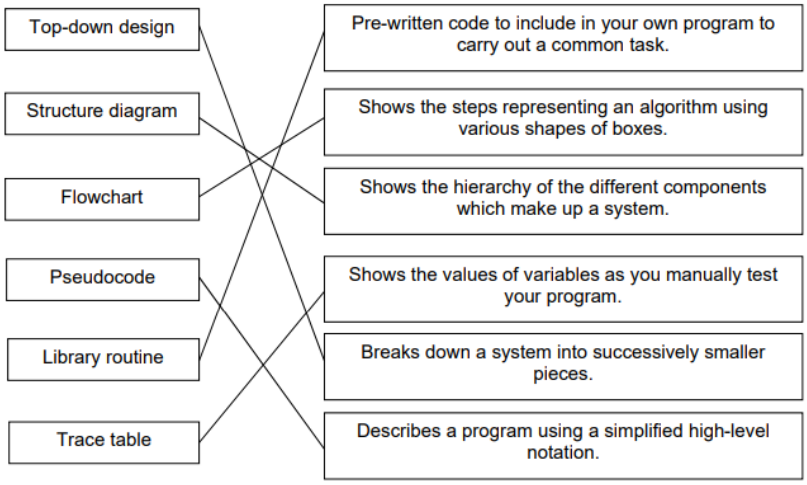
8. Programming

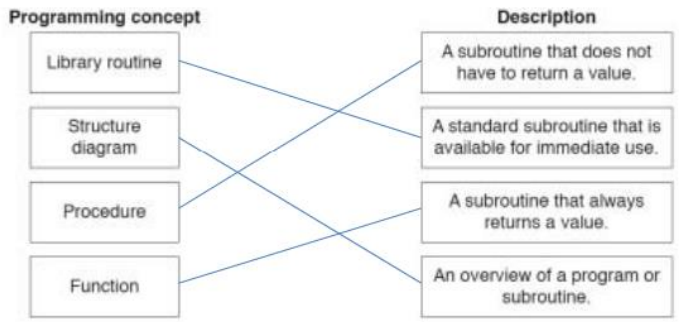
8.1 Programming concepts

Marking Scheme

Q1)

	1 mark for each correct line, maximum 5 marks **Term** **Description**  	**5**

Q2)

Question	Answer	Marks
	1 mark for each correct line (max 3) Each box must have only one connection.  	**3**

Q3)

<table>
<tr><th>Question</th><th>Answer</th><th>Marks</th></tr>
<tr><td></td><td>
One mark for each correct row
<table>
<tr><th>Description</th><th>Structure diagram</th><th>Flowchart</th><th>Library routines</th></tr>
<tr><td>A modelling tool used to show the hierarchy of a system</td><td>✓</td><td></td><td></td></tr>
<tr><td>A collection of standard programs available for immediate use</td><td></td><td></td><td>✓</td></tr>
<tr><td>A graphical representation used to represent an algorithm</td><td></td><td>✓</td><td></td></tr>
<tr><td>A graphical representation to show how a system is broken into sub-systems</td><td>✓</td><td></td><td></td></tr>
</table>
</td><td>4</td></tr>
</table>

Q4)

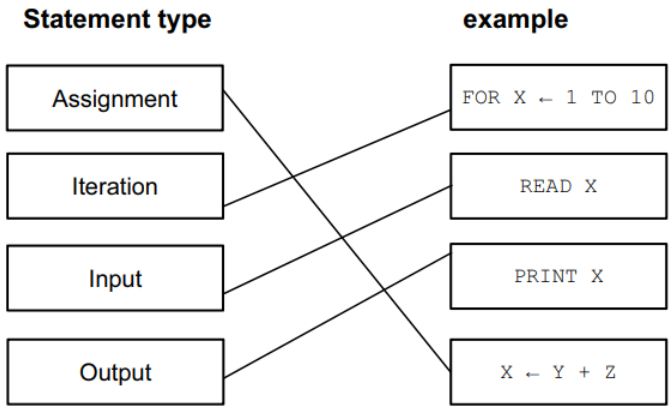
<table>
<tr><th>Question</th><th>Answer</th><th>Marks</th></tr>
<tr><td></td><td>
Constants
Two from:
∞ The value cannot be changed accidentally …
∞ … during the execution of the program
∞ Value only needs to be changed once if circumstances change/during the initialisation process

