def is\_prime(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 twin\_primes(start, end):

Primes = [n for n in range(start, end + 1) if is\_prime(n)]

Twin\_primes\_list = [(primes[i], primes[I + 1]) for I in range(len(primes) – 1) if primes[I + 1] – primes[i] == 2]

Return twin\_primes\_list

Start = int(input(“Enter start: “))

End = int(input(“Enter end: “))

Print(“Twin Primes:”, twin\_primes(start, end))

2,

def is\_prime(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 is\_palindrome(n):

Return str(n) == str(n)[::-1]

def palindromic\_primes(start, end):

Return [n for n in range(start, end + 1) if is\_prime(n) and is\_palindrome(n)]

Start = int(input(“Enter start: “))

End = int(input(“Enter end: “))

Print(“Palindromic Primes:”, palindromic\_primes(start, end))