\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1

#1

convert = {1:'One', 2:'Two', 3:'Three', 5:'Five', 7:'Seven', 9:'Nine', 11:'Eleven', 13:'Thirteen', 17:'Seventeen', 19:'Nineteen', 20:'Twenty', 30:'Thirty', 40:'Fourty', 50:'Fifty', 60:'Sixty', 70:'Seventy', 80:'Eighty', 90:'Ninety'}

def count\_prime\_letters(n):

primeNumbers = []

sum = 0

for i in range(2,n):

flag = 0

for j in range(2, int((i/2)+1)):

if i%j==0:

flag = flag + 1

if flag == 1:

break

if(flag == 0):

primeNumbers.append(i)

#print(primeNumbers)

for i in primeNumbers:

if i<=19:

temp = len(convert[i])

else:

j = i%10

k = i-j

temp = len(convert[j]+convert[k])

sum = sum + temp

return sum

n = int(input("N = "))

print(count\_prime\_letters(n))

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

2.

#2

def get\_square\_sum (M , N ) :

temp = N

while(M!=0):

sum = 0

i = 0

flag = 0

while(flag==0):

if(i\*i>=temp):

flag=1

else:

sum = sum+(i\*i)

i = i + 1

print("Sum of all perfect squares <=",temp," is: ",sum)

temp = sum

M = M-1

m = int(input("M = "))

n = int(input("N = "))

print(get\_square\_sum(m,n))

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

3.

#3

def check\_bit (X) :

count=1

j = 0

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

print(i)

j = j+1

if(X[i] == '1'):

if(count==2):

return j

else:

count = count+1

return -1

n = int(input("N : "))

x = "{0:b}".format(n)

print("Binary representation of ",n," is",x)

print("Output = ",check\_bit(x))

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

4.

def find\_multiple ( n ) :

return n+n

n = input("Enter 3 digit prime number : ")

print(find\_multiple ( n ))

#n = int(input("Enter 3 digit prime number : "))

#print("%d%d" % (n,n))

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

5.

def find\_the\_winner ( matches\_won , matches\_drawn , goals\_scored ) :

sum = []

max = 0

j=0

for i in range(0, n):

temp = matches\_won[i]+matches\_drawn[i]

if temp == max:

if goals\_scored[i]>goals\_scored[j]:

max = temp

j = i

if temp>max:

max = temp

j = i

sum.append(temp)

return j

n = int(input("Enter size : "))

matches\_won = []

matches\_drawn = []

goals\_scored = []

print("matches\_won : ")

for i in range(0, n):

matches\_won.append(int(input()))

print("matches\_drawn : ")

for i in range(0, n):

matches\_drawn.append(int(input()))

print("goals\_scored : ")

for i in range(0, n):

goals\_scored.append(int(input()))

print("Output : ", find\_the\_winner ( matches\_won , matches\_drawn , goals\_scored ))

6

def decoded\_String(S):

arr = {}

temp = 0

result = ""

for i in S:

if i in arr:

temp = arr.get(i)

temp = temp + 1;

if(temp<3):

result= result+i

arr[i]= temp

else:

arr[i]= 1

else:

arr[i]=1

result= result+i

return result

S = input("Enter string :")

print(decoded\_String(S))

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*