Variables
Two from:
∞ Stores a value that can change …
∞ … during the execution of the program
∞ Can use a variable without knowing its value

Arrays
Two from:
∞ A list of items of the same data type …
∞ … stored under a single name
∞ To reduce the number of variables used
∞ Any item can be found using an index number to show its place in the list
</td><td>6</td></tr>
</table>

Q5)

1 mark for each correct line, maximum **3** (zero correct **0**, one correct **1**, two correct **2**, three or four correct **3**), each box must have only one connection.



[3]

Q6)

– data structure (one—dimensional) array …….
– …… reason to simplify programming/ make programs shorter, etc. [2]

Q7)

– IF (… THEN … ELSE … ENDIF)

– CASE (… OF … OTHERWISE … ENDCASE) [2]

Q8)

For each example: **1 mark** for **correct structure, 1 mark** for **appropriate content, 1 mark** for **the reason.** There are many correct answers these are only examples

```
IF X > 0 AND X <= 10
  THEN PRINT 'In Range'
  ELSE PRINT 'Out of Range'
ENDIF
```

– e.g. checking a condition that may be complex//uses relational operators// checking for a range of values// only 2 options

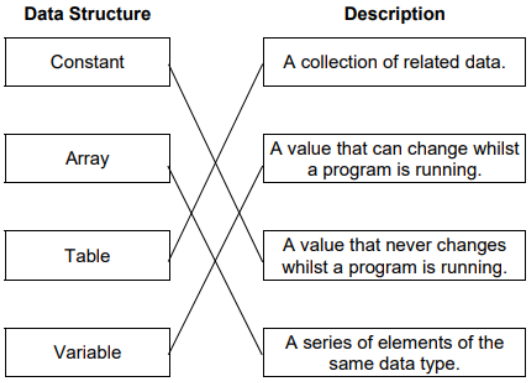
```
CASE X OF
  1 : PRINT 'Option 1'
  2 : PRINT 'Option 2'
  3 : PRINT 'Option 3'
  OTHERWISE PRINT 'Incorrect choice'
ENDCASE
```

– e.g. checking for discrete/large number/more than 2 of values

[6]

Q9)

Question	Answer	Marks
	1 mark for each correct line, max **3** marks.	3



Q10)

Question	Answer	Marks

	2 marks for identification, 1 mark for description, 1 mark for reason.
	
	Identification:
	`CASE …`
	`… OF … OTHERWISE … (ENDCASE)` or
	`… OF … (OTHERWISE) … ENDCASE`
	
	Description:
	– a statement that allows for multiple selections // not any of the above
	
	Reason:
	– to simplify pseudocode/ make pseudocode more understandable etc.
		4

Q11)

Question	Answer	Marks
	One mark for each (max **three**) *10.00* boundary/erroneous data // the price should be rejected // value is out of range *9.99* boundary/extreme/normal data // the prices should be accepted // value is within normal range *ten* erroneous/abnormal data // input should be rejected // value is wrong type	3

Q12)

Question	Answer	Marks
	Answers must be given in context. There are many possible answers. E.g.: Selection use of `IF` statement to check the values of the meter readings (1 mark) `IF Reading > 400 and Reading < 900 THEN …` (1 mark) Repetition use of `FOR` loop to check all 2000 meter readings (1 mark) `FOR Meter = 1 TO 2000 … NEXT` (1 mark)	4

Q13)

Question	Answer	Marks
	Section B	
	Integer – 1 mark for description 1 mark for example e.g. Any whole number … … for example a week number / 26 String – 1 mark for description 1 mark for example e.g. Any data item that contains letters and/or numbers and/or special characters … for example someone's name / def7773@.	**4**

Q14)

Question	Answer	Marks

	Condition controlled loop – 1 mark for each correct answer e.g.
	
	`WHILE Number > 0 DO … ENDWHILE // REPEAT … UNTIL Number > 0`
	
	Conditional statement - 1 mark for each correct answer e.g.
	
