1.

(a) (i)

Identifier	Data type	Description
Ounces	**INTEGER**	Variable used as control variable in FOR loop
Grams	**REAL/Float/single/decimal/double**	Variable used for storing result of conversion calculation

        1 mark for each data type   [2]

(ii)
```
OUTPUT "   Conversion Table"
OUTPUT "Ounces            Grams"
FOR Ounces ← 1 TO 16
   Grams ← Ounces * 28.35
   Grams ← ROUND(Grams)
   OUTPUT Ounces, "              ", Grams
ENDFOR  // NEXT Ounces
```

        1 mark for each correct line   [4]

(b) (i)   20 DIV 6 = **3**
              20 MOD 6 = **2**   [2]

(ii)

```
FUNCTION CalculateNumberOfBoxes(NumberOfEggs : INTEGER)    ⎫
                                    RETURNS INTEGER        ⎬  1 mark
                                                           ⎭
   DECLARE NumberOfBoxes : INTEGER                            1 mark
   NumberOfBoxes ← NumberOfEggs DIV 6                         1 mark
   IF NumberOfEggs MOD 6 > 0              // 6>= 1            1 mark
      THEN
         NumberOfBoxes ← NumberOfBoxes + 1                    1 mark
   ENDIF
   RETURN NumberOfBoxes
ENDFUNCTION
```

                                                                                                [5]

(c) A function always returns a value. A procedure may or may not return one or more values [1]

2.

(i)

x	ThisValue	y	List[y]	(List[y] > ThisValue) AND (y > 0)	List [1]	[2]	[3]	[4]
–	–	–	–	–	56	23	67	12
2	23	1	56	TRUE		56		
		0		FALSE	23			
3	67	2	56	FALSE			(67)	
4	12	3	67	TRUE				67
		2	56	TRUE			56	
		1	23	TRUE		23		
		0		FALSE	12			

1 mark for each column correct [9]

(ii) (insertion) sort // ascending order [1]

3.

(b)

```
Index ← 1
```

```
Word1[Index] < Word2[Index]
```

```
Index ← Index + 1
```

```
Length(Word1) < Index
```

```
OUTPUT
Word1,Word2
WWord2Word2
Word2
```

```
OUTPUT
Word2, Word1
```

[6]

4.

(a) – within the function [1]

(b) – line 5 [1]

(c) – Calc(3)
– Calc(1)
– Calc(-1) [3]

(d) – 12 [1]

5.

(a) `BOOLEAN` [1]

Flags when the book title is found [1]

`STRING (for SearchBook)` [1]

```
OPENFILE Book.txt for Output
INPUT SearchBook                                                [1]
IsFound ← FALSE

REPEAT
    READ next book data value and assign to NextBook
    IF  NextBook = SearchBook                                   [1]
        THEN
            IsFound ← TRUE
            OUTPUT "FOUND"
    ENDIF
UNTIL (IsFound = TRUE) OR EOF                                   [1]

IF IsFound = FALSE // NOT IsFound                               [1]
    THEN
        OUTPUT "Book title was NOT FOUND"
ENDIF
CLOSEFILE                                                       [1]
```

(b) The search will read on average 125 records [1]

(c) (i) The data items must be in order [1]

(ii) The function makes a call to itself (in two places) [1]

(iii)
```
BinarySearch(BookTitle, "Tortoise Care", 1, 11)
    High < Low is FALSE
    Middle = 6
    BookTitle[6] > "Tortoise Care" is FALSE
    BookTitle[6] < "Tortoise Care" is TRUE
    BinarySearch(BookTitle, "Tortoise Care"  7, 11)           [1]
```

```
    High < Low is FALSE
    Middle = 9                                                 [1]
    Booktitle[9] > "Tortoise Care" is FALSE
    Booktitle[9] < "Tortoise Care" is TRUE
    BinarySearch(BookTitle, "Tortoise Care" 10, 11)            [1]

        High < Low is FALSE
        Middle = 10
        BookTitle[10] > "Tortoise Care" is FALSE               [1]
        Booktitle[10] < "Tortoise Care" is FALSE
        RETURN 10
    ENDFUNCTION

    ENDFUNCTION                                                [1]
```

```
ENDFUNCTION
```

[Total: 16]

6.

Award [4 marks] as follows.
Award [1 mark] for going 3 times through the loop (with COUNT from 1 to 3).
Award [1 mark] for incrementing correctly SUM (when N mod COUNT = 0).
Award [1 mark] for the correct output ("perfect").
Award [1 mark] for showing all working in a trace table with at least three columns (eg COUNT, SUM, OUTPUT).
Award the first 3 marks for an evident trace but working not shown in a trace table.

Example answer 1:

COUNT	N mod COUNT=0	SUM	SUM=N	output
1	TRUE	1		
2	TRUE	3		
3	TRUE	6		
			TRUE	perfect

Example answer 2:

COUNT	N mod COUNT	SUM	output
		0	
1	0	1	
2	0	3	
3	0	6	perfect

[4]