QUESTION 7.

5 A firm employs workers who assemble amplifiers. Each member of staff works a... of hours each day.

The firm records the number of completed amplifiers made by each employee each day.

Management monitor the performance of all its workers.

Production data was collected for 3 workers over 4 days.

	Daily hours worked
Worker 1	5
Worker 2	10
Worker 3	10

Production data

	Worker 1	**Worker 2**	**Worker 3**
Day 1	10	20	9
Day 2	11	16	11
Day 3	10	24	13
Day 4	14	20	17

A program is to be written to process the production data.

(a) The production data is to be stored in a 2-dimensional array `ProductionData`, declared as follows:

```
DECLARE ProductionData ARRAY[1:4, 1:3] : INTEGER
```

(i) Describe **two** features of an array.

1 .....

.....

2 .....

.....[2]

(ii) Give the value of `ProductionData[3, 2]`.

.....[1]

(iii) Describe the information produced by the expression:

```
ProductionData[2, 1] + ProductionData[2, 2] + ProductionData[2, 3]
```

.....

.....[2]

(b) Complete the trace table for the pseudocode algorithm below.

```
FOR WorkerNum ← 1 TO 3
   WorkerTotal[WorkerNum] ← 0
ENDFOR

FOR WorkerNum ← 1 TO 3
   FOR DayNum ← 1 TO 4
      WorkerTotal[WorkerNum] ← WorkerTotal[WorkerNum] +
                                    ProductionData[DayNum, WorkerNum]
   ENDFOR
ENDFOR

FOR WorkerNum ← 1 TO 3
   WorkerAverage ← WorkerTotal[WorkerNum]/
                                    (4 * DailyHoursWorked[WorkerNum])
   IF WorkerAverage < 2
      THEN
         OUTPUT "Investigate", WorkerNum
   ENDIF
ENDFOR
```

WorkerNum	DayNum	WorkerAverage	OUTPUT

WorkerTotal		
1	2	3

[8]

(c) An experienced programmer suggests that the pseudocode would be best in procedure `AnalyseProductionData`.

Assume that both arrays, `DailyHoursWorked` and `ProductionData`, are availab procedure from the main program and they are of the appropriate size.

```
PROCEDURE AnalyseProductionData(NumDays : INTEGER, NumWorkers : INTEGER)

   DECLARE .....

   DECLARE .....

   DECLARE .....

   DECLARE .....

   FOR WorkerNum ← 1 TO 3
      WorkerTotal[WorkerNum] ← 0
   ENDFOR

   FOR WorkerNum ← 1 TO 3
      FOR DayNum ← 1 TO 4
         WorkerTotal[WorkerNum] ← WorkerTotal[WorkerNum] +
                                        ProductionData[DayNum, WorkerNum]
      ENDFOR
   ENDFOR

   FOR WorkerNum ← 1 TO 3
      WorkerAverage ← WorkerTotal[WorkerNum]/
                                       (4 * DailyHoursWorked [WorkerNum])
      IF WorkerAverage < 2
         THEN
            OUTPUT "Investigate", WorkerNum
      ENDIF
   ENDFOR

ENDPROCEDURE
```

(i) Complete the declaration statements showing the local variables. [4]

(ii) The original pseudocode has been 'pasted' under the procedure header.

Circle all the places in the original pseudocode where changes will need to be made.

Write the changes which need to be made next to each circle. [3]

(iii) Write the statement for a procedure call which processes data for 7 days for 13 workers.

.....[1]

BLANK PAGE

16

BLANK PAGE

QUESTION 8.

5 A program is to process a set of integers.

The integers are stored in an array, `Num`. The first `N` elements are to be processed.

The pseudocode for this program is shown below:

```
FOR i ← 1 TO (N - 1)
   j ← 1
   REPEAT
      IF Num[j] > Num[j + 1]
         THEN
            Temp ← Num[j]
            Num[j] ← Num[j + 1]
            Num[j + 1] ← Temp
      ENDIF
      j ← j + 1
   UNTIL j = (N - i + 1)
ENDFOR
```

(a) (i) Trace the execution of the pseudocode for the value `N` = 5 and the given array of integers.

N	i	j	Temp		Num 1	2	3	4	5
5					11	16	13	7	8

[8]

(ii) State the purpose of the algorithm.

.....

.....

(iii) Describe what evidence from the trace table suggests that the given pseudocode is inefficient.

.....

.....[1]

(b) Complete the identifier table documenting the use of each of the variables.

Identifier	Data type	Description
`Num`	`ARRAY[1:100] OF INTEGER`	The array of numbers.
`N`		
`i`		
`j`		
`Temp`		

[5]

QUESTION 9.

3 A string encryption function is needed. The encryption uses a simple character-sub

In this method, a new character substitutes for each character in the original string. This the encrypted string.

The substitution uses the 7-bit ASCII value for each character. This value is used as an index to 1D array, `Lookup`, which contains the substitute characters.
`Lookup` contains an entry for each of the ASCII characters. It may be assumed that the original string and the substitute characters are all printable.

For example:

- 'A' has ASCII value 65
- Array element with index 65 contains the character 'Y' (the substitute character)
- Therefore, 'Y' substitutes for 'A'
- There is a different substitute character for every ASCII value

The programmer writes a function, `EncryptString`, to return the encrypted string. This function will receive two parameters, the original, `PlainText` string and the 1D array.

(a) The first attempt at writing the pseudocode for this function is shown below.

Complete the pseudocode.

