

Logic Lab

Escape the lab by wiring logic gates: from a single NOT gate to truth tables, Boolean algebra and a binary adder.

UK Years 9–13

Ages 13–18

US grades 8–12

Computer Science · Electronics

Levels

LEVEL 9 · YEAR 13

The Calculator

★★★★

Brief

Hint

A1 A, 2s bit

1

A0 A, 1s bit

1

B1 B, 2s bit

1

B0 B, 1s bit

0

3 + 2 = 5

S2 4s bit

1

S1 2s bit

0

S0 1s bit

1

Truth table

Expression

16 of 16 rows right

IN				WANT			YOURS		
A1	A0	B1	B0	S2	S1	S0	S2	S1	S0
0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	1	0	0	1
0	0	1	0	0	1	0	0	1	0
0	0	1	1	0	1	1	0	1	1
0	1	0	0	0	0	1	0	0	1
0	1	0	1	0	1	0	0	1	0
0	1	1	0	0	1	1	0	1	1
0	1	1	1	1	0	0	1	0	0
1	0	0	0	0	1	0	0	1	0
1	0	0	1	0	1	1	0	1	1
1	0	1	0	1	0	0	1	0	0
1	0	1	1	1	0	1	1	0	1
1	1	0	0	0	1	1	0	1	1
1	1	0	1	1	0	0	1	0	0
1	1	1	0	1	0	1	1	0	1
1	1	1	1	1	1	0	1	1	0

Every row is right! Press "Test it" to see your stars.

AND

OR

NOT

XOR

Eraser

GATES

7

3 stars with 7 · 2 stars with 9

Test it

Inside this pack

1. Overview: learning outcomes, curriculum links, key words
2. How the game works (teacher background)
3. Level-by-level guide with 3-star solutions
4. Lesson 1: Logic gates and truth tables (Years 9–10)
5. Lesson 2: Boolean algebra and adders (Years 11–13)
6. Worksheet 1 and Worksheet 2 (photocopiable)
7. Answers and mark scheme

Play it free

dowplay.com/games/logic-lab

Runs in any browser on Chromebooks, iPads, laptops and phones. No accounts, no ads, nothing to install. Progress is saved only on the device being used.

A separate PDF has the worksheets on their own, ready to print for students.

1 · OVERVIEW

What students learn

Logic Lab is an escape room built from logic gates. Students are locked in the lab after dark, and every light, door and alarm needs wiring up. They drag gates onto a board and connect them with wires, from a night light (one NOT gate) to a 2-bit adder that unlocks the way out. A live truth table shows, row by row, what the circuit should do and what it actually does, and an Expression tab writes the circuit as Boolean algebra. Its nine levels run from Year 9 to Year 13.

By the end, students can

- recognise the symbols for **NOT**, **AND**, **OR**, **XOR** and **NAND** gates
- complete **truth tables** for single gates and for circuits with up to three inputs
- turn a written rule into a **logic circuit**, and a circuit into a **Boolean expression**
- simplify an expression, and use **De Morgan's laws** with NAND gates
- explain how **half adders** and **full adders** add binary numbers

You will need

- One device per pair (any browser)
- A projector for the starters
- Worksheet 1 or 2 for each student

Timing

Each lesson plan fills 60 minutes. The levels also work as a homework or a club activity: each takes 5–15 minutes.

Curriculum links

Course	Content	Levels
KS3 Computing (England)	Simple Boolean logic (AND, OR and NOT) and some of its uses in circuits	1–4
GCSE Computer Science	Logic gates (NOT, AND, OR, and XOR on some specifications); truth tables; logic circuits with up to three inputs; Boolean expressions	1–6
GCSE Electronics, D&T	Logic gates in systems and control: sensors, switches and outputs	1–6
A-level Computer Science	NAND, NOR and XOR gates; Boolean algebra and De Morgan's laws; half adders and full adders	5–9
US	AP Computer Science Principles: Boolean values and logic (AND, OR, NOT); CSTA standards on how computers represent and process data	All

Exam boards differ on which gates and which notation they use: check against your own specification. The game's Expression tab uses both words (A AND B) and symbols ($A \cdot B$, $A + B$, $A \oplus B$, a bar for NOT).

