



F8.4 Binary addition, overflow and shifts

Data representation · GCSE · OCR J277 1.2.4, AQA 8525 3.3.4, Edexcel
1CP2 2.1.4 · about 15 min

Name _____

Class _____

Date _____

What this lesson is about

Adding in binary with carries, overflow, and shifting to multiply and divide.

- Adding binary
- Overflow
- Shifts

Questions

5 marks in all

1. Add 01011010 and 00110111 in binary. [1 mark]

2. What is overflow? [1 mark]

- ☐ A A result too large to fit in the number of bits available
- ☐ B A number with too many decimal places
- ☐ C Running out of memory for files
- ☐ D A loop that never ends

3. In 8 bits, $200 + 100$ overflows. What value do the 8 bits kept hold? [1 mark]

4. What does this program print? [1 mark]

```
print(13 << 1, 13 << 2, 13 >> 1)
```

5. What does shifting a binary number two places left do? [1 mark]

- ☐ A Multiplies it by 4
- ☐ B Multiplies it by 2
- ☐ C Divides it by 4
- ☐ D Adds 2

The task: an 8-bit adder

Write `add_binary(a, b)` yourself (no `int(..., 2)` or `bin` inside it) that adds two 8-bit strings and returns the 8-bit answer and the carry out. Print `01011010 + 00110111 = 10010001` and, on the next line, `11001000 + 01100100 = 00101100 overflow`, working both answers out with your function.

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

def add_binary(a, b):
    return "00000000", 0
```

Plan your program here, then type it in and press Run.



Do it on the robot

www.bugbotlab.com/learn/f8-4-binary-addition-overflow-and-shifts/

The simulator checks it and tells you when it passes. Nothing to install, no account.

Challenges

1. Add three 8-bit numbers with your function, and check when overflow happens.
2. Write `shift_left(bits)` on strings: drop the first character and add a 0 at the end. Check it doubles the number.
3. Why does shifting right lose information? Find a number that shifting right then left does not bring back.