Justify why each line is correct. Each line should be either
- a law
- a definition, or
- follow from the previous line.

Say which!

Question 1

(a) Use Boolean algebra to show that $(X+Y).(X+Z) = X + Y.Z$

$$\begin{aligned}(X+Y).(X+Z) &= X.X + X.Y + X.Z + Y.Z \checkmark\\ \text{Justify} \quad &= X.(X+Y+Z) + Y.Z \checkmark\\ &= X + Y.Z \checkmark\end{aligned}$$

(b) Use Boolean algebra to show that $(X+Y).(\overline{X}+Z) = X.Z + \overline{X}.Y$

$$\begin{aligned}(X+Y).(\overline{X}+Z) &= X.\overline{X} + X.Z + \overline{X}.Y + Y.Z \quad \text{Justify}\\ &= 0 + X.Z + \overline{X}.Y + \underline{X.Y.Z + \overline{X}.Y.Z}\\ &= X.Z.(1+Y) + \overline{X}.Y.(1+Z)\\ &= X.Z.1 + \overline{X}.Y.1\\ &= X.Z + \overline{X}.Y \checkmark\end{aligned}$$

Good.

(c) Which common functional unit (i.e. standard digital circuit component) does the Boolean expression in part (b) describe?

A switch - where X controls whether the output from Y or the output from Z is transmitted. ✓

(2:1 mux) ~ Good. X is control or select input.

Question 2

The months of the year are coded in binary with January represented by $A_3A_2A_1A_0 =$ (0001) and December by (1100). Find a simplified sum-of-products expression in terms of A_3, A_2, A_1, A_0 for the months without an r in their name. Show that a simpler expression is obtained by changing the coding so January is represented by (0000) and December by (1011).

The months which don't have r in their name are: May, June, July, August – Which have binary representations of: (0101), (0110), (0111), (1000) respectively ✓
There are also several numbers which do not represent any months and so can be considered as "don't cares": (0000), (1101), (1110) and (1111). A Karnaugh Map of this gives:

		A_3A_2			
		00	01	11	10
A_1A_0	00	X	0	0	1
	01	0	1	X	0
	11	0	1	X	0
	10	0	1	X	0

✓ Good.

Using the Karnaugh Map we can see that the SOP form is $A_2A_1 + A_2A_0 + \overline{A_2A_1A_0}$

Careful! $\overline{A_2}.\overline{A_1}.\overline{A_0} \neq \overline{A_2.A_1.A_0}$ (✓)

If January is represented by (0000) and December by (1011) then the minterms are: (0100), (0101). (0110) and (0111). The "don't cares" will be (1100), (1101), (1110) and (1111).
On a Karnaugh Map this gives:

		A_3A_2			
		00	01	11	10
A_1A_0	00	0	1	X	0
	01	0	1	X	0
	11	0	1	X	0
	10	0	1	X	0

We can see that there is now only one prime implicant: A_2. ✓
So the simplified SOP form is A_2.

Good.

Question 3

(a) Explain the Quine-McCluskey method for a function of four variables, by analogy to the "equivalent" steps on a 4-variable Karnaugh Map. Show side-by-side the execution of the Quine-McCluskey method and what the equivalent steps on the Karnaugh map for the function Q(A, B, C, D) = $\sum$(0, 1, 3, 4, 7, 12, 13, 15).

The Quine-McCluskey ✓ ? method starts by writing down all the minterms ✓. This is the equivalent of marking out the minterms on a Karnaugh Map. ✓
It then checks all the minterms to see whether any differ by only one bit. If so it "ticks" those two terms and puts the combined version in the next column. If there are any at one stage which are not used then they are prime implicants and are starred. This is the equivalent of checking Karnaugh Maps to try and find groups which cannot be made any larger (the prime implicants). ✓
Finally, we would use a prime implicant chart to determine which prime implicants were in the covering set - and which were not. In the Karnaugh Map we can do this by inspection.

Explain why we do this, and why it's correct in each case.

Karnaugh Map Method: ~ how is this done?

		AB			
		00	01	11	10
CD	00	1	1	1	0
	01	1	0	1	0
	11	1	1	1	0
	10	0	0	0	0

✓ Show all PIs.

