1. **Implement the Propositional basic logic gates along with implies and biconditional**

**Code to implement AND gate**

def AND (a, b):

if a == 1 and b == 1:

return True

else:

return False

# main function

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

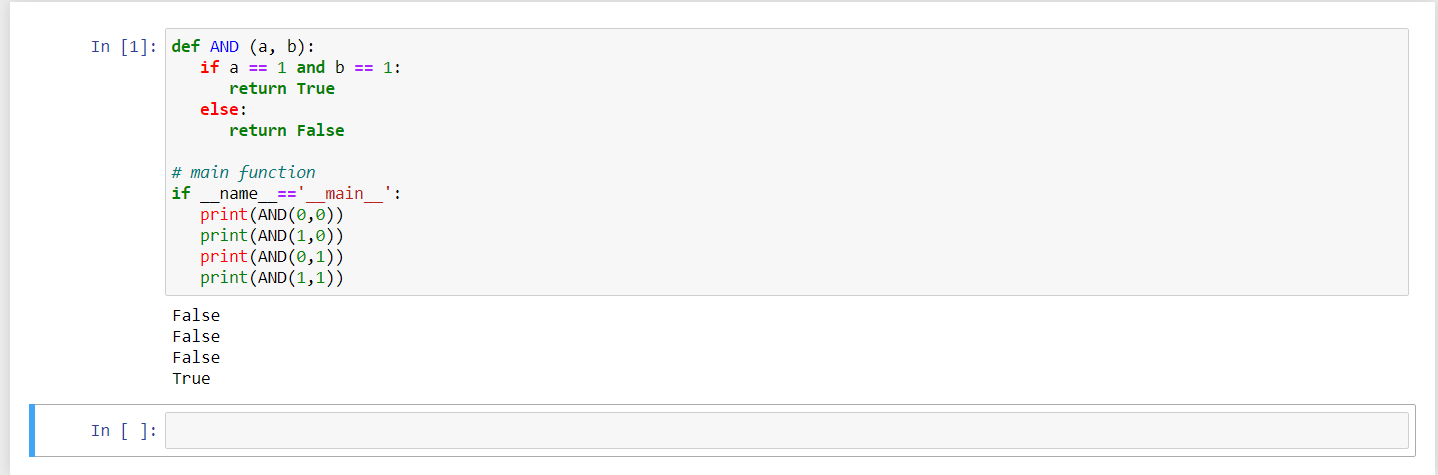
print(AND(0,0))

print(AND(1,0))

print(AND(0,1))

print(AND(1,1))

**OUTPUT:**



**Code to implement OR gate**

def OR(a, b):

if a == 1:

return True

elif b == 1:

return True

else:

return False

# main function

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

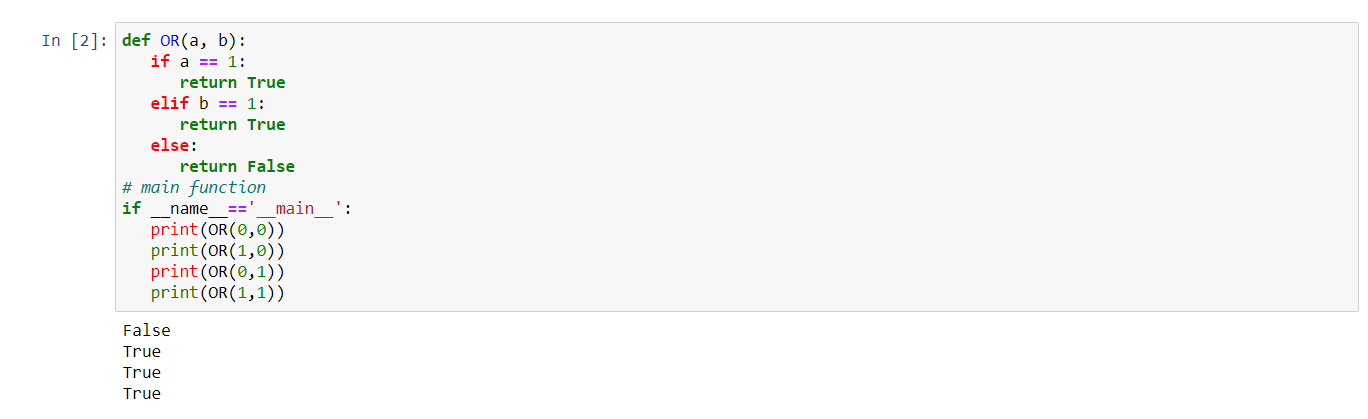
print(OR(0,0))

print(OR(1,0))

print(OR(0,1))

print(OR(1,1))

