



F9.1 Logic gates and truth tables

Logic and computer systems · GCSE · OCR J277 2.4.1, AQA 8525 3.4.2,
Edexcel 1CP2 1.3.1 · about 15 min

What this lesson is about

AND, OR and NOT, their symbols and truth tables, and a robot decision as a circuit.

- AND, OR and NOT
- Truth tables
- A robot decision as a circuit
- Three inputs

Questions

6 marks in all

1. When does an AND gate output 1?

[1 mark]

- ☐ A Only when both inputs are 1
- ☐ B When at least one input is 1
- ☐ C When the inputs are different
- ☐ D When both inputs are 0

Answer: A. AND needs both inputs on.

2. When does an OR gate output 0?

[1 mark]

- ☐ A Only when both inputs are 0
- ☐ B When at least one input is 0
- ☐ C When the inputs are different
- ☐ D Never

Answer: A. OR is 1 if any input is 1, so it is 0 only when both are 0.

3. How many rows does a truth table for 3 inputs have?

[1 mark]

Answer: 8. $2 \times 2 \times 2 = 8$ combinations.

4. A is 1 and B is 1. What is A AND (NOT B)?

[1 mark]

Answer: 0. NOT B is 0, and 1 AND 0 is 0.

5. What does this program print?

[1 mark]

```
def AND(a, b): return 1 if a and b else 0
def NOT(a): return 1 - a
print(AND(1, NOT(0)), NOT(AND(1, 1)))
```

Answer:

1 0

NOT 0 is 1, so AND(1, 1) is 1; AND(1, 1) is 1, so NOT gives 0.

6. Which gate has only one input?

[1 mark]

- ☐ A NOT
- ☐ B AND
- ☐ C OR
- ☐ D XOR

Answer: A. NOT flips a single input.

The task: truth tables

Write `AND(a, b)`, `OR(a, b)` and `NOT(a)` as functions that take and return 0 or 1. Print the truth table for `A AND (NOT B)`, one row per line in the form `A=0 B=1 Q=0`, trying every combination with loops.

```
# the two lines every program starts with: the commands, then the robot
from bugbot import *
connect()

def AND(a, b):
    return 0
```

The hint students can ask for: Two nested loops give you every combination of the two inputs. Work Q out from your gate functions rather than writing the answers down.

A solution

```
from bugbot import *
connect()
def AND(a, b):
    return 1 if a == 1 and b == 1 else 0

def OR(a, b):
    return 1 if a == 1 or b == 1 else 0

def NOT(a):
    return 1 - a

for a in [0, 1]:
    for b in [0, 1]:
        print(f"A={a} B={b} Q={AND(a, NOT(b))}")
```

Any program that meets the task's checks is marked correct in the simulator; this is one way, not the only way.