L,R =map(int,input().split())

def la\_so\_palindrome(n):

goc = n

dao\_nguoc = 0

while n > 0:

dao\_nguoc = dao\_nguoc \* 10 + n % 10

n //= 10

return goc == dao\_nguoc

def tong\_cac\_chu\_so(n):

tong = 0

while n > 0:

tong += n % 10

n //= 10

return tong

def la\_so\_nguyen\_to(n):

if n < 2:

return False

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

if n % i == 0:

return False

return True

def dem\_palindrome\_hop\_le(L, R):

dem = 0

for so in range(L, R + 1):

if la\_so\_palindrome(so) and la\_so\_nguyen\_to(tong\_cac\_chu\_so(so)):

dem += 1

return dem

print(dem\_palindrome\_hop\_le(L, R))