1 A 1D array, `Product`, of type `STRING` is used to store information about a range of products in a shop. There are 100 elements in the array. Each element stores one data item.

The format of each data item is as follows:

```
<ProductID><ProductName>
```

- `ProductID` is a four-character string of numerals
- `ProductName` is a variable-length string

The following pseudocode is an initial attempt at defining a procedure, `ArraySort`, which will perform a bubble sort on `Product`. The array is to be sorted in ascending order of `ProductID`. Line numbers have been added for identification purposes only.

```
01   PROCEDURE SortArray
02     DECLARE Temp : CHAR
03     DECLARE FirstID, SecondID : INTEGER
04     FOR I ← 1 TO 100
05       FOR J ← 2 TO 99
06          FirstID ← MODULUS(LEFT(Product[J], 6))
07          SecondID ← MODULUS(LEFT(Product[J + 1], 6))
08          IF FirstID > SecondID
09             THEN
10                Temp ← Product[I]
11                Product[I] ← Product[J + 1]
12                Product[J + 1] ← Temp
13        ENDFOR
14          ENDIF
15     ENDFOR
16   ENDPROCEDURE
```

The pseudocode on page 8 contains a number of errors. Complete the following table to show:

- the line number of the error
- the error itself
- the correction that is required.

Note:

- If the same error occurs on more than one line, you should only refer to it ONCE.
- Lack of optimisation should not be regarded as an error.

Line number	**Error**	**Correction**
01	Wrong procedure name – “SortArray”	`PROCEDURE ArraySort`

[10]

2 A sports club maintains a record of the email address of each of its members. The details are stored in a text file, `EmailDetails.txt`. The format of each line of the text file is as follows:

```
<MembershipNumber><EmailAddress>
```

- `MembershipNumber` is a four-character string of numerals
- `EmailAddress` is a variable-length string

When members leave the club their details need to be removed from the file.

A procedure, `RemoveDetails` is required. This will perform the following actions:

1. Input the `MembershipNumber` of the club member to be removed
2. Create a new file, `NewEmailDetails.txt`
3. Copy all the lines from `EmailDetails.txt` to `NewEmailDetails.txt`, except the line with the matching `MembershipNumber`

Write **pseudocode** for the procedure, `RemoveDetails`.

For the built-in functions list, refer to the **Appendix** on page 15.

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

.... [10]

Appendix

Built-in functions (pseudocode)

In each function, if the function call is not properly formed, the function returns an error.

```
MODULUS(x : INTEGER, y : INTEGER) RETURNS INTEGER
```

returns the remainder when `x` is divided by `y` using integer arithmetic.
Example: `MODULUS(5, 2)` will return `1`

```
INT(x : REAL) RETURNS INTEGER
```

returns the integer part of `x`.
Example: `INT(27.5415)` returns `27`

```
LENGTH(ThisString : STRING) RETURNS INTEGER
```

returns the integer value representing the length of string `ThisString`.
Example: `LENGTH("Happy Days")` returns `10`

```
LEFT(ThisString : STRING, x : INTEGER) RETURNS STRING
```

returns leftmost `x` characters from `ThisString`.
Example: `LEFT("ABCDEFGH", 3)` returns string `"ABC"`

```
RIGHT(ThisString : STRING, x : INTEGER) RETURNS STRING
```

returns rightmost `x` characters from `ThisString`.
Example: `RIGHT("ABCDEFGH", 3)` returns string `"FGH"`

```
TONUM(ThisString : STRING) RETURNS INTEGER
```

returns a numeric value equivalent to `ThisString`.
Example: `TONUM("1201")` returns integer value `1201`

Operators (pseudocode)

Operator	Description
`&`	Concatenates (joins) two strings. Example: `"Summer" & " " & "Pudding"` produces `"Summer Pudding"`
`AND`	Performs a logical `AND` on two Boolean values. Example: `TRUE AND FALSE` produces `FALSE`
`OR`	Performs a logical `OR` on two Boolean values. Example: `TRUE OR FALSE` produces `TRUE`

3 The module headers for three modules in a program are defined in pseudocode as follows:

Pseudocode module header
`PROCEDURE Lookup(P4 : INTEGER, BYREF M4 : STRING)`
`FUNCTION Update(T4 : INTEGER) RETURNS INTEGER`
`FUNCTION Validate(S2 : INTEGER, P3 : STRING) RETURNS BOOLEAN`

A fourth module, `Renew()`, calls the three modules in the following sequence.

```
Validate()
Lookup()
Update()
```

Draw a structure chart to show the relationship between the four modules and the parameters passed between them.

[8]

4 The following pseudocode algorithm checks whether a string is a valid email address.

```
FUNCTION Check(InString : STRING) RETURNS BOOLEAN

   DECLARE Index : INTEGER
   DECLARE NumDots : INTEGER
   DECLARE NumAts : INTEGER
   DECLARE NextChar : CHAR
   DECLARE NumOthers : INTEGER

   NumDots ← 0
   NumAts ← 0
   NumOthers ← 0

   FOR Index ← 1 TO LENGTH(InString)

      NextChar ← MID(InString, Index, 1)
      CASE OF NextChar
         '.': NumDots ← NumDots + 1
         '@': NumAts ← NumAts + 1
         OTHERWISE NumOthers ← NumOthers + 1
      ENDCASE

   ENDFOR

   IF (NumDots >= 1 AND NumAts = 1 AND NumOthers > 5)
      THEN
         RETURN TRUE
      ELSE
         RETURN FALSE
   ENDIF

ENDFUNCTION
```

(a) Describe the validation rules that are implemented by this pseudocode. Refer **only** to the contents of the string and **not** to features of the pseudocode.

....

....

....

.... [4]

(b) (i) Complete the trace table by dry running the function when it is called as follows:

```
Result ← Check("Jim.99@skail.com")
```

Index	**NextChar**	**NumDots**	**NumAts**	**NumOthers**

[7]

(ii) State the value returned when function `Check` is called as shown in **part (b)(i)**.

.... [1]

5 A text file, Library.txt, stores information relating to a book collection. The file stores four pieces of information about each book on separate lines of the file, as follows:

```
Line n:       <Book Title>
Line n + 1:   <Author Name>
Line n + 2:   <ISBN>
Line n + 3:   <Location>
```

Information is stored as data strings.

Information relating to two books is shown:

File line	Data
100	"Learning Python"
101	"Brian Smith"
102	"978-14-56543-21-8"
103	"BD345"
104	"Surviving in the mountains"
105	"C T Snow"
106	"978-35-17635-43-9"
107	"ZX001"

(a) (i) A function, FindBooksBy(), will search Library.txt for all books by a given author.

The function will store the Book Title and Location in the array Result, and will return a count of the number of books found.

Array Result is a global 2D array of type STRING. It has 100 rows and 2 columns.

Write **pseudocode** to declare the array Result.

....

....

.... [4]

(ii) Function FindBooksBy() will:

- receive the Author Name as a parameter
- search Library.txt for matching entries
- store the Book Title and Location of matching entries in the Result array
- return an integer value giving the number of books by the author that were found.

Write **program code** for the function `FindBooksBy()`.

Visual Basic and Pascal: You should include the declaration statements for variables.
Python: You should show a comment statement for each variable used with its data type.

Programming language ....

Program code

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

.... [6]