2 · TEACHER BACKGROUND

How the game works

The board is a real logic circuit:

1. Inputs (switches, sensors, keys, buttons and bits) are on the left; each is a 1 or a 0. Tap one to change it. Outputs (lamps, doors, a bell, a siren, lights) are on the right.
2. Students drag gates from the bar into the slots, and draw wires from an **output pin** (an input or a gate's output) to an **input pin** (a gate's input or an output). Each input pin takes one wire; an output pin can feed as many as needed.
3. The game works out every signal. Wires carrying 1 glow orange; 0 is grey. Anything not fully connected gives ?, and wires that would loop back into themselves aren't allowed.

The gates

Gate	Symbol	Output is 1 when...	Boolean expression
NOT		the input is 0 (it flips the input)	NOT A, or $\bar{A}$
AND		both inputs are 1	A AND B, or $A \cdot B$
OR		either input is 1, or both	A OR B, or $A + B$
XOR		the inputs are different	A XOR B, or $A \oplus B$
NAND		the inputs are not both 1 (NOT AND)	A NAND B, or $\overline{A \cdot B}$

Each level offers only some gates: level 1 just NOT; levels 2–6 NOT, AND and OR; level 7 only NAND; levels 8–9 add XOR.

2 · TEACHER BACKGROUND (CONTINUED)

Reading the screen

The **truth table** on the right lists every combination of inputs, counting up in binary. For each row it shows what the outputs **should** be ("Want") and what the student's circuit **gives** ("Yours"), with a tick or a cross. The row for the inputs as they're set now is outlined; tapping a row sets the inputs to it. Students can check their circuit row by row without leaving the board.

The **Expression** tab writes the circuit as Boolean algebra, in words and in symbols. **Test it** runs through every row so students can watch the circuit work, then gives the stars.

Stars

Stars are for using as few gates as possible: 1 star for a circuit that works, 2 for staying within a few gates of the best, 3 for using the fewest gates we know. For levels 4–8 the game's checker has tried every smaller circuit, so 3 stars really is the minimum.

Key words

Logic gate	A circuit that takes inputs of 1 or 0 and gives an output of 1 or 0 by a fixed rule.	Truth table	Every combination of inputs, with the output for each. n inputs give 2^n rows.
Boolean expression	A circuit written as algebra, such as $Q = A \text{ AND } (B \text{ OR } C)$.	Bit	A binary digit: 1 or 0.
Half adder	Adds two bits: $\text{sum} = A \text{ XOR } B$, $\text{carry} = A \text{ AND } B$.	Full adder	Adds two bits and a carry in, giving a sum and a carry out.
Universal gate	A gate (NAND or NOR) from which any other gate can be built.	De Morgan's laws	$\overline{A \cdot B} = \overline{A} + \overline{B}$ and $\overline{A + B} = \overline{A} \cdot \overline{B}$

Real chips

Every gate is made from a few transistors: a NAND gate needs just four in the CMOS chips used today, which is one reason designers like NAND. A modern processor has billions of transistors, and its adders work exactly like the one in The Calculator, one column at a time with a carry.

What the model leaves out

Real gates take a tiny time to switch, so signals ripple through an adder; in the game the answer appears at once. The game doesn't allow wires that loop back, so there's no memory: flip-flops and latches, which store bits by feeding outputs back to inputs, are the next step. Real inputs left unconnected float unpredictably; in the game they give ?. Good discussion points with older students.

3 · LEVEL GUIDE

Level by level

"3 stars" is the fewest gates we know. Keep it to yourself until students have had a go: the fun is in the search.

1 Lights Out

Year 9 · NOT

Watch for The sensor wired straight to the lamp: on in daylight, off in the dark.
3 stars, 1 gate A NOT gate between the light sensor and the lamp: $Q = \text{NOT } A$.
Ask "What would two NOT gates in a row do?" (Nothing: they cancel out.)