For the built-in functions list, refer to the **Appendix** on page 18.

```
FUNCTION EncryptString(.............................) RETURNS STRING

    DECLARE ..................... , ...................... : CHAR

    DECLARE OldCharValue : ....................................

    DECLARE n : INTEGER

    DECLARE OutString : STRING


    ............................. //initialise the return string


    //loop through PlainText to produce OutString

    FOR n ← 1 TO ................. //from first to last character

       OldChar ← ...................//get next character

       OldCharValue ← ..............//find the ASCII value

       NewChar ← ...................//look up substitute character

       ..............................//concatenate to OutString

    ENDFOR


    ..............................................................

ENDFUNCTION
```

[10]

(b) Additional code needs to be written to allow the user to change some of the
array `Lookup`.

The user will input:

- the array start position
- the number of elements to change
- each new substitute character

At the end, the program will finally output a confirmation message.

The first version of the algorithm is represented by the flowchart on the following page.

(i) Write **program code** to declare the array `Lookup`.

Programming language .....

.....

.....

.....[2]

(ii) Write **program code** to implement the flowchart design.

In addition to the `Lookup` array, assume that the following variables have been declared:

```
StartPos, NumToChange, n, NewChar
```

Programming language .....

.....

.....

.....

.....

.....

.....

.....

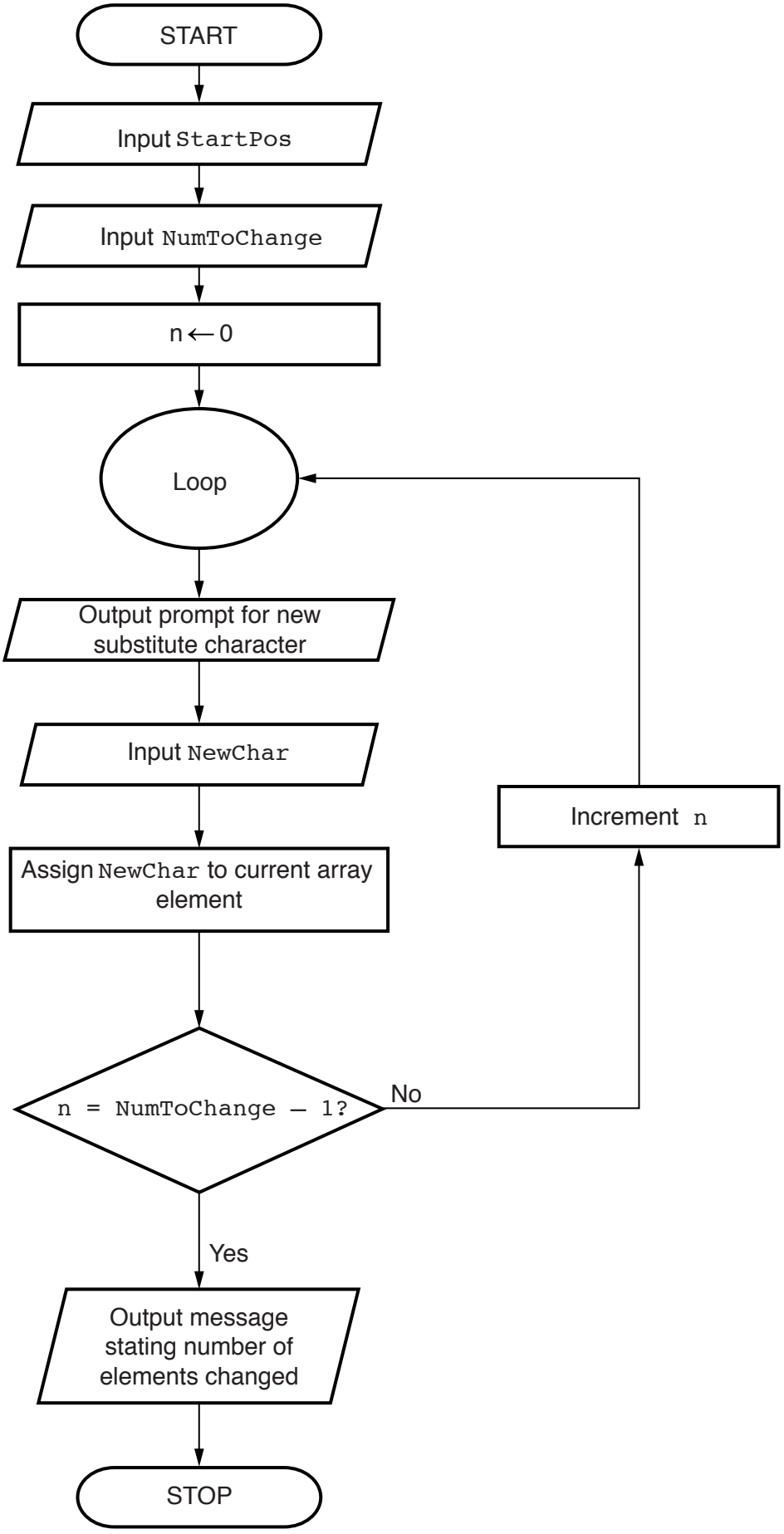
.....

.....

.....

.....

.....[6]


START
Input StartPos
Input NumToChange
n ← 0
Loop
Output prompt for new substitute character
Input NewChar
Assign NewChar to current array element
n = NumToChange – 1?
No
Increment n
Yes
Output message stating number of elements changed
STOP

QUESTION 10.

3 String encryption was implemented using a simple character-substitution met
`Decrypt`, is needed to reverse the encryption process and return the original chara

The encryption uses the 7-bit ASCII value for each character. This value is used as an in
1D array, `Lookup`, which contains the substitute characters.
`Lookup` contains an entry for each of the ASCII characters.

This function, `Decrypt`, will accept two parameters, a single character, `CipherChar`, and the 1D array, `Lookup`.

