Part 1:

Part 1a

The truth table below describes the operation of a full adder.

We use the inputs A, B, CI (carry-in) and the outputs are S (sum) and CO (carry-out).

Derive the Boolean equations for both outputs. Apply logic minimization techniques to come up with a simplified full adder circuit.

Then, **complete the truth table** by filling in the values of CO and S.

CI	B	A	CO	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Part 1b

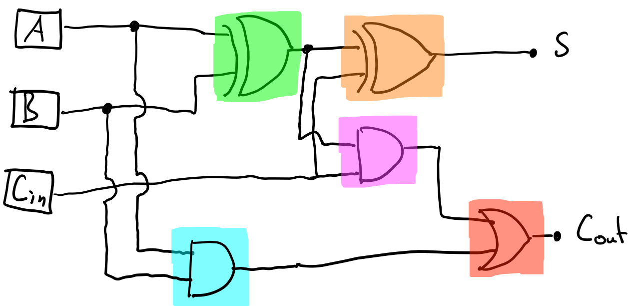
Derive Boolean equations for CO and S (using whatever method you like).

CO = (A AND B) OR (C_{in} AND (A XOR B))

S = (A XOR B) XOR C_{in}

Part 1c

Draw the schematic of the full adder circuit according to the equations you have derived.



Part 2-4:

```
module FullAdder( A,B,S);
input [3:0] A,B;
output [4:0] S;
wire n0,n1,n2;
carry carry0(A[0],B[0],0,S[0],n0);
carry carry1(A[1],B[1],n0,S[1],n1);
carry carry2(A[2],B[2],n1,S[2],n2);
carry carry3(A[3],B[3],n2,S[3],S[4]);
endmodule




module carry(A, B, Cin, Y, Cout);
// SUM = (A XOR B) XOR Cin = (A (+) B) (+) Cin
// CARRY-OUT = A AND B OR Cin(A XOR B) = A * B + Cin(A (+) B)
input A,B,Cin;
wire n0,n1,n2,n3;
output Y,Cout;

//SUM:
xor my_xor0(n0, A, B);
xor my_xor1(Y, n0, Cin);

//CARRY:
and my_and0(n3,A,B);
xor my_xor2(n1,A,B);
and my_and1(n2,Cin,n1);
or my_or(Cout,n3,n2);

endmodule
```