2 Two Keys

Year 9 · AND

Watch for An OR gate: the vault opens with only one key (2 of the 4 rows wrong).
3 stars, 1 gate An AND gate: $Q = A \text{ AND } B$.
Ask "Why would a bank want two people to turn keys?"

3 Ring the Bell

Year 9 · OR

Watch for An AND gate: the bell rings only when both buttons are pressed.
3 stars, 1 gate An OR gate: $Q = A \text{ OR } B$.
Ask "In which row is the bell silent?" (Only 0 0.)

4 Burglar Alarm

Year 10 · Combining gates

Watch for $(A \text{ AND } B) \text{ OR } C$: the open window sets off the siren even when the alarm isn't set (6 of 8 rows right).
3 stars, 2 gates $B \text{ OR } C$, fed into an AND gate with A: $Q = A \text{ AND } (B \text{ OR } C)$. $(A \text{ AND } B) \text{ OR } (A \text{ AND } C)$ also works, with 3 gates: 2 stars.
Ask "Why does a 3-input truth table have 8 rows?" ($2 \times 2 \times 2$.)

5 Landing Light

Years 10–11 · Exclusive OR

Watch for A plain OR gate: the lamp stays on when both switches are on (3 of 4 rows right).
3 stars, 4 gates $(A \text{ OR } B) \text{ AND NOT } (A \text{ AND } B)$. $(A \text{ AND NOT } B) \text{ OR } (\text{NOT } A \text{ AND } B)$ also works, with 5 gates: 2 stars.
Ask "Say the rule in one sentence." (The lamp is on when the switches are different: XOR.)

3 · LEVEL GUIDE (CONTINUED)

Level by level

6 Majority Vote

Year 11 · Simplifying

- Watch for** All three guards ANDed together: the door needs everyone (5 of 8 rows right).
- 3 stars, 4 gates** $(A \text{ AND } B) \text{ OR } (C \text{ AND } (A \text{ OR } B))$. The obvious $(A \text{ AND } B) \text{ OR } (B \text{ AND } C) \text{ OR } (A \text{ AND } C)$ uses 5 gates: 2 stars.
- Ask** "Why do the two circuits give the same truth table?"

7 Only NAND

Year 12 · Universal gates

- Watch for** A single NAND (2 of 4 rows right), or an AND made from two NANDs.
- 3 stars, 3 gates** A NAND with both inputs joined to A makes NOT A; the same for B; a third NAND combines them: $\text{NOT } A \text{ NAND NOT } B = A \text{ OR } B$.
- Ask** "How would you make an AND gate from NANDs?" (NAND, then a NAND used as a NOT: 2 gates.)

8 Half Adder

Year 12 · Adding bits

- Watch for** An OR gate for the sum: $1 + 1$ gives a sum of 1 (3 of 4 rows right).
- 3 stars, 2 gates** XOR for the sum, AND for the carry. Building the XOR from AND, OR and NOT (and sharing the carry's AND) takes 4 gates: 2 stars.
- Ask** "Why is $1 + 1$ written 10 in binary?"

9 The Calculator

Year 13 · A 2-bit adder

- Watch for** Forgetting the carry from the 1s column into the 2s column (12 of 16 rows right).
- 3 stars, 7 gates** A half adder for the 1s column (XOR, AND) and a full adder for the 2s column (2 XOR, 2 AND, 1 OR): $S1 = A1 \oplus B1 \oplus \text{carry}$, $S2 = A1 \cdot B1 + \text{carry} \cdot (A1 \oplus B1)$.
- Ask** "How many gates would a 4-bit adder need?" ($2 + 3 \times 5 = 17$.)

4 · LESSON PLAN · YEARS 9–10 · 60 MINUTES

Lesson 1: Logic gates and truth tables

Objectives

- Recognise NOT, AND, OR and XOR gates and their symbols.
- Complete truth tables for gates and simple circuits.
- Design a logic circuit from a written rule.

