

Solutions

Each answer shows the steps, then the **result**.

2.1

- denary $\rightarrow$ 8-bit binary: place values 128, 64, 32, 16, 8, 4, 2, 1
- $45 = 32 + 8 + 4 + 1$
- **00101101**

2.2

- binary $\rightarrow$ hex: split into nibbles from the right
- $1100 = 12 = \text{C}$ and $1010 = 10 = \text{A}$
- **CA**

2.3

- denary $\rightarrow$ binary: divide by 2, record remainders, read **bottom to top**

```
200 ÷ 2 = 100  rem 0
100 ÷ 2 =  50  rem 0
 50 ÷ 2 =  25  rem 0
 25 ÷ 2 =  12  rem 1
 12 ÷ 2 =   6  rem 0
  6 ÷ 2 =   3  rem 0
  3 ÷ 2 =   1  rem 1
  1 ÷ 2 =   0  rem 1  ← left-most bit
```

- remainders bottom $\rightarrow$ top: **11001000**
- check: $128 + 64 + 8 = 200$

2.4

- hex $\rightarrow$ denary: 3F means $3 \times 16 + \text{F}$ and $\text{F} = 15$
- $48 + 15 = \mathbf{63}$

2.5

- any **two** reasons: shorter than binary / easier to read and copy / each hex digit is exactly four bits
- any **two** uses: MAC addresses / IPv6 addresses / HTML colour codes / memory dumps / error codes

3.1 B

- an 8-bit register stores $2^8 = 256$ different values, **0 to 255**
- A is 7-bit unsigned; C goes one past the last code; D misses 0

3.2

(a)

- add column by column from the right

$$\begin{array}{r}
 0110 \ 1100 \\
 + 0011 \ 1010 \\
 \hline
 1010 \ 0110
 \end{array}$$

- **10100110**

(b)

- 1-bits at $128 + 32 + 4 + 2 = \mathbf{166}$
- $166 \leq 255$ (no ninth bit), so **no overflow**

3.3

- 00110100 shifted left 2 places, zeros fill the right
- **11010000**; a left shift of 2 multiplies by $2^2 = 4$

3.4

(a)

- $172 = 128 + 32 + 8 + 4$
- **10101100**

(b)

- split 1010 1100
- $1010 = 10 = A$ and $1100 = 12 = C$, so **AC**

(c)

- add column by column from the right

$$\begin{array}{r}
 1011 \ 0101 \\
 + 0101 \ 1100 \\
 \hline
 1 \ 0001 \ 0001
 \end{array}$$

- stored 8-bit result **00010001**, with a carry out of bit 7

(d)

- **yes, overflow**
- the true sum needs 9 bits, but the 8-bit register cannot store the ninth, so the stored answer is wrong

(e)

- 00011010 shifted left 3 places $\rightarrow$ **11010000**
- denary was $16 + 8 + 2 = 26$ and becomes $128 + 64 + 16 = 208$
- a left shift of 3 multiplies by $2^3 = 8$, and $26 \times 8 = 208$

(f)

- two's complement: the MSB is worth -128
- $-128 + 64 + 32 + 8 + 4 = -\mathbf{20}$