The steps involved in `Decrypt` are follows:

- Search for the character in the array
- Note the index value where the character is found (the index value is the ASCII value of the original character)
- Use the index value to obtain the original character

(a) The first attempt at writing the pseudocode for this function is shown below.

Complete the pseudocode.

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

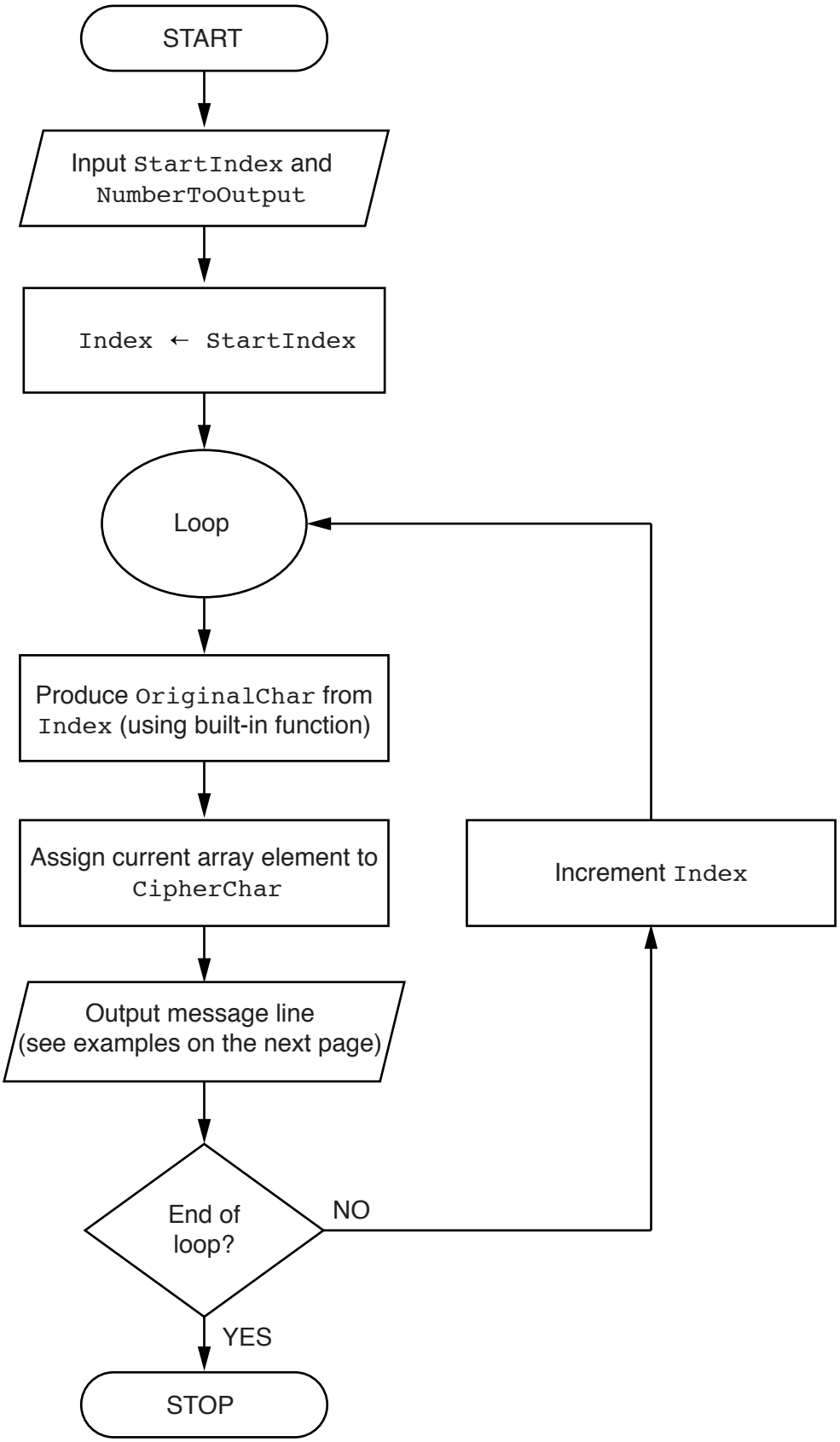
```
FUNCTION Decrypt(..... , .....) RETURNS CHAR

   DECLARE Found      : .....

   DECLARE ..... : INTEGER

   DECLARE OriginalChar      : CHAR

   Index ← 1         // .....

   Found ← FALSE
   //search for CipherChar in Lookup:

   WHILE .....
      //compare CipherChar with this array element:

      IF .....
         THEN
            .....//Set the flag
         ELSE
            Index .....//Move to next array element
      ENDIF
   ENDWHILE
   //dropped out of loop so must have found CipherChar:

   .....//convert Index to original character

   RETURN .....

ENDFUNCTION
```

[11]

Question 3 continues on page 8.

(b) A program is to be written to output part of the `Lookup` array.

The design of the algorithm is shown below.
It may be assumed that the characters output from `Lookup` are all printable.



For example, for the input of 65 and 3, the output will be:

```
Index 65: Character A has substitute character Y
Index 66: Character B has substitute character Q
Index 67: Character C has substitute character F
```

Write **program code** to implement the flowchart design.

In addition to the `Lookup` array, assume that the following variables have been declared:

`StartIndex, NumberToOutput, Index`

Programming language .....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

[6]

QUESTION 11.

4 A company employs Ahmed as a programmer.

(a) At College, before joining the company, Ahmed used two items of software for pro

- a text editor
- a compiler

Describe how he could have developed programs using these software tools.

Include in the description the terms 'object code' and 'source code'.

.....

.....

.....

.....

.....

.....