Resources

- Worksheet 1 (one each)
- Devices, one per pair
- Logic Lab on the projector

Time	Activity
Starter 10 min	Show Level 2, Two Keys . "The vault must open only when both keys are turned. How many different ways can the two keys be set?" Build the 4-row truth table together, then try an OR gate: the table shows two crosses. Swap it for AND: four ticks.
Main 1 15 min	Worksheet 1, Part A: name the gate symbols and complete their truth tables.
Main 2 25 min	Pairs play levels 1–5 and fill in the record table (Part B). Before they build, one partner fills in the "Want" column in their head and predicts which rows will be 1. Circulate and ask: "Which row is wrong? What should that gate do there?"
Plenary 10 min	Part C together: the burglar alarm's truth table and expression. Exit ticket: <i>what's the difference between OR and XOR?</i>

Support

Use the in-game hints. Have students say each rule aloud as a sentence ("the vault opens when key 1 AND key 2 are turned") before choosing a gate. Tapping the truth table's rows lets them test one row at a time.

Stretch

Get 3 stars on Landing Light (4 gates), then open the Expression tab and read the expression aloud. Try Level 6, Majority Vote.

Misconceptions to listen for

- "OR means one or the other, but not both." In logic, OR includes both; "one but not both" is XOR.
- "A NOT gate has two inputs." It has one: it just flips it.
- "I've written all the rows." Count up in binary (00, 01, 10, 11...) so none is missed: n inputs give 2^n rows.

Homework: Worksheet 1, Part D, and get 3 stars on Burglar Alarm.

5 · LESSON PLAN · YEARS 11–13 · 60 MINUTES

Lesson 2: Boolean algebra and adders

Objectives

- Write and simplify Boolean expressions.
- Use De Morgan's laws to build gates from NAND.
- Explain how half and full adders add binary numbers.

Resources

- Worksheet 2 (one each)
- Devices, one per pair
- Logic Lab on the projector

Time	Activity
Starter 10 min	"What's $1 + 1$?" Then: "In binary?" (10.) Write the four sums $0 + 0$ to $1 + 1$ as a truth table with two outputs, carry and sum. Ask which gates give each column, then build Level 8, Half Adder on the projector.
Main 1 15 min	Worksheet 2, questions 1–2: the majority vote and simplifying it, then NAND and De Morgan's laws.
Main 2 25 min	Pairs play levels 6–9 (Year 11 mainly levels 6–7; Years 12–13 levels 8–9). Use questions 3–5 to guide the adders.
Plenary 10 min	Plan question 6 together: how a computer adds with logic gates. Finish it as homework.

Support

Have students write each gate's output on the board's wires as they go. For The Calculator, build the 1s column first and check its rows in the truth table before starting the 2s column.

Stretch

Prove the majority-vote simplification with Boolean algebra. Design a 3-bit adder on paper, and count its gates.

Misconceptions to listen for

- "+ means add." In Boolean algebra, $A + B$ means $A \text{ OR } B$, so $1 + 1 = 1$. Binary addition is done by the adder circuits.
- " $\text{NOT}(A \text{ AND } B) = \text{NOT } A \text{ AND NOT } B$." De Morgan's law swaps the operator: it's $\text{NOT } A \text{ OR NOT } B$.
- "A full adder is two half adders." It's two half adders plus an OR gate to combine their carries.

Homework: Worksheet 2, question 6 (6 marks).

Logic gates and truth tables

Name: _____ Class: _____ Date: _____

In logic, **1** means on, true or yes, and **0** means off, false or no.

NOT flips its input · **AND** gives 1 if both inputs are 1 · **OR** gives 1 if either input is 1 (or both) · **XOR** gives 1 if the inputs are different

Part A · Gates

A1 Name each gate.

[4]



(a)



(b)



(c)



(d)

A2 Complete the truth tables.

