = "0" + a

elif(len(b)<len(a)):

for i in range(len(b)-len(a)):

b = "0" + b

w = ""

r = 0

n = len(a)

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

s = int(a[-i]) + int(b[-i]) + r

m = s % 2

r = s // 2

w = str(m) + w

if r > 0:

w = str(r) + w

return w

# Wypisz wszystkie licz1by binarne 6 cyfrowe których liczba jedynek

# Jest 2 razy wieksza niz zer. (Cztery 1 i Dwa 0)

def ConvToBin(n):

wynik = ""

while n != 0:

if n % 2 == 0:

wynik = "0" + wynik

n = n // 2

if n % 2 == 1:

wynik = "1" + wynik

n = n // 2

return wynik

for i in range(1, 100):

binarna\_liczba = ConvToBin(i)

ilosc\_1 = binarna\_liczba.count("1")

ilosc\_0 = binarna\_liczba.count("0")

if len(binarna\_liczba) == 6 and ilosc\_1 == 4 and ilosc\_0 == 2:

print(binarna\_liczba)

#Szybkie Potęgowanie

def FastPow(a,n):

w = 1

while n>0:

if n % 2 == 1:

w = w \* a

a = a \* a

n = n // 2

return w

# Szybkie Potęgowanie rekurencyjnie

def FastPow2(a,n):

if n == 0:

return 1

if n % 2 == 1:

return FastPow2(a,n-1) \* a

else:

return FastPow2(a\*a, n // 2)

# Odejmowanie binarki

def odejmowanie\_binarnych(a,b):

if len(a) != len(b):

dlugosc\_roznica = abs(len(a)-len(b))

if(len(a) > len(b)):

for i in range(dlugosc\_roznica):

a = "0" + a

else:

for i in range(dlugosc\_roznica):

b = "0" + b

wynik = ""

pozyczka = 0

for i in range(len(a) - 1, -1, -1):

roznica = int(a[i]) - int(b[i]) – pozyczka

if roznica < 0:

roznica += 2

pozyczka = 1

else:

pozyczka = 0

wynik = str(roznica) + wynik

return wynik

def mnozenie\_binarnych(a, b):

len1 = len(a)

len2 = len(b)

wynik = [0] \* (len1 + len2)

for i in range(len1 - 1, -1, -1):

for j in range(len2 - 1, -1, -1):

temp = int(a[i]) \* int(b[j])

pozycja1 = i + j

pozycja2 = i + j + 1

temp += wynik[pozycja2]

wynik[pozycja1] += temp // 2

wynik[pozycja2] = temp % 2

return wynik

def dzielenie\_binarne(dzielna, dzielnik):

if int(dzielnik, 2) == 0:

print("Dzielnik równy zero!")

exit()

reszta = dzielna

iloraz = ""

while len(reszta) >= len(dzielnik):

temp = reszta[:len(dzielnik)]

if temp >= dzielnik:

iloraz += '1'

reszta = bin(int(temp, 2) - int(dzielnik, 2))[2:]

else:

iloraz += '0'

reszta = temp[1:]

if iloraz == "":

iloraz = "0"

return iloraz