	`IF Number = 0 THEN (… ELSE) Number ← 1 ENDIF //`
	`CASE Number OF`
	`0: Number ← 1`
	`(… OTHERWISE) … (ENDCASE)`
	
	Totalling - 1 mark for each correct answer e.g.
	
	`Total ← Total + Number`
		3

Q15)

Question	Answer	Marks
	Many possible answers, those given are examples only. 1 mark for each correct description and 1 mark for each correct example **Char** Description: A single character (from the keyboard) Example: A / # / 2 **String** Description: An (ordered) sequence of characters Example: Hello world / #123?Y / 234 78963 **Boolean** Description: A data type with two possible values Example: TRUE / FALSE	**6**

Q16)

Question	Answer	Marks

(a)	Many possible answers, those given are examples only.
	1 mark per bullet:
	• IF
	• Condition and outcome
	
	Example answer:
	`IF X < 0`
	`  THEN`
	`  PRINT "Negative"`
	`  ELSE`
	`  PRINT "Not negative"`
	`ENDIF`
	
	OR
	
	1 mark per bullet:
	• CASE
	• Condition and outcome
	
	Example answer:
	`CASE X OF`
	`  1: PRINT ("ONE")`
	`  2: PRINT ("TWO")`
	`  OTHERWISE PRINT ("Less than ONE or more than TWO")`
	`ENDCASE`
		2

(b)	• To allow different routes through a program • dependent on meeting certain criteria	2

Q17)

Question	Answer	Marks

Statements	Selection	Repetition

`FOR A ← 1 TO 100`
`  B ← B + 1`
`NEXT A`
		✓

`CASE A OF`
`  100: B ← A`
`  200: C ← A`
`ENDCASE`
	✓	

`IF A > 100`
`  THEN`
`  B ← A`
`ENDIF`
	✓	

`REPEAT`
`  A ← B * 10`
`UNTIL A > 100`
		✓

1 mark for each correct row

4

Q18)

Question	Answer	Marks
	Real Integer Char/String String Boolean	5

Q19)

Question	Answer	Marks
	One mark per mark point, max **four** `DIV`, max **two** • To perform integer division • Meaning only the whole number part of the answer is retained • Example of `DIV` For example `DIV(9,4) = 2` `ROUND`, max **two** • To return a value rounded to a specified number of digits / decimal places • The result will either be rounded to the next highest or the next lowest value • … depending on whether the value of the preceding digit is >=5 or <5 • Example of `ROUND` for example, `ROUND(4.56, 1) = 4.6`	**4**

Q20)

Question	Answer	Marks
	One mark for each correct feature, max **two** **One** mark for each correct accompanying reason, max **two** **For example:** Meaningful identifiers – to enable the programmer (or future programmers) to easily recognize the purpose of a variable / array / constant // to enable easy tracking of a variable / constant / array through the program Use of comments – to annotate each section of a program so that a programmer can find specific sections / so that the programmer knows the purpose of that section of code Procedures and functions – to make programs modular and easier to update / add functionality	**4**

Q21)

Question	Answer	Marks

(a)	**One** mark per mark point, max **three**
	• Storing string in `Phrase`
	• Correct use of `LENGTH` function
	• Correct use of `UCASE` function
	• Correct outputs of `LENGTH` and `UCASE`
	
	For example:
	
	`Phrase ← "The beginning is the most important part"`
	`OUTPUT LENGTH(Phrase)`
	`OUTPUT UCASE(Phrase)`
		3

(b)	**One** mark for each correct line, max **two**
	
	`40`
	`THE BEGINNING IS THE MOST IMPORTANT PART`
		2

Q22)

Question	Answer	Marks

(a)	**One** mark for any two correct lines
	
	`DECLARE P : STRING`
	`P ← "The world"`
	`DECLARE Q : CHAR`
	`Q ← 'W'`
		2

Question	Answer	Marks

(b)	**One** mark for each point (**max four**)
	• converting `P` to upper case
	• finding the length of `P`
	• using a loop to check for position of `Q`
	• using the string operation substring
	• storing the loop counter in `Position` if the value is found
	
