*# # Function /method  
#a = 1+2  
#b = 3+4  
#c = 5+6  
#d = 7+8  
#e = 8+6  
#print(a)  
#  
# def ShowMyName():  
# a = 3+4  
# print(3+4)  
# print("Chandra")  
# print("Ravi")  
# print("Tej")  
# print("Narendra")  
# print("Prem")  
# print("haritha")  
#  
#print("Function completed")  
#  
#  
# # using the fucntion which defined above  
# ShowMyName()  
#  
# # param - name  
# # def - keyword  
# # ShowMyName - Function name  
# def ShowMyName(name):  
# # print("Contract Employee")  
# print("Contract Employee",name," - ICICI")  
#  
# # parametrised  
# ShowMyName("Chandra")  
# ShowMyName("Ram")  
# ShowMyName("Jaya")  
#  
# def multiplication(a):  
# print(a\*10)  
#  
# multiplication(10)  
# multiplication(12)  
#  
#***def** calculateintrest(amount,intrestrate):  
 *"""This function is used to  
 calculate interest of amount  
 based on rate"""* print(amount+(amount\*(intrestrate/100)) , **" "**)  
  
*#  
# # docstring  
# print(calculateintrest.\_\_doc\_\_)  
#  
# calculateintrest(5000,10)  
# # calculateintrest(5000,8)  
  
# def ShowMyName(name) :  
# print(name)  
#  
# myname = ShowMyName("Mohan")  
# print("Myname value" ,myname)  
#  
# # return statement , we will return value or name or details anything***def** TestReturn(name):  
 **return** name+**" Contract Employee"***#  
#  
# myempname = "Chandra"  
# num =20  
# print(myempname)  
# updatedname = TestReturn()  
# print(updatedname)  
  
# global,local*x = 10  
  
**def** globlExample():  
 *# Local means this x is accesible inside function only  
 # x=20* print(x \*4)  
  
*# globlExample()  
# you can't access local variable outside of the function  
# print(x)  
# if it is global it will access to every one  
  
# def CheckIsPrimeNumber(number):  
# if number >1 :  
# for i in range(2,number):  
# if(number%i == 0):  
# return "Number is not prime"  
# else : return "Number is prime"  
#  
# def FactorialOfNumber(num):  
# if num == 1:  
# return 1  
# else :  
# return num\* FactorialOfNumber(num-1)  
#  
# print(FactorialOfNumber(4))  
#  
# def FibonacciSeriesOfNumber(num):  
# x,y = 0,1  
# count = 0  
# while count < num :  
# print(x)  
# n = x+y  
# x = y  
# y = n  
# count += 1  
#  
# FibonacciSeriesOfNumber(5)  
# # [] ,()*bankcustomersdetails = {234355:{**"customerName"**: **"Nanda"**,**"AccountNumber"** : 124243455,**"Branch"**: **"Anantapur"**},  
 77777:{**"customerName"**: **"Jack"**,**"AccountNumber"** : 96863379,**"Branch"**: **"BLR"**}}  
  
  
  
**def** GetCustomerDetails(customerId):  
 checkcustomerexists = customerId **in** bankcustomersdetails  
 **if** checkcustomerexists == **True** :  
 customerdetails = bankcustomersdetails[customerId]  
 **return** customerdetails  
 **else** :  
 **return "This customer is not exists in our bank records"**requestedcustomer = GetCustomerDetails(234355)  
print(requestedcustomer)  
  
*# empdct = {"Jack":"AP","Hulk":"KA","Spiderman":"TN"}  
# emplst = ["Jack","Hulk","Spiderman"]  
#  
# isempexists = "Jack" in empdct  
#  
# print(isempexists)  
  
  
# in -keyword this  
# not in -- keyword  
# == operator - this is used to check whether two values are equal or not  
# =  
# !=  
# -  
# +  
# % 4%2 = 0  
# \*  
# / 4/2 = 2  
  
# x = 5  
# print(x)  
# # x = x+3  
# x += 3  
# print(x)  
# x -=2  
# print(x)  
#*z = 2 *# = is called assignment***if** z != 2 : *# comparison operator* **pass**z = z+3  
print(x)  
*#*x = 10  
**if** x>4 **and** x < 9: *# logical and operators both conditions should be true* print(**"x is between 6 and 9"**)  
  
**if** x<12 **or** x>11: *# logical or operator any of the condition should be true* print(**"Or condition is passed"**)  
  
**if not**(x< 1 **and** x >6) : *# False* print(**" Not condition check"**)  
*# not(True) ==> False  
# not(False) ==> True*x = 5  
y = 6  
**if** x **is not** y :  
 print(**"both x and y are same"**)  
*# Memebership operator  
# in # is exists inside the data*x = x\*2  
*# or , is , not***if** x > 5 **and** x <10 :  
 **pass***# 1,2,3,5,7,  
# 5 will start with 2 , till number -1 ,***def** CheckIsPrimerNumber(number):  
 **if** number ==1 **or** number == 2:  
 **return True  
 for** i **in** range(2,number) :  
 **if**(number % i == 0):  
 **return False  
 return True**prime = CheckIsPrimerNumber(2)  
print(prime)  
print(5%5)  
  
  
**def** CheckIsPrimerNumber():  
 lst = [1,2]  
 **for** n **in** range(3,101):  
 **for** i **in** range(2, n):  
 **if** (n % i != 0):  
 lst.append(n)  
  
 **return** lst  
  
print(CheckIsPrimerNumber())  
[1,2,3,5,7,9,11,13,17]