Define function isprime(n)

// The function returns true if n is prime

if n equals 2, return True

if n equals 3, return True

if n % 2 equals 0, return False

if n % 3 equals 0, return Falsse

i = 5, w = 2

while i \* i <= n:

if n % i equals 0, return False

i += w

w = 6 - w

return True

Define function find\_d(n)

//this function returns value of d and r

//n is odd so n-1 is even. we have to find d such that n-1=d.2^r

count=0, m=n-1

while (m%2 equals 0)

m=m//2

count=count+1

d=m, s=count

return d,s

//Code to tabulate first nos Fermat Pseudoprimes

num=3, a=2

t=time.time()

//It stores the present time in a variable t

//we have specified a as 2 because we are checking only for strong base 2 pseudoprime

While nos< 10

if(not isprime(num)) //if the number is not prime then only it tests for primality

d,s=find\_d(num) //The function finds the value of d and s such that n-1=d. (2^s)

for r in range(0,s)

p=d\*(2\*\*r)

if ((pow(a,d,num)==0)), print(num)

elif((pow(a,p,num)==num-1)), print(num)

nos=nos+1

num=num+2

b=time.time() //stores the current time in b

print (b-t)