	For example:
	`P ← UCASE(P)`
	`Counter ← 1`
	`Position ← 0`
	`REPEAT`
	`  IF SUBSTRING(P, Counter, 1) = Q`
	`  THEN`
	`  Position ← Counter`
	`  ENDIF`
	`  Counter ← Counter + 1`
	`UNTIL Position <> 0 OR Counter = LENGTH(P)`
		4

(c)	5	**1**

Q23)

Question	Answer	Marks
	B	**1**

Q24)

Question	Answer	Marks
	One mark per mark point, max **four** `MOD`, max **two** • To perform (integer) division when one number is divided by another • … and find the remainder • Allow example e.g. `7 MOD 2 = 1` `RANDOM`, max **two** • To generate (pseudo) random numbers • …(usually) within a specified range • Allow example e.g. `RANDOM() * 10` returns a random number between `0` and `10`	**4**

Q25)

Question	Answer	Marks
	One mark per mark point, max **three** MP1 A **call statement** is used in order to make use of a function // the function is called **using its identifier** MP2 Parameters are / may be passed (from the main program) to the function (to be used within the function) MP3 The function performs its task … MP4 … and returns a value / values to the main program	3

Q26)

Question	Answer	Marks
(a)	`DECLARE Saying : STRING`	1

Question	Answer	Marks

b)	**One** mark per mark point, max **five**
	MP1 input a string into `Saying`
	MP2 correct use of `OPENFILE` to write data
	MP3 correct use of `WRITEFILE` to write `Saying`
	MP4 correct use of `CLOSEFILE`
	MP5 correct use of filename `Quotations.txt` throughout
	
	For example:
	
	`INPUT Saying`
	`OPENFILE "Quotations.txt" FOR WRITE`
	`WRITEFILE "Quotations.txt", Saying`
	`CLOSEFILE "Quotations.txt"`
		5

Q27)

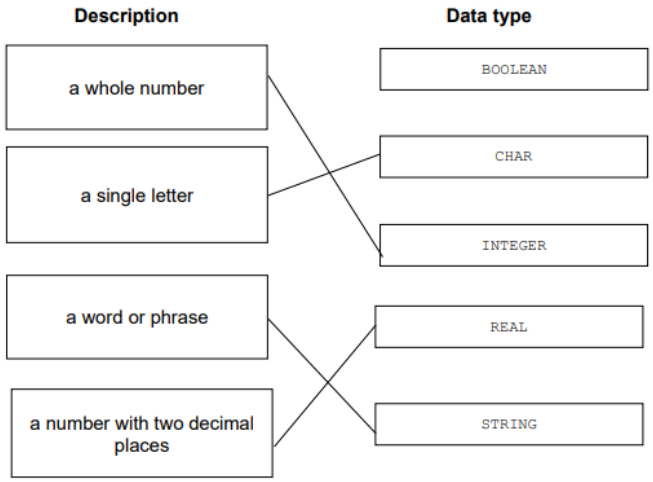
Question	Answer	Marks
	C	1

Q28)

Question	Answer	Marks
	B	1

Q29)

Question	Answer	Marks
	One mark for each correct line from description to data type	4



Q30)

Question	Answer	Marks
	One mark per mark point, max **four** Procedures, max **three** MP1 to enable the programmer to write a collection of programming statements under a single identifier MP2 to allow modular programs to be created // to allow procedures to be re-used within the program or in other programs MP3 to make program creation faster because procedures can be re-used // to enable different programmers to work on different procedures in the same project MP4 to make programs shorter (than using the repeated code) / using less duplication of code // to make programs easier to maintain due to being shorter. Parameters, max **three** MP5 to pass values from the main program to a procedure / function MP6 …so that they can be used in the procedure / function MP7 allow the procedure / function to be re-used **with different data**.	4

Q31)

Question	Answer	Marks
	B	**1**

Q32)

Question	Answer	Marks
(a)	**One** mark for each correct line **Description** → **Programming concept** a subroutine that may **not** return a value → procedure a value that is declared and used within a specific procedure → local variable a value that a procedure expects you to supply when it is called → parameter a subroutine that will always return a value → function	**4**
(b)	**One** mark per mark point • Correct key word - `CALL` • Procedure name with correct parameters - `Average(25, 50)` `CALL Average(25, 50)`	**2**