[4]

AND

A	B	Q
0	0	
0	1	
1	0	
1	1	

OR

A	B	Q
0	0	
0	1	
1	0	
1	1	

XOR

A	B	Q
0	0	
0	1	
1	0	
1	1	

NOT

A	Q
0	
1	

A3 In **Ring the Bell**, the bell rings when the front button OR the back button is pressed. Explain why an AND gate would be the wrong choice.

[2]

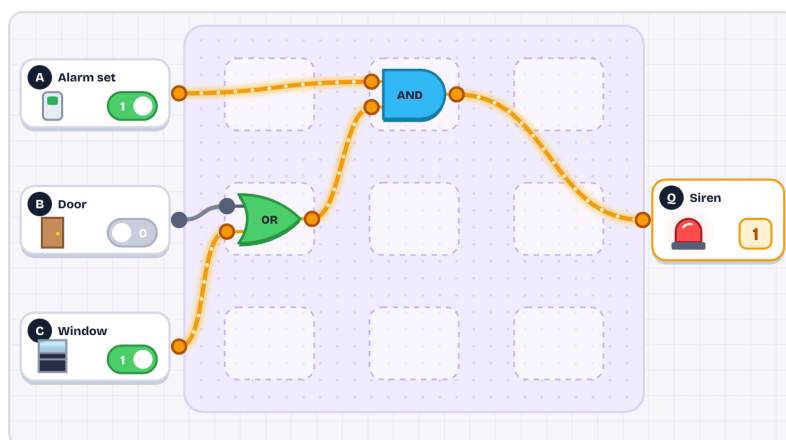
Part B · Play levels 1 to 5

Before you build, fill in the "Want" column in your head: which rows should give 1? Record your best circuit for each level.

Level	Gates I used	Stars	What went wrong, and how I fixed it
1			
2			
3			
4			
5			

Part C · The burglar alarm

The siren (Q) should sound when the alarm is set (A) AND the door (B) OR the window (C) is open. Here is the circuit, with A = 1, B = 0 and C = 1.



C1 Complete the truth table.

[3] **C2** Write the Boolean expression for Q.

[1]

A	B	C	B OR C	Q
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

C3 Why does the truth table have 8 rows?

[1]

C4 The owner adds a panic button, P. The siren should also sound whenever P is pressed, even if the alarm isn't set. Write the new expression.

[2]

Part D · Design a circuit

D1 A car's seatbelt warning light (Q) comes on when the engine is running (E = 1) AND the driver's seatbelt is NOT fastened (S = 0).

a) Complete the truth table.

[2] b) Draw the logic circuit, using two gates.

[2]

E	S	Q
0	0	
0	1	
1	0	
1	1	

c) Write the Boolean expression for Q.

[1]

Boolean algebra, NAND and adders

Name: _____ Class: _____ Date: _____

$A \cdot B$ means A AND B · $A + B$ means A OR B · $\overline{A}$ means NOT A · $A \oplus B$ means A XOR B
De Morgan's laws: $\overline{A \cdot B} = \overline{A} + \overline{B}$ and $\overline{A + B} = \overline{A} \cdot \overline{B}$ · Binary: $0 + 1 = 1$, $1 + 1 = 10$, $1 + 1 + 1 = 11$

1 Majority Vote. Three guards press buttons A, B and C. The lift door (Q) opens when at least two of them press.

a) Complete the Q column of the table. [2]

A	B	C	Q	$A \cdot B$	$A + B$	$C \cdot (A + B)$	$A \cdot B + C \cdot (A + B)$
0	0	0					
0	0	1					
0	1	0					
0	1	1					
1	0	0					
1	0	1					
1	1	0					
1	1	1					

b) Write Q as three AND terms joined by OR. [1]

c) Complete the other columns to show that $Q = A \cdot B + C \cdot (A + B)$ gives the same output. [2]

d) How many 2-input gates does each expression need? [1]

2 Only NAND.

a) Write down the NAND gate's output for the inputs 00, 01, 10 and 11. [1]

b) Show that a NAND gate with both of its inputs joined to A acts as a NOT gate. [1]

c) Use De Morgan's law to show that $\overline{A} \text{ NAND } \overline{B} = A + B$. [2]

d) Suggest why chip makers often build circuits entirely from NAND gates. [1]

3 Half adder. It adds two bits, A and B.

a) Complete the table.

