



A7.2 Binary arithmetic and signed integers

Data representation · A level · OCR H446 1.4.1, AQA 7517 4.5.2.1, Eduqas
A500QS 2.3 · about 30 min

Name _____

Class _____

Date _____

What this lesson is about

Unsigned addition and multiplication, sign and magnitude, two's complement, subtraction and overflow.

- Unsigned binary
- Sign and magnitude
- Two's complement
- Carry is not overflow
- Hexadecimal and signed values

Questions

6 marks in all

1. What is the denary value of the 8-bit two's complement number 11101100? [1 mark]

2. Write -37 as an 8-bit two's complement binary number. [1 mark]

3. What is the range of an 8-bit two's complement integer? [1 mark]

- ☐ A -128 to 127
- ☐ B -127 to 127
- ☐ C 0 to 255
- ☐ D -255 to 255

4. Write -5 in 8-bit sign and magnitude. [1 mark]

5. 01100100 + 01000110 is calculated in 8-bit two's complement. What happens? [1 mark]

- ☐ A Overflow: two positive numbers give a result with the sign bit set
- ☐ B A carry out of the top bit, which is an error
- ☐ C The correct answer, 170
- ☐ D Nothing unusual: the result is positive

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

```
a = 0b11111011
b = 0b00000101
total = a + b
print(format(total & 0xFF, "08b"), total >> 8)
```

The task: a signed adder

Write `add8(a, b)`. Each parameter is a string of exactly 8 characters, each `0` or `1`, holding an 8-bit two's complement number. It returns a tuple `(total, carry, overflow)` :- `total` : the 8-character bit string of the sum, with any carry out of the leftmost column dropped - `carry` : the carry out of the leftmost column, the integer 0 or 1 - `overflow` : `True` when both inputs have the same sign bit and `total` has the other one, otherwise `False`. Add column by column from the right; do not convert to denary with `int(... , 2)`, `bin` or `format`. The loop prints each line as `01110000 + 00110000 = 10100000 carry 0 overflow True`.

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

def add8(a, b):
    total = ""
    carry = 0
    # add the columns from right to left here
    return total, carry, False

tests = [("01110000", "00110000"), ("11111011", "00000101"),
         ("10000000", "11111111"), ("00101101", "11110110")]
for a, b in tests:
    total, carry, overflow = add8(a, b)
    print(a, "+", b, "=", total, "carry", carry, "overflow", overflow)
```

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



Do it on the robot

www.bugbotlab.com/learn/a7-2-binary-arithmetic-and-signed-integers/

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

Challenges

1. Write `sub8(a, b)` that uses `add8` and your own `negate` function, and test `50 - 25` and `-100 - 100`.
2. Write `mul8(a, b)` for unsigned bytes by shift and add. When should it report overflow?
3. Why can a positive number added to a negative number never overflow?