.....[3]

(b) Ahmed now uses an Integrated Development Environment (IDE) for programming.

(i) State **one** feature an IDE provides to help with the identification of syntax errors.

.....

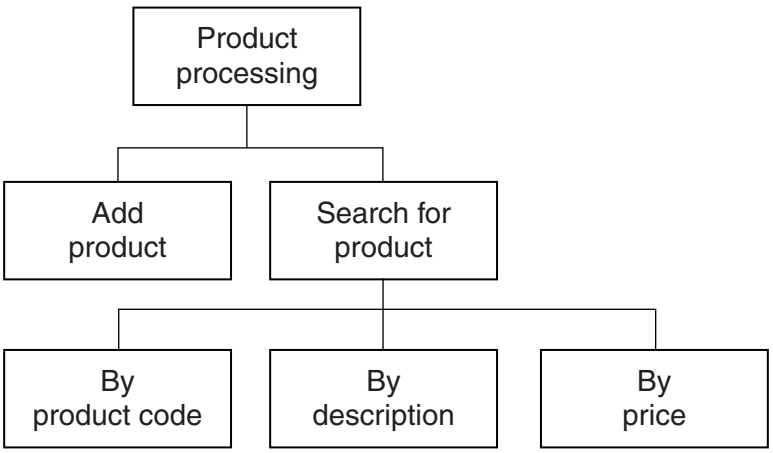
.....[1]

(ii) State **one** feature an IDE provides to carry out white box testing.

.....

.....[1]

(c) The company maintains a file of product data. Ahmed is to write a program to add a new product and search for a product based on the structure diagram shown:



The program records the following data for each product:

- product code
- product description
- product retail price

The text file `PRODUCTS` stores each data item on a separate line, as shown below:

File **PRODUCTS**

0198
Plums(10kg)
11.50
0202
Onions(20kg)
10.00
~
0376
Mango chutney(1kg)
02.99
~
0014
Mango(10kg)
12.75

The program uses the variables shown in the identifier table.

Identifier	Data type	Description
PRODUCTS	TEXT FILE	Storing the code, description and retail price for all current products
PCode	ARRAY[1:1000] OF STRING	Array storing the product codes
PDescription	ARRAY[1:1000] OF STRING	Array storing the product descriptions
PRetailPrice	ARRAY[1:1000] OF REAL	Array storing the product retail prices
i	INTEGER	Array index used by all three arrays

(i) The first operation of the program is to read all the product data held i
and write them into the three 1D arrays.

Complete the pseudocode below.

```
OPEN .....

i ← 1

WHILE .....

  READFILE ("PRODUCTS", .....)

  READFILE ("PRODUCTS", .....)

  READFILE ("PRODUCTS", .....)

  .....

  .....

ENDWHILE

CLOSE "PRODUCTS"

OUTPUT "Product file contents written to arrays"
```

[5]

When Ahmed designed the `PRODUCTS` file, he considered the alternative file structure shown opposite.

It stores one product per line in the text file.

File **PRODUCTS**

0198 Plums(10kg)	11.50
0202 Onions(20kg)	10.00
~	
0376 Mango chutney(1kg)	02.99
~	
0014 Mango(10kg)	12.75

(ii) State **one** benefit and **one** drawback of this file design.

Benefit .....

.....

Drawback .....

.....[2]

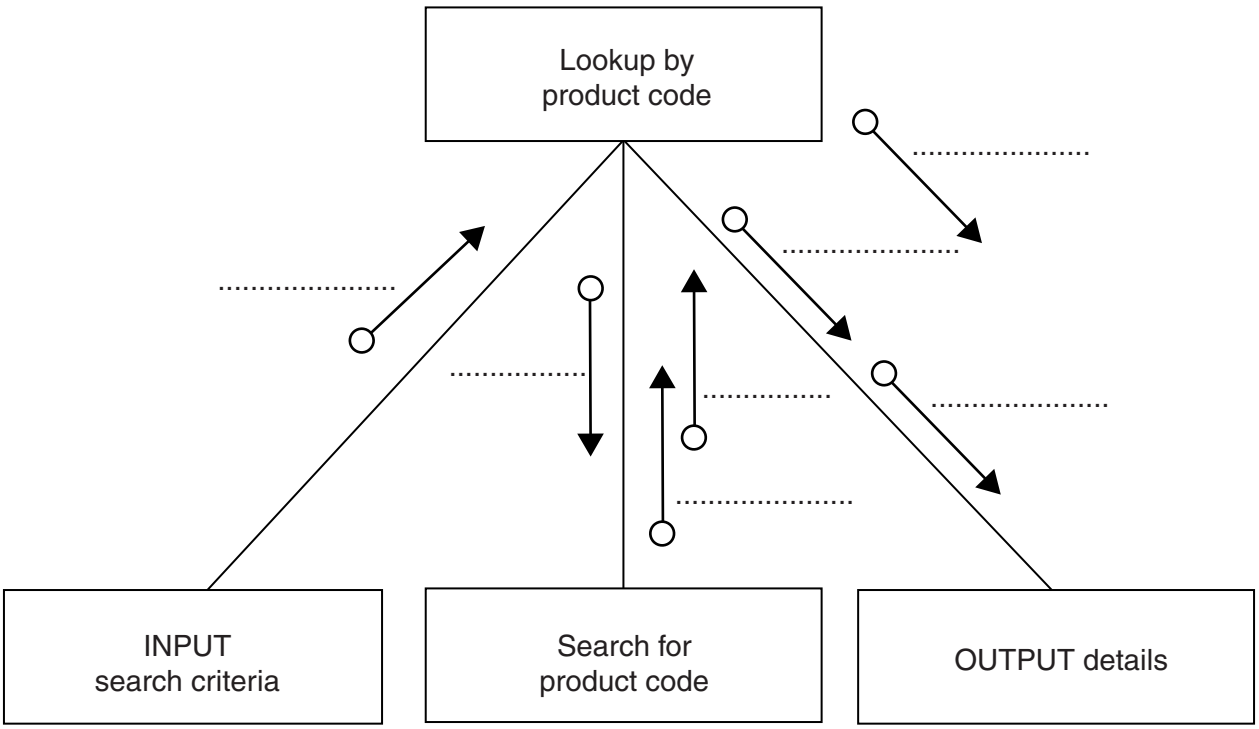
(d) To code the 'Search by product code' procedure, Ahmed draws a structure c… different stages.