We would then look at the Karnaugh Map and see the largest groups we could make. In this case they would be: $\overline{ACD}$, $B\overline{CD}$, $\overline{ABC}$, $\overline{AB}D$, $AB\overline{C}$, ABD, $\overline{A}CD$, BCD
Visually we can look at this and see that there are lots of overlaps and so we only need half of them to make the covering set.
So a covering set (in this case there are two) is: $\overline{ABC}, \overline{A}CD, ABC, B\overline{CD}$.

Again, careful with ~ notation!

The Quine-McCluskey Method:

0000 ✓	000-
0001 ✓	0-00
0100 ✓	00-1
0011 ✓	-100
1100 ✓	0-11
0111 ✓	110-
1101 ✓	-111
1111 ✓	11-1

From this we can see all the prime implicants. Quine-McCluskey would then use a prime-implicant chart to decide which prime implicants are redundant. ~ Again, how? ✓

000-	X	X						
0-00	X			X				
00-1		X	X					
-100				X		X		
0-11			X		X			
110-						X	X	
-111					X			X
11-1							X	X
	0	1	3	4	7	12	13	15

From this Quine-McCluskey would see that

How do you know this is the minimum covering set?

there are no minterms which are represented by only one prime implicant. ✓ So Quine-McCluskey would select a prime implicant, say 000− and make the covering set from that. This would lead to the same covering set as the Karnaugh Map: $\overline{ABC}$, $\overline{A}CD$, ABC, $B\overline{CD}$ ✓

(b) (Optional.) Write a program in your favourite programming language that implements the Quine-McCluskey algorithm to simplify a combinatorial logic function. I'm not going to have time to look through your source code in detail, so show me some examples of input and output to prove it works!

Implemented in Python.

Question 4

(a) What is a static hazard in a combinatorial logic circuit?

A static hazard in combinatorical logic is where the delay in logic gates causes a single change in input which should not change the output to cause a short change in output. "glitch" ✓

(b) Consider the four-variable function Z(A, B, C, D) = ∑(1, 3, 5, 7, 8, 9, 12, 13). Identify potential static 0 or static 1 hazards when the function is implemented in

i. Sum-of-products form

Hmm... You should simplify this more first.

A SOP expression is $\overline{C}D + \overline{A}CD + A\overline{CD}$.
Summary: 1001 -> 1000 causes a static 0 hazard. ✗
If $A = 1$, $B = 0$, $C = 0$, $D = 1$ then the logic circuit is 1. If then D changes to 0, then the logic circuit is in a positive state (since $A\overline{CD}$ is 1). However, since $C\overline{D}$ takes one fewer logic gate to turn to 0 than $A\overline{CD}$ does to change to 1, the logic circuit will think that both are 0 for a short time and pulse 0. ✗

Good for indicating where the hazard arises

This can be prevented by using the SOP form: $\overline{A}D + B\overline{C}$ instead.

✗ But this no longer implements the same function.

ii. Product-of-sums form

A POS expression is $(A + D).(\overline{A} + \overline{C})$ ✓
Summary: 0010 -> 1010 causes a static 1 hazard. ✗
Consider if initially $A = 0$, $B = 0$ $C = 1$, $D = 0$ and then A changes to 1. A will change $(A+D)$ to 1 and then compare that to $(\overline{A}+\overline{C})$ before $(\overline{A}+\overline{C})$ has changed to 0. Meaning the circuit will pulse as 1 despite having transitioned from a negative state to another negative state. ✗

This can be prevented by adding another term - so that the expression is now $(\overline{A}+\overline{C}).(A+D).(C+D)$. ✗

We'll go through this in the supervision - not sure I understand how you're thinking.

Good for doing the extensions!

Question 5

How many Boolean functions of two arguments are there? Why?

For a Boolean function with two arguments, there are 4 possible inputs. For each input there are two possible outputs. So the number of possible boolean functions with two arguments is given by 2^4, which is 16.

So there are 16 possible Boolean functions with two arguments. ✓

Question 6

Using a four-variable Karnaugh map, fill it with 1s and 0s to find a function for which the minimised POS form is simpler than the minimised SOP form.

