**Aim: To implement diffie hellman algorithm.**

Program:

from random import choice

def get\_primes(max\_n):

a = [1] \* max\_n

a[0] = a[1] = 0

for (i, isprime) in enumerate(a):

if isprime:

yield i

for n in range(i\*i, max\_n, i):

a[n] = False

def algo(x, y, n, g):

a = (g\*\*x)%n

b = (g\*\*y)%n

print("A: {}\tB: {}".format(a, b))

k1 = (b\*\*x)%n

k2 = (a\*\*y)%n

try:

assert k1 == k2

print("Secret key value: {}".format(k1))

except:

print("Some implementation error...")

def main():

primes = list(get\_primes(10000))

n, g = choice(primes), choice(primes)

x, y = int(input("Enter value of x: ")), int(input("Enter value of y: "))

print("n: {}\tg: {}".format(n, g))

print("x: {}\ty: {}".format(x, y))

algo(x, y, n, g)

if \_\_name\_\_ == '\_\_main\_\_':

main()

Output:

Enter value of x: 5

Enter value of y: 13

n: 5879 g: 463

x: 5 y: 13

A: 1659 B: 4892

Secret key value: 5068