A	B	Carry	Sum
0	0		
0	1		
1	0		
1	1		

b) Write Boolean expressions for the sum and the carry. [2]

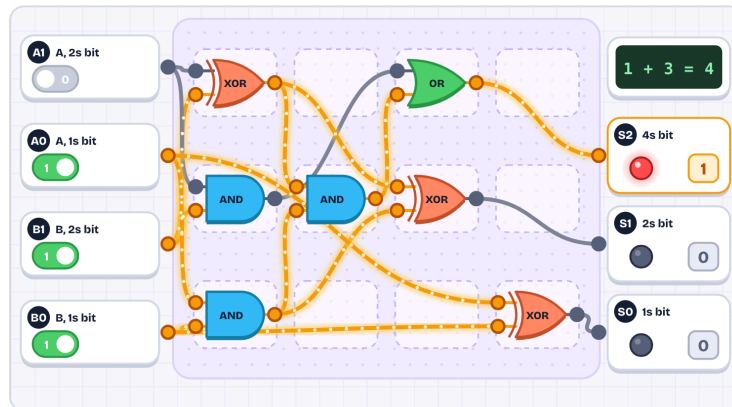
4 Full adder. It adds A, B and a carry in (C_{in}).

a) Complete the table. [2]

A	B	C_{in}	C_{out}	Sum
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

4b) A full adder uses $Sum = A \oplus B \oplus C_{in}$ and $C_{out} = A \cdot B + C_{in} \cdot (A \oplus B)$. Check both for $A = 1, B = 0, C_{in} = 1$. [2]

5 The Calculator adds two 2-bit numbers, A ($A_1 A_0$) and B ($B_1 B_0$). The 1s column is a half adder; the 2s column is a full adder. Here it is adding $1 + 3$.



a) Write 1, 3 and the answer in binary, and check the outputs S2 S1 S0 in the figure. [1]

b) In the 1s column, $A_0 = 1$ and $B_0 = 1$. What are S_0 and the carry? [1]

c) The 2s column adds $A_1 = 0, B_1 = 1$ and the carry. What are S_1 and S_2 ? [1]

d) A half adder takes 2 gates and a full adder 5. How many gates would a 4-bit adder need? An 8-bit adder? [2]

[6]

Plan your answer first.

Idea	What happens, with evidence from Logic Lab
Adding one column	
Carrying to the next column	
Adding bigger numbers	
Building from one gate type	

Now write your answer, finishing with a conclusion.

7 · ANSWERS

Worksheet 1 answers

Part A

A1 (a) **OR**, (b) **NOT**, (c) **AND**, (d) **XOR**.

A2 Q, for A B = 00, 01, 10, 11: AND **0 0 0 1**; OR **0 1 1 1**; XOR **0 1 1 0**. NOT, for A = 0, 1: **1 0**. (1 mark per table.)

A3 An AND gate gives 1 only when both inputs are 1 (1), so the bell would ring only if both buttons were pressed at once, not when a visitor presses just one (1).

Part C

C1 B OR C, rows 000 to 111: **0 1 1 1 0 1 1 1** (1). Q: **0 0 0 0 1 1 1 1** (2; 1 mark if one row is wrong).

C2 $Q = A \text{ AND } (B \text{ OR } C)$, or $Q = A \cdot (B + C)$.

C3 Each of the 3 inputs can be 0 or 1, so there are $2 \times 2 \times 2 = 2^3 = 8$ combinations.

C4 $Q = (A \text{ AND } (B \text{ OR } C)) \text{ OR } P$ (2), or $Q = A \cdot (B + C) + P$. (1 mark for adding P with OR.)

