2. Multiple Control Constructions

請以CCX量子邏輯閘搭配X量子邏輯閘完成以下程式碼，以下程式是一個實作CCCCNOT的程式碼，q[0]~q[3]是控制位元，q[4]是目標位元，qaux[0]~qaux[2]為輔助位元，請在**q[]**及**qaux[]**填入數字來完成此程式，當四個控制量子位元處於狀態 |1010⟩ 時，將 NOT 運算符應用於目標量子位元。以下程式碼已包含驗證程式，請直接將完成後的程式檔繳交至pc2(包含非紅色部分也要在程式中打出)，本題無範例輸出，直接繳交程式即可。

from qiskit import QuantumRegister, ClassicalRegister, QuantumCircuit, execute, Aer

all\_inputs=[]

for q4 in ['0','1']:

    for q3 in ['0','1']:

        for q2 in ['0','1']:

            for q1 in ['0','1']:

                for q0 in ['0','1']:

                    all\_inputs.append(q4+q3+q2+q1+q0)

#print(all\_inputs)

print("input --> output")

for the\_input in all\_inputs:

    # create the circuit

    qaux = QuantumRegister(2,"qaux")

    q = QuantumRegister(5,"q")

    c = ClassicalRegister(5,"c")

    qc = QuantumCircuit(q,qaux,c)

    # set the initial value of the circuit w.r.t. the input

    if the\_input[0] =='1': qc.x(q[4])

    if the\_input[1] =='1': qc.x(q[3])

    if the\_input[2] =='1': qc.x(q[2])

    if the\_input[3] =='1': qc.x(q[1])

    if the\_input[4] =='1': qc.x(q[0])

    #

**# your code is here**

**qc.x(q[])**

**qc.x(q[])**

**qc.ccx(q[],q[],qaux[])**

**qc.ccx(q[],q[],qaux[])**

**qc.ccx(qaux[],qaux[],q[])**

**qc.ccx(q[],q[],qaux[])**

**qc.ccx(q[],q[],qaux[])**

**qc.x(q[])**

**qc.x(q[])**

    #

    # measure the main quantum register

    qc.measure(q,c)

    # execute the circuit

    job = execute(qc,Aer.get\_backend('qasm\_simulator'),shots=1)

    counts = job.result().get\_counts(qc)

    for key in counts: the\_output = key

    printed\_str = the\_input[0:4]+" "+the\_input[4]+" --> "+the\_output[0:4]+" "+the\_output[4]

    if (the\_input!=the\_output): printed\_str = printed\_str + " the output is different than the input"

    print(printed\_str)