The procedure uses the variables shown in the identifier table.

Identifier	Data type	Description
`SearchCode`	`STRING`	Product code input by the user
`ThisIndex`	`INTEGER`	Array index position for the corresponding product
`ThisDescription`	`STRING`	Product description found
`ThisRetailPrice`	`REAL`	Product retail price found

You can assume that before the procedure is run, all the product data is read from file `PRODUCTS` and then stored in three 1D arrays as described in **part (c)(i)**.

Label the structure chart to show the input(s) and output(s).



[4]

(e) A first attempt was made at writing the 'Search for product code' module.
Ahmed designs this as a function `ProductCodeSearch`.

The function returns an integer value as follows:

- if the product code is found, it returns the index position of the 1D array `PCode` be searched
- if the product code is not found, the function returns -1

Write **program code** for function `ProductCodeSearch`.

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 .....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....[6]

QUESTION 12.

5 A team keeps a record of the scores made by each of their eight players in a num

The data in the two tables below shows:

- the scores of the eight players after twenty games
- the eight player names.

	1	2	3		8
1	12	17	67		31
2	35	82	44		29
3	61	39	80		17
4	81	103	21		11
5	56	0	98		4
…					
19	45	6	81		77
20	12	11	3		6

1	Vorma
2	Ravi
3	Chada
4	Nigam
5	Bahri
6	Smith
7	Goyal
8	Lata

The team wants a computer program to input and record the player data.

(a) A programmer designs the following pseudocode for the input of a player's score from one game.

```
01  INPUT GameNumber
02  INPUT PlayerNumber
03  INPUT PlayerGameScore
04  PlayerScore[GameNumber, PlayerNumber] ← PlayerGameScore
```

Describe the data structure the programmer has used for the storage of all player scores.

..... [2]

(b) The player names are permanently stored in a text file `NAMES.TXT`, with on line. The player names will be read by the program and stored in a 1D array.

The design given in **part (a)** will be expanded so that the user is prompted for th name instead of the player number. Step `02` now becomes:

```
02.1    Read the player names from file NAMES.TXT into the array PlayerName
02.2    INPUT ThisPlayerName
02.3    Search the PlayerName array for ThisPlayerName to find the PlayerNumber
```

(i) State the computing term for the expansion of one or more steps in the original design.

..... [1]

(ii) Write the **program code** for step `02.1`

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 .....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [4]

(iii) Program code is to be designed and written for step `02.3`

The program will use these identifiers:

Identifier	Data type	Description
`PlayerName`	`ARRAY[1 : 8] OF STRING`	Stores the player names (read from the file)
`ThisPlayerName`	`STRING`	Input by the user (step `02.2`)
`Found`	`BOOLEAN`	Flags when `ThisPlayerName` is found when searching the `PlayerName` array
`i`	`INTEGER`	Array index

Write **program code** to carry out the linear search for step `02.3`

There is no requirement to declare or comment on variables used.

Programming language .....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [4]

(c) The team wants the program to produce a report, with the following specifica

The program outputs the total number of player scores that are:

- 50 and over but less than 100
- 100 or higher.

You can assume that before the section runs, the program has assigned all eight player scores to the `PlayerScore` data structure.

A first attempt at the pseudocode is shown below:

```
01 Total50 ← 0
02 Total100 ← 0
03 FOR PlayerIndex  ← 1 TO 8
04   FOR GameIndex ← 1 TO 20
05     IF PlayerScore[GameIndex, PlayerIndex] > 100
06       THEN
07         Total100 ← Total100 + 1
08       ELSE
09         IF PlayerScore[GameIndex, PlayerIndex] > 50
10           THEN
11             Total50 ← Total50 + GameIndex
12         ENDIF
13     ENDIF
14   ENDFOR
15 ENDFOR
16 OUTPUT Total50
17 OUTPUT Total100
```

(i) Describe the control structure used in lines `03` and `04` and lines `14` and `15`.

.....

.....

..... [2]

(ii) Consider the following two statements.

Write either TRUE **or** FALSE next to each statement.

Statement	TRUE or FALSE
The pseudocode considers all the scores for a player, before progressing to the next player.	
The pseudocode considers all scores in a game, before progressing to the next game.	

[1]

(iii) The programmer has made logic errors in the design.

State a line number at which an error occurs.

Explain the error or write the corrected pseudocode statement.

Line number .....

Explanation .....

..... [1]

QUESTION 13.

5 A multi-user computer system records user login information in a text file, `Login`... time a user successfully logs into the system, the following information is recorded:

Item	Information	Example data
1	A five character user ID	`"JimAA"`
2	A four character port ID	`"3456"`
3	A fourteen character time and date	`"08:30Jun012015"`

The data items are concatenated to form a single string. Each string is saved as a separate line in the text file.

The example data in the preceding table would result in the following text line in the file:

"JimAA345608:30Jun012015"

The computer system can produce a list of the successful login attempts by a given user.

The file `LoginFile.txt` is searched for a given user ID and the corresponding data are copied into a 2D array, `LoginEvents`.