**OUTPUT:**



**Code to implement NOT gate**

def NOT(a):

if(a == 0):

return 1

elif(a == 1):

return 0

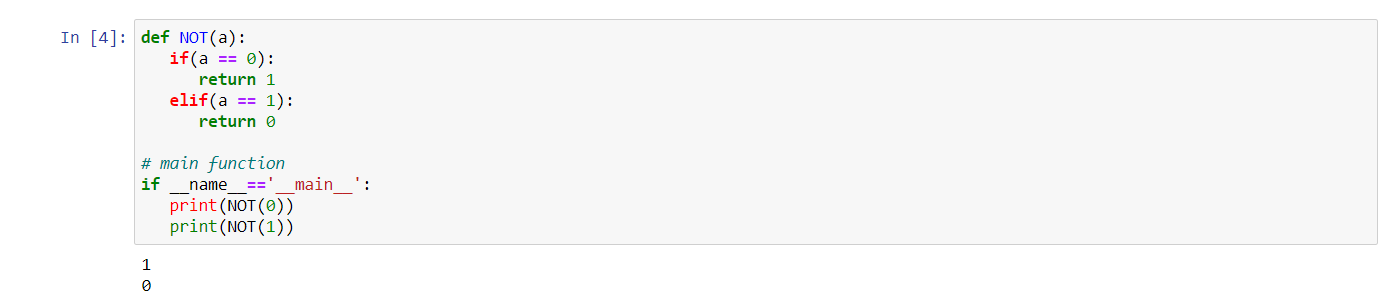
# main function

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

print(NOT(0))

print(NOT(1))

**OUTPUT:**



**2. Design the simplex reflex vacuum cleaner**

flag=True

count=1

while flag:

perc=input("enter the percept\n")

loc=input("enter the location\n")

if loc=="A":

if perc=="dirty":

print("action: suck...turn right")

else:

print("action: turn right")

else:

if perc=="dirty":

print("action: suck.....turn left")

else:

print("action: turn left")

print("Do you want to continue?")

Cont=input("Enter Y or N")

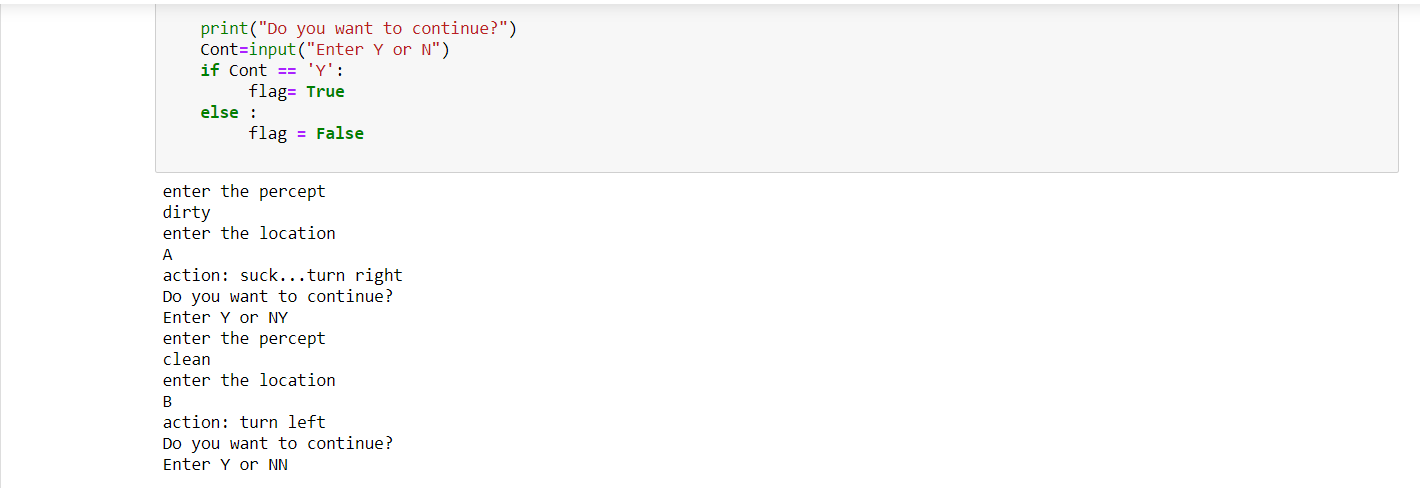
if Cont == 'Y':

flag= True

else :

flag = False

**OUTPUT:**



**Assume that there are 3 floors and 4 rooms in each floor. Design the vacuum cleaner to ensure the rooms are clean. You may make suitable assumption for initial state**.

# Given M x N grid(floor) create an agent that moves around the grid until the entire grid is clean

floor = [[1, 0, 0, 0], # '1' represents dirty and '0' represents clean

         [0, 1, 0, 1],

         [1, 0, 1, 1]]

def clean(floor):

    m = len(floor[0]) # no of cols

    n = len(floor)    # no of rows

    no\_of\_tiles = m \* n

    tiles\_checked = 0

    row = 0

    col = 0

    while tiles\_checked < no\_of\_tiles:

        # Current position

        print\_floor(floor, row, col)

        # Suck if dirty

        if floor[row][col] == 1:

            floor[row][col] = 0

            print('Sucked the dirt')

        else:

            print('Already Clean')

        # Next tile

        if row % 2 == 0:          # Even rows the bot moves right to the next tile

            if col < m-1:

                col += 1

            else:

                row += 1  # Move to next row if we reached the last col

        elif row % 2 == 1:        # Odd rows the bot moves left to the next tile

            if 0 < col:

                col -= 1

            else:

                row += 1  # Move to next row if we reached the last col

        tiles\_checked += 1

        print('---------------')

    print('Cleaned!!!')

def print\_floor(floor, row, col):

    temp = floor[row][col]

    floor[row][col] = 'VC'

    for x in floor:

        print(x)

    floor[row][col] = temp

# Call the function

clean(floor)

**OUTPUT:**



