

Solutions

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

2.1

- (a) **LENGTH**
- (b) **RIGHT**
- (c) **STR_TO_NUM**
- (d) **ASC**
- (e) **INT**

2.2

- (a) **8**
- (b) **"ter"**
- (c) **"pu"** – positions count from 1, so position 4 is the p
- (d) **9 – STR_TO_NUM** gives 9.7 and **INT** takes the whole-number part

2.3

- (a) the first three characters: **"Key"**
- (b) two characters from position 4: **"bo"**
- (c) the inner function first: **RIGHT(Word, 5)** is **"board"**, so the result is **"BOARD"**
- (d) **LENGTH(Word)** is 8, and **8 DIV 3** is **2**

2.4

- (a) **'C'** is two letters after **'A'**: $65 + 2 = 67$
- (b) $70 - 65 = 5$ letters after **'A'**: **'F'**
- (c) **&** joins the strings: **"AB12"**
- (d) the string becomes the number 12, and $12 + 3 = 15$

2.5

- **Dice** $\leftarrow$ **INT(RAND(6)) + 1**
- **RAND** returns a **real** from 0 up to but not including 6, so **INT** is needed to make it a whole number; the **+ 1** shifts the range from 0-5 to 1-6

2.6

- **+** cannot join a string to an integer: the two are different types
- **Message** $\leftarrow$ **"Total: " & NUM_TO_STR(Score) – &** concatenates, and the number must be converted to a string first

3.1

- (a) the inner call runs first: **RIGHT("Program", 2)** is **"am"**, so the answer is **"AM"**

- (b) **MID("Computer", 3, 4)** is **"mput"**, so **LENGTH** of it is **4**

- (c) **'Q'** is 81 and **'A'** is 65, so the answer is **16**

3.2

- the first three characters $\rightarrow$ **LEFT**
- a character to its code $\rightarrow$ **ASC**
- the string **"27"** must become the number 27 $\rightarrow$ **STR_TO_NUM**
- **"book"** starts at position 5 and has 4 characters $\rightarrow$ **5, 4**

3.3

- **INPUT**, then **TO_UPPER** and **LENGTH**

```
OUTPUT "Enter a word:"
INPUT Word
OUTPUT TO_UPPER(Word)
OUTPUT "Length: ", LENGTH(Word)
```

3.4

- **LEFT(Surname, 3)** for the first three letters
- **TO_UPPER(...)** around it: it accepts a whole string, whereas **UCASE** takes one **CHAR** and is not on the insert
- **LEFT(FirstName, 1)** for the initial
- **NUM_TO_STR(Year)**, and the three parts joined with **&**

```
Username  $\leftarrow$  TO_UPPER(LEFT(Surname, 3)) & LEFT(FirstName, 1) &
 $\rightarrow$  NUM_TO_STR(Year)
```

- for Surname **"Chen"**, FirstName **"Yuwei"** and Year 12 this gives **"CHEY12"**

3.5 any three:

- the routines have already been written, compiled and **tested**, so they are less likely to contain errors
- they save development time, because the code does not have to be written again
- they may do things the programmer could not write themselves, such as complex statistics or graphics; they are written by experts; and a routine with a fixed interface can be called from anywhere in the program, and reused across many programs

3.6

- (a) **LEFT(Code, 2)** $\rightarrow$ **"TV"**

- **MID(Code, 4, 4)** is **"2025"**, which becomes the number 2025, and $2025 + 1 = 2026$
- **LENGTH(Code)** is 12, and $12 - 8 = 4$

- (b) the last four characters hold the product number: **RIGHT(Code, 4)** (or **MID(Code, 9, 4)**)

- they are converted with **STR_TO_NUM**

- the result is assigned to `ProductNumber`: `ProductNumber ← STR_TO_NUM(RIGHT(Code, 4))`

(c) the two letters: `LEFT(Code, 2)`

- made small with `TO_LOWER`
- the last two digits of the year: `MID(Code, 6, 2)`
- the two parts joined with `&` and assigned: `Label ← TO_LOWER(LEFT(Code, 2)) & MID(Code, 6, 2)`