`LoginEvents` has been declared in pseudocode as:

```
DECLARE LoginEvents[1 : 1000, 1 : 2] OF STRING
```

A procedure, `SearchFile`, is needed to search the file and copy selected data to the array.

The main steps of the procedure are as follows:

- Input a user ID.
- Search `LoginFile.txt` for entries with matching user ID.
- For matching entries, copy items 2 and 3 above into the `LoginEvents` array.

You can assume that:

- the system initialises all elements of `LoginEvents` to an empty string " ", before it calls `SearchFile`
- there will be no more than 1000 successful logins for a single user.

Write **program code** for the procedure `SearchFile`.

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

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....[10]

QUESTION 14.

5 A multi-user computer system records user login data. Each time a user success
system, it records the following data.

Data item	Example data
User ID	`"Jim27"`
Port ID	`"3456"`
Time and date	`"08:30 Jun 01 2015"`

The data items are concatenated (joined) using a separator character to form a single string. Each string represents one log entry.

(a) (i) Suggest a suitable separator character. Give the reason for your choice.

Character .....

Reason .....

.....

[2]

(ii) The concatenated strings are stored in an array, `LogArray`, which may contain up to 20 log entries.

Use **pseudocode** to declare `LogArray`.

..... [2]

(b) A procedure, `LogEvents`, is required to add the log entry data from `LogArra` text file, `LoginFile.txt`.

Unused array elements are assigned the value `"****"`. These can occur anywhe array and should not be written to the file.

Write **program code** for the procedure `LogEvents`.

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

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [8]

QUESTION 15.

3 (a) A multi-user computer system stores information about users. It use
`UserNameArray`, of type `STRING`. There are 100 elements in the array.

The format of the string in each element of the array is as follows:

```
<UserID><UserName>
```

- `UserID` is a six-character string of numerals.
- `UserName` is a variable-length string.

Write **pseudocode** for a procedure, `BubbleSort`, to perform an efficient bubble sort on `UserNameArray`. The array is to be sorted in ascending order of `UserID`.

You should assume that `UserNameArray` has been declared as a global variable.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....[8]

(b) The value of `UserID` should be unique for each user but a problem ha
repeated `UserID` values may have been issued.

The array is sorted by `UserID`, so any repeated `UserID` values will appear in co
array elements.

A procedure, `FindRepeats` is required.

This will:

- compare each element with the previous element and output the `UserID` and `UserName` if the `UserID` is repeated
- output the total number of `UserID`s that are repeated.

For example, the `UserNameArray` contains the following entries.

Array element	Comment
	
`122222Jim Moriarty`	
	
`123456Fred Smith`	
`123456Eric Sykes`	Repeated User ID
`123456Kevin Turvey`	Repeated User ID
	
`222244Alice Chan`	
`222244Myra Singh`	Repeated User ID
	
`333333Yasmin Halim`	
	

For this example, the output is:

```
123456Eric Sykes
123456Kevin Turvey
222244Myra Singh
There are 3 repeated UserIDs
```

If no repeated `UserID`s are found, the output is:

```
The array contains no repeated UserIDs
```

Write **program code** for the procedure, `FindRepeats`.
You should assume that `UserNameArray` has been declared as a global varia

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 typ

Programming language .....

Program code

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....[8]

(c) (i) The `FindRepeats` procedure forms part of a program.

Name **three** stages in a program development cycle.

1 .....

2 .....

3 .....

[3]

(ii) The program containing `FindRepeats` will be created using an IDE.

State what is meant by IDE.

.....

.....[1]

(iii) Name **two** features provided by an IDE that assist in the program development cycle.

1 .....

.....

2 .....

.....

[2]

(iv) The procedure, `FindRepeats`, is written assuming there are 100 elements in `UserNameArray`.

In the main program, the global array, `UserNameArray`, has been declared with only 50 elements.

State the type of error this will cause.

.....[1]

QUESTION 16.

3 A 1D array, `Product`, of type `STRING` is used to store information about a range 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 t

- 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`

[8]

QUESTION 17.

4 Part of a program written in pseudocode is shown.

```
01  DECLARE NumElements : INTEGER
...
10  FUNCTION ScanArray(SearchString : STRING) RETURNS INTEGER
11
12     DECLARE ArrayIndex : INTEGER
13     DECLARE ArrayString : STRING
14     DECLARE NumberFound : INTEGER
15
16     ArrayIndex ← 0
17     NumberFound ← 0
18
19     FOR ArrayIndex ← 1 TO NumElements
20        ArrayString ← ResultArray[ArrayIndex, 1]
21        IF ArrayString = SearchString
22           THEN
23              CALL SaveToFile(ArrayString)
24              NumberFound ← NumberFound + 1
25        ENDIF
26     ENDFOR
27
28     RETURN NumberFound
29
30  ENDFUNCTION
```

(a) (i) Examine the pseudocode **and** complete the following table.

	Answer
The identifier name of a global integer	
The identifier name of a user-defined procedure	
The line number of an unnecessary statement	
The scope of `ArrayString`	

[4]

(ii) Describe in detail the purpose of lines `19` to `26` in the function `ScanArray()`.
Do **not** use pseudocode in your answer.

.....

.....

.....

.....

.....

.....

.....[4]

(b) The function `ScanArray()` needs to be amended so that the compari
sensitive. For example, comparing "`Aaaa`" with "`AAAa`" should evaluate to `TRUE`

Write **program code** to implement the **amended** `ScanArray()` function.

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]

(c) The function `ScanArray()` is one of a number of sub-tasks within a progra

Name the process that involves the splitting of a problem into sub-tasks **and** advantages of this approach.