The Karnaugh Map below is simpler when solved using a POS form than a SOP form:

		AB			
		00	01	11	10
CD	00	0	0	0	0
	01	0	1	0	1
	11	0	1	0	1
	10	0	1	0	1

Let the expression be represented by f.
In SOP form:
$f = \overline{A}BC + \overline{A}BD + A\overline{B}C + A\overline{B}D$
Which has 12 literals and 15 logic gates.
In POS form:
$\overline{f} = \overline{A}\overline{B} + AB + \overline{C}\overline{D}$
$f = (A + B).(\overline{A} + \overline{B}).(C + D)$
Which has 6 literals and 7 logic gates.
So the POS form for this function is simpler than the SOP form. ✓

Good example.

Question 7

(a) Using the results in Q1 (a) and (b) simplify $P = (A + B + \overline{C}).(A + B + D).(A + B + \overline{E}).(A + \overline{D} + E).(\overline{A} + C)$

$$\begin{aligned}
P &= (A + B + \overline{C}).(A + B + D).(A + B + \overline{E}).(A + \overline{D} + E).(\overline{A} + C) \\
&= ((A + B) + \overline{C}).((A + B) + D)((A + B) + \overline{E}).(A + \overline{D} + E).(\overline{A} + C) \\
&= ((A + B) + \overline{C}D).((A + B) + \overline{E}).(A + \overline{D} + E).(\overline{A} + C) \\
&= ((A + B) + \overline{C}D\overline{E}).(A + \overline{D} + E).(\overline{A} + C) \\
&= (A + (B + \overline{C}D\overline{E})).(\overline{A} + C).(A + \overline{D} + E) \\
&= (AC + \overline{A}B + \overline{AC}D\overline{E}).(A + \overline{D} + E) \\
&= A.(AC + \overline{A}B + \overline{AC}D\overline{E}) + (\overline{D} + E).(AC + \overline{A}B + \overline{AC}D\overline{E}) \\
&= AAC + A\overline{A}B + A\overline{AC}D\overline{E} + AC\overline{D} + ACE + \overline{A}B\overline{D} + \overline{A}BE + \overline{AC}D\overline{D}\overline{E} + \overline{AC}D\overline{E}E \\
&= AC.(1 + \overline{D} + E) + 0B + 0\overline{C}D\overline{E} + \overline{A}B\overline{D} + \overline{A}BE + \overline{AC}0\overline{E} + \overline{AC}D0 \\
&= AC + \overline{A}B\overline{D} + \overline{A}BE \\
&= AC + \overline{A}B.(\overline{D} + E)
\end{aligned}$$

(b) Confirm your answer using a 5-variable Karnaugh map. (HINT: you may have to introduce a new kind of cell adjacency in order to be able to represent 5 variables on the map. Alternatively, you may find your answer from (a) useful in organising the Karnaugh map.)

In order to use a Karnaugh Map with 5 variables we need to have 2 separate 4 variable tables and create a relation between position i in table 0 and position i in table 1.

For the variables $ABCDE$ and the curcuit $P = (A+B+\overline{C}).(A+B+D).(A+B+\overline{E}).(A+\overline{D}+E).(\overline{A}+C)$ we can have the first table with $E = 0$ and the second table with $E = 1$.

$E = 0$

		AB			
		00	01	11	10
CD	00	0	1	0	0
	01	0	1	0	0
	11	0	0	1	1
	10	0	0	1	1

$E = 1$

		AB			
		00	01	11	10
CD	00	0	1	0	0
	01	0	1	0	0
	11	0	1	1	1
	10	0	1	1	1

In the first table (where $E = 0$), $P = 1$ where $AC = 1$ or $\overline{A}B\overline{D}$.

And in the second table (where $E = 1$), $P = 1$ where $AC = 1$ or $\overline{A}B\overline{D}$ or $\overline{A}B$.

So:

$$\begin{aligned}
P &= AC\overline{E} + ACE + \overline{A}B\overline{D}\overline{E} + \overline{A}B\overline{D}E + \overline{A}BE \\
&= AC(E + \overline{E}) + \overline{A}B\overline{D}(E + \overline{E}) + \overline{A}BE \\
&= AC1 + \overline{A}B\overline{D}1 + \overline{A}BE \\
&= AC + \overline{A}B(\overline{D} + E)
\end{aligned}$$

Which is the same as the algebraic manipulation approach.