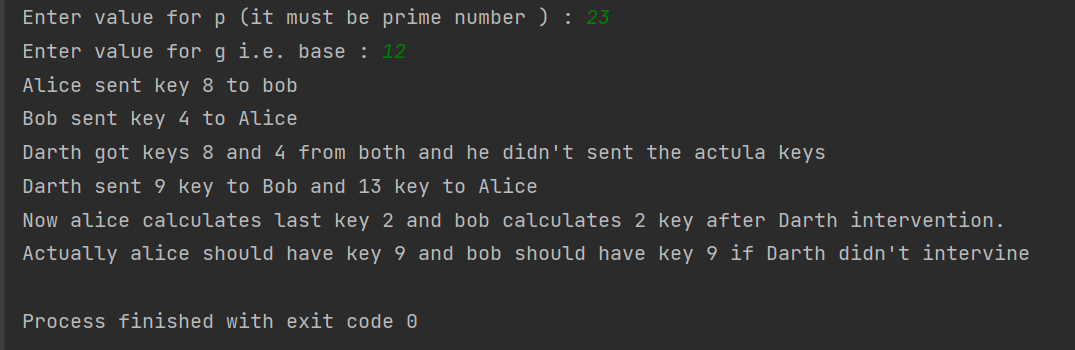
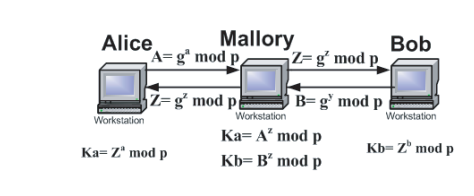
**Source Code :**

import random   
from time import sleep   
  
if \_\_name\_\_ == "\_\_main\_\_":  
 p = int(input("Enter value for p (it must be prime number ) : "))  
 g = int(input("Enter value for g i.e. base : "))  
  
 alice\_a = random.randint(1, 20)  
 alice\_A = (g \*\* alice\_a) % p   
 print("Alice sent key {} to bob".format(alice\_A))  
 sleep(3)  
  
 bob\_b = random.randint(1, 20)  
 bob\_B = (g \*\* bob\_b) % p   
 print("Bob sent key {} to Alice".format(bob\_B))  
 sleep(3)  
  
  
 print("Darth got keys {} and {} from both and he didn't sent the actula keys".format(alice\_A, bob\_B))  
 sleep(3)  
  
 eve\_c = random.randint(1, 20)  
 eve\_d = random.randint(1, 20)  
  
 eve\_C = (g \*\* eve\_c) % p   
 eve\_D = (g \*\* eve\_d) % p  
 print("Darth sent {} key to Bob and {} key to Alice".format(eve\_C, eve\_D))  
 sleep(3)  
  
 alice\_Calculates = (g \*\* (eve\_d \* alice\_a )) % p   
 bob\_calculates = (g \*\* ( bob\_b \* eve\_c )) % p  
  
 print("Now alice calculates last key {} and bob calculates {} key after Darth intervention.".format(alice\_Calculates, bob\_calculates))  
 sleep(3)  
   
 alice\_acutal\_key = (bob\_B \*\* alice\_a) % p  
 bob\_actual\_key = (alice\_A \*\* bob\_b) % p   
 print("Actually alice should have key {} and bob should have key {} if Darth didn't intervine".format(alice\_acutal\_key, bob\_actual\_key))

**Output :**





**Fig**. **Man in middle Attack against Diffe-Hellman algorithm**