Name .....

Advantage 1 .....

.....

Advantage 2 .....

.....

[3]

(d) `ResultArray` is a 2D array of type `STRING`. It represents a table containing 100 rows and 2 columns.

Write **program code** to declare `ResultArray` **and** set all elements to the value `'*'`.

Programming language .....

Program code

.....

.....

.....

.....

.....

.....

.....

.....[3]

Question 5 begins on the next page.

QUESTION 18.

4 The following pseudocode is a string handling function.

For the built-in functions list, refer to the **Appendix** on page 16.

```
FUNCTION Clean(InString : STRING) RETURNS STRING

   DECLARE NewString : STRING
   DECLARE Index : INTEGER
   DECLARE AfterSpace : BOOLEAN
   DECLARE NextChar : CHAR
   CONSTANT Space = ' '

   AfterSpace ← FALSE
   NewString ← ""

   FOR Index ← 1 TO LENGTH(InString)
      NextChar ← MID(InString, Index, 1)
      IF AfterSpace = TRUE
         THEN
            IF NextChar <> Space
               THEN
                  NewString ← NewString & NextChar
                  AfterSpace ← FALSE
            ENDIF
         ELSE
            NewString ← NewString & NextChar
            IF NextChar = Space
               THEN
                  AfterSpace ← TRUE
            ENDIF
      ENDIF
   ENDFOR

   RETURN NewString

ENDFUNCTION
```

(a) (i) Complete the trace table by performing a dry run of the function whe
follows:

```
Result ← Clean("X∇∇∇Y∇and∇∇Z")
```

The symbol `'∇'` represents a space character. Use this symbol to represent a spa
character in the trace table.

Index	AfterSpace	NextChar	NewString

[6]

(ii) State the effect of the function `Clean()`.

.....

..... [1]

(iii) The pseudocode is changed so that the variable `AfterSpace` is initialis

Explain what will happen if the function is called as follows:

```
Result ← Clean("∇∇X∇∇∇Y∇and∇∇Z")
```

.....

.....

.....

..... [2]

(b) The following pseudocode declares and initialises an array.

```
DECLARE Code : ARRAY[1:100] OF STRING
DECLARE Index : INTEGER

FOR Index ← 1 TO 100
   Code[Index] ← ""
ENDFOR
```

The design of the program is changed as follows:

- the array needs to be two dimensional, with 500 rows and 4 columns
- the elements of the array need to be initialised to the string `"Empty"`

Re-write the **pseudocode** to implement the new design.

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [4]

(c) State the term used for changes that are made to a program in response to a specification change.

..... [1]

Question 5 begins on the next page.

QUESTION 19.

6 A text file, `StudentContact.txt`, contains a list of names and telephone num in a school. Not all students in the school have provided a contact telephone numbe their name will not be in the file.

Each line of the file is stored as a string that contains a name and telephone number, separat the asterisk character (`'*'`) as follows:

`<Name>'*'<TelNumber>`, for example:

`"Bill Smith*081234567"`

A 1D array, `ClassList`, contains the names of students in a particular class. The array consists of 40 elements of string data type. You can assume that student names are unique.
Unused elements contain the empty string `""`.

A program is to be written to produce a **new** text file, `ClassContact.txt`, containing student names and numbers for all students in a particular class.

For each name contained in the `ClassList` array, the program will:

- search the `StudentContact.txt` file
- copy the matching string into `ClassContact.txt` if the name is found
- write the name together with "*No number" into `ClassContact.txt` if the name is not found.

The program will be implemented as three modules. The description of these is as follows:

Module	Description

`ProcessArray()`	• Check each element of the array:
	  ○ Read the student name from the array
	  ○ Ignore unused elements
	  ○ Call `SearchFile()` with the student name
	  ○ If the student name is found, call `AddToFile()` to write the student details to the class file
	  ○ If the student name is not found, call `AddToFile()` to write a new string to the class file, formed as follows:
	  <Name>"*No number"
	• Return the number of students who have not provided a telephone number

`SearchFile()`	• Search for a given student name at the start of each line in the file `StudentContact.txt`:
	  ○ If the search string is found, return the text line from `StudentContact.txt`
	  ○ If the search string is not found, return an empty string

`AddToFile()`	• Append the given string to a specified file, for example, `AddToFile(StringName, FileName)`

(a) Write **program code** for the module `SearchFile()`.

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

Programming language .....

Program code

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [8]

(b) Write **pseudocode** for the module `ProcessArray()`.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [9]

(c) `ProcessArray()` is modified to make it general purpose. It will now be parameters as follows:

- an array
- a string representing the name of a class contact file

It will still return the number of students who have not provided a contact telephone number.

Write **program code** for the header (declaration) of the modified `ProcessArray()`.

Programming language .....

Program code

.....

.....

.....

..... [3]

Appendix

Built-in functions (pseudocode)

Each function returns an error if the function call is not properly formed.

`MID(ThisString : STRING, x : INTEGER, y : INTEGER) RETURNS STRING`
returns a string of length `y` starting at position `x` from `ThisString`

Example: `MID("ABCDEFGH", 2, 3)` returns `"BCD"`

`LENGTH(ThisString : STRING) RETURNS INTEGER`
returns the integer value representing the length of `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 `"ABC"`

`RIGHT(ThisString : STRING, x : INTEGER) RETURNS STRING`
returns rightmost `x` characters from `ThisString`

Example: `RIGHT("ABCDEFGH", 3)` returns `"FGH"`

`INT(x : REAL) RETURNS INTEGER`
returns the integer part of `x`

Example: `INT(27.5415)` returns `27`

`NUM_TO_STRING(x : REAL) RETURNS STRING`
returns a string representation of a numeric value.