Question	Answer	Marks
(c)	**One** mark per mark point, **max three** • Procedures / functions divide the program into smaller manageable segments • … making it more readable / easier to understand / easier to debug • Procedures / functions with meaningful names help to provide documentation for the program / enable/help/provide abstraction • Procedures and functions may be re-used (in the program / in other programs / as part of a library) / run from a single line • Procedures and functions can reduce / eliminate (repeated) code	**3**

Q33)

Question	Answer	Marks
	One mark for a correct statement about each data type … … **one** mark for a correct example of data for each data type. Example answer **Integer** A whole number [1] Example 27 [1] **Real** A number that contains a fractional part [1] … Example 18.75 [1]	4

Q34)

Question	Answer	Marks
	One mark for each description, **one** mark for each example • arithmetic – used in calculations (1) `A ← B + C` (1) • Boolean – used for operations with true or false values (1) `IF B AND C` (1) • logical – used in comparisons/conditional statements/selection statements (1) `IF B > C` (1)	6

Q35)

Question	Answer	Marks
	One mark for a correct statement about each data type and **one** mark for a correct example of data for each data type. For example: **String** A group of characters consisting of letters, numbers and special characters [1], Cambridge2024 [1] **Char** A single character [1] X [1]	4

Q36)

Question	Answer	Marks

(a)	**One** mark per mark point
	
	MP1 Assignment of given string to `FullText`
	MP2 Correct use of `SUBSTRING` and assignment of reduced string to own variable
	MP3 Correct use of `UCASE`
	MP4 Output of both strings
	
	Example:
	
	`FullText ← "IGCSE Computer Science at Cambridge"`
	`PartText ← SUBSTRING(FullText, 7, 16)`
	`OUTPUT PartText, UCASE(FullText)`
		4

(b)	**One** mark per mark point
	
	MP1 Opening the correct text file for writing
	MP2 Writing the variable from **part (a)** to the file
	MP3 Closing the text file after writing
	
	Example:
	
	`OPENFILE "Subjects.txt" FOR WRITE`
	`WRITEFILE "Subjects.txt", PartText`
	`CLOSEFILE "Subjects.txt"`
		3

Q37)

Question	Answer	Marks
	C	1

Q38)

Question	Answer	Marks
	One mark for each correct line **Operator** → **Operator type** >= → Logical AND → Boolean DIV → Arithmetic + → Arithmetic	4

Q39)

Question	Answer	Marks

	One mark per mark point (**max six**)
	
	• Declaration of appropriate string variable(s)
	• Input of a line of text
	• Correct use of `LCASE`
	• Correct use of `LENGTH`
	• Opening of at least **one** of the required text files using given names (`Main.txt`, `Lowercase.txt`)
	• Storing of correct line of text to `Main.txt`
	• Storing of correct lines to `Lowercase.txt`
	• Closing of both text files
	• Correct output
	
	For example:
	
	`DECLARE LineOfText : STRING`
	`DECLARE LowercaseText: STRING`
	`INPUT LineOfText`
	`OPENFILE Main.txt FOR WRITE`
	`WRITEFILE Main.txt, LineOfText`
	`CLOSEFILE Main.txt`
	`LowercaseText ← LCASE(LineOfText)`
	`OUTPUT LowercaseText, LENGTH(LineOfText)`
	`OPENFILE Lowercase.txt FOR WRITE`
	`WRITEFILE Lowercase.txt, LowercaseText`
	`CLOSEFILE Lowercase.txt`
		6

Q40)

Question	Answer	Marks
(a)	**One** mark for each point • `06 C ← 1` • `08 W ← W + A[C]` • `11 X ← W / (C - 1) // ROUND(W / (C - 1),0)`	3

(b)	**One** mark for outputting `X`
	One mark for outputting `C - 1`
	One mark for suitable messages
	
	Example:
	
	`12 OUTPUT "Number of values stored in the array is ", C - 1`
	`13 OUTPUT "Average of non-zero elements in the array is ", X`
		3

Question	Answer	Marks
(c)	**One** mark for meaningful identifier for the array `A` **`Values`** **Two** marks for 3 meaningful identifiers for the variables **or** **One** mark for 1 to 2 meaningful identifiers for the variables `C` **`Index`** `X` **`Average`** `W` **`Total`**	3