Part D

D1 a) Q, for E S = 00, 01, 10, 11: **0 0 1 0** (2; 1 mark if one row is wrong).

b) S goes through a NOT gate (1); its output and E go into an AND gate, which drives the light (1).

c) $Q = E \text{ AND NOT } S$, or $Q = E \cdot \bar{S}$.

Exit ticket

OR gives 1 when either input is 1 *or both*; XOR gives 1 only when the inputs are different, so it gives 0 when both are 1.

More from DowPlay

Free engineering games for ages 12–18: **Light It Up** (circuits), **Power Lines** (the National Grid), **Grid Manager** (renewable energy), **Gear Garage** (gears and machines) and **Bridge Builder** (structures), each with a free teacher pack. Ideas for new games? Get in touch at dowplay.com/about#contact.

Worksheet 2 answers

1 a) Q, rows 000 to 111: **0 0 0 1 0 1 1 1**.

b) $Q = A \cdot B + B \cdot C + A \cdot C$.

c) $A \cdot B$: 0 0 0 0 0 0 1 1; $A + B$: 0 0 1 1 1 1 1 1; $C \cdot (A + B)$: 0 0 0 1 0 1 0 1; total: **0 0 0 1 0 1 1 1**, the same as Q (2).

d) **5 gates** (3 AND, 2 OR) and **4 gates** (2 AND, 2 OR).

2 a) **1 1 1 0**.

b) With both inputs A: $A = 0$ gives $\text{NAND}(0, 0) = 1$; $A = 1$ gives $\text{NAND}(1, 1) = 0$. That's NOT A.

c) $\overline{A} \text{ NAND } \overline{B} = \text{NOT}(\overline{A} \cdot \overline{B})$ (1). By De Morgan's law this is $\text{NOT}(\overline{A} + \overline{B}) = A + B$, because $\text{NOT NOT } A = A$ (1).

d) Any one: a NAND gate is the simplest and cheapest to make (four transistors in CMOS); one type of gate makes chips simpler to design and manufacture; any circuit can be built from NANDs.

3 a) Carry, Sum for 00, 01, 10, 11: **0 0, 0 1, 0 1, 1 0**.

b) $\text{Sum} = A \oplus B$ (1); $\text{Carry} = A \cdot B$ (1).

4 a) C_{out} , Sum for rows 000 to 111: **00, 01, 01, 10, 01, 10, 10, 11** (2; 1 mark for one or two wrong rows).

b) $A \oplus B = 1$, so $\text{Sum} = 1 \oplus 1 = 0$ (1); $C_{\text{out}} = 1 \cdot 0 + 1 \cdot 1 = 1$ (1). $1 + 0 + 1 = 10$ in binary.

5 a) $01 + 11 = 100$: $S_2 = 1$, $S_1 = 0$, $S_0 = 0$, as in the figure.

b) $S_0 = 0$, carry = 1.

c) $0 + 1 + 1 = 10$, so $S_1 = 0$ and $S_2 = 1$.

d) $2 + 3 \times 5 = 17$ (1); $2 + 7 \times 5 = 37$ (1).

Question 6 mark scheme (6 marks)

Level	Description
3 (5–6 marks)	A clear explanation of how half and full adders add columns and pass carries along, linked to evidence from the game, with a balanced evaluation of building from one gate type and a justified conclusion.
2 (3–4 marks)	Explains adding with gates, with some detail on carries, and makes at least one evaluative point. The conclusion is partial.
1 (1–2 marks)	Simple statements about adders or gates, with little explanation.
0	Nothing worthy of credit.

Indicative content

Adding a column

- Sum = $A \oplus B$, carry = $A \cdot B$
(Half Adder: 2 gates)
- $1 + 1 = 10$: sum 0, carry 1

Carries and bigger numbers

- Each further column needs a full adder: A, B and the carry in
- Carries ripple from column to column (The Calculator: $1 + 3 = 100$)
- n bits: $2 + 5(n - 1)$ gates

One gate type

- NAND is universal (Only NAND: OR from 3 NANDs)
- Simple, cheap, identical parts
- But more gates are needed than with XOR, AND and OR