Example: `NUM_TO_STRING(x)` returns `"87.5"` if `x` has the value `87.5`
Note: This function will also work if `x` is of type `INTEGER`

`STRING_TO_NUM(x : STRING) RETURNS REAL`
returns a numeric representation of a string.

Example: `STRING_TO_NUM(x)` returns `23.45` if `x` has the value `"23.45"`
Note: This function will also work if `x` is of type `CHAR`

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`

BLANK PAGE

QUESTION 20.

6 Account information for users of a library is held in one of two text files; `UserLis` `UserListNtoZ.txt`

The format of the data held in the two files is identical. Each line of the file is stored as a s contains an account number, name and telephone number separated by the asterisk cha ('*') as follows:

```
<Account Number>'*'<Name>'*'<Telephone Number>
```

An example of one line from the file is:

```
"GB1234*Kevin Mapunga*07789123456"
```

The account number string may be **six** or **nine** characters in length and is **unique for each person**. It is made up of alphabetic and numeric characters only.

An error has occurred and the same account number has been given to different users in the two files. There is **no** duplication of account numbers **within each individual file**.

A program is to be written to search the two files and to identify duplicate entries. The account number of any duplicate found is to be written to an array, `Duplicates`, which is a 1D array of 100 elements of data type `STRING`.

The program is to be implemented as several modules. The outline description of three of these is as follows:

Module	Outline description
`ClearArray()`	• Initialise the global array `Duplicates`. Set all elements to the empty string.

`FindDuplicates()`	• Read each line from the file `UserListAtoM.txt`
	  ◦ Check whether the account number appears in file `UserListNtoZ.txt` using `SearchFileNtoZ()`
	  ◦ If the account number does appear then add the account number to the array.
	• Output an error message and exit the module if there are more duplicates than can be written to the array.

`SearchFileNtoZ()`	• Search for a given account number in file `UserListNtoZ.txt` ◦ If found, return `TRUE`, otherwise return `FALSE`

(a) State **one** reason for storing data in a file rather than in an array.

.....

..... [1]

(b) Write **program code** for the module `SearchFileNtoZ()`.

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

Programming language .....

Program code

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [7]

(c) Write **pseudocode** for the module `FindDuplicates()`.

The module description is given in the table on page 12.

.....

.....

.....

.....

.....

.....

..... [8]

(d) `ClearArray()` is to be modified to make it general purpose. It will be used to initialise any 1D array of data type `STRING` to any value.

It will now be called with three parameters as follows:

1. The array
2. The number of elements
3. The initialisation string

You should assume that the lower bound is 1.

(i) Write **pseudocode** for the modified `ClearArray()` procedure.

.....

.....

.....

.....

.....

.....

.....

..... [3]

(ii) Write **program code** for a statement that calls the modified `ClearArray()` procedure to clear the array `Duplicates` to `"Empty"`.

Programming language .....

Program code

.....

.....

[2]

Appendix

Built-in functions (pseudocode)

Each function returns an error if the function call is not properly formed.

`MID(ThisString : STRING, x : INTEGER, y : INTEGER) RETURNS STRING`
returns a string of length `y` starting at position `x` from `ThisString`

Example: `MID("ABCDEFGH", 2, 3)` returns `"BCD"`

`LENGTH(ThisString : STRING) RETURNS INTEGER`
returns the integer value representing the length of `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 `"ABC"`

`RIGHT(ThisString: STRING, x : INTEGER) RETURNS STRING`
returns rightmost `x` characters from `ThisString`

Example: `RIGHT("ABCDEFGH", 3)` returns `"FGH"`

`INT(x : REAL) RETURNS INTEGER`
returns the integer part of `x`

Example: `INT(27.5415)` returns `27`

`NUM_TO_STRING(x : REAL) RETURNS STRING`
returns a string representation of a numeric value.
Note: This function will also work if `x` is of type `INTEGER`

Example: `NUM_TO_STRING(87.5)` returns `"87.5"`

`STRING_TO_NUM(x : STRING) RETURNS REAL`
returns a numeric representation of a string.
Note: This function will also work if `x` is of type `CHAR`

Example: `STRING_TO_NUM("23.45")` returns `23.45`

`ASC(ThisChar : CHAR) RETURNS INTEGER`
returns the ASCII value of `ThisChar`

Example: `ASC('A')` returns `65`

`CHR(x : INTEGER) RETURNS CHAR`
returns the character whose ASCII value is `x`

Example: `CHR(87)` returns `'W'`

`UCASE(ThisChar : CHAR) RETURNS CHAR`
returns the character value representing the upper case equivalent of `ThisChar`
If `ThisChar` is not a lower case alphabetic character, it is returned unchanged.

Example: `UCASE('a')` returns `'A'`

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`

BLANK PAGE

BLANK PAGE

QUESTION 21.

6 A text file, `Library.txt`, stores information relating to a book collection. The 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`.

.....

.....

..... [3]

(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 va
Python: You should show a comment statement for each variable used with its d

Programming language .....

Program code

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [8]

(b) The function `FindBooksBy()` has already been called and has stored values in the array `Result`.

The procedure, `DisplayResults()`, will output the information from the array.

The procedure receives the following two parameters:

- a string containing the author name
- an integer value representing the number of books found

The output should be formatted as in the following example:

```
Books written by: Brian Smith

Title                    Location
Learning Python          BD345
Arrays are not lists     CZ562
Learning Java            CZ589

Number of titles found: 3
```

If no books by the author are found, the following should be output:

```
Search found no books by: Brian Smith
```

Write **pseudocode** for the procedure `DisplayResults()`.

Refer to the **Appendix** on page 16 for the list of built-in functions and operators.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [7]