

9618 Computer Science Paper 2 - Programming

Topical questions | 96 questions

1. Question part 1/4 | 9618 May-June 2021 P2 V1 Q4

4 Study the following pseudocode. Line numbers are for reference only.

```
10 FUNCTION Convert(Name : STRING) RETURNS STRING
11
12   DECLARE Flag: BOOLEAN
13   DECLARE Index : INTEGER
14   DECLARE ThisChar : CHAR
15   DECLARE NewName : STRING
16
17   CONSTANT SPACECHAR = ' '
18
19   Flag ← TRUE
20   Index ← 1
21   NewName ← ""           // formatted name string
22
23   WHILE Index <= LENGTH(Name)
24     ThisChar ← MID(Name, Index, 1)
25     IF Flag = TRUE THEN
26       NewName ← NewName & UCASE(ThisChar)
27       IF ThisChar <> SPACECHAR THEN
28         Flag ← FALSE
29       ENDIF
30     ELSE
31       NewName ← NewName & ThisChar
32     ENDIF
33     IF ThisChar = SPACECHAR THEN
34       Flag ← TRUE
35     ENDIF
36     Index ← Index + 1
37   ENDWHILE
38
39   RETURN NewName
40
41 ENDFUNCTION
```

Topical questions | 96 questions

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

Note: The symbol '∇' has been used to represent a space character. Use this symbol for any space characters in the trace table.

The first row has been completed for you.

[illegible]

(b) The pseudocode for `Convert()` contains a conditional loop.

State a more appropriate loop structure.

Justify your answer.

Loop structure

.....

Justification

.....

.....

[2]

(c) Two changes need to be made to the algorithm.

Change 1: Convert to lower case any character that is not the first character after a space.

Change 2: Replace multiple spaces with a single space.

(i) Change 1 may be implemented by modifying one line of the pseudocode.

Write the modified line.

.....

.....[1]

(ii) Change 2 may be implemented by moving one line of the pseudocode.

Write the number of the line to be moved and state its new position.

Line number

New position

.....

[2]

BLANK PAGE

- declare a 1D array `RNum` of 100 elements of type `INTEGER`
- assign each element a random value in the range 1 to 200 inclusive
- count and output how many numbers generated were between 66 and 173 inclusive.

[6]

..... [3]

(b) The following is a pseudocode function.

Line numbers are given for reference only.

```
01  FUNCTION StringClean(InString : STRING) RETURNS STRING
02
03      DECLARE NextChar : CHAR
04      DECLARE OutString : STRING
05      DECLARE Counter : INTEGER
06
07      OutString ← ""
08
09      FOR Counter ← 1 TO LENGTH(InString)
10          NextChar ← MID(InString, Counter, 1)
11          NextChar ← LCASE(NextChar)
12          IF NOT((NextChar < 'a') OR (NextChar > 'z')) THEN
13              OutString ← OutString & NextChar
14          ENDIF
15      NEXT Counter
16
17      RETURN OutString
18
19  ENDFUNCTION
```

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

Answer	
Give a line number containing an example of an initialisation statement.	
Give a line number containing the start of a repeating block of code.	
Give a line number containing a logic operation.	
Give the number of parameters to the function MID().	

[4]

(ii) Write a simplified version of the statement in line 12.

.....
..... [2]

BLANK PAGE

9618 Computer Science Paper 2 - Programming

Topical questions | 96 questions

3. Question part 1/2 | 9618 May-June 2021 P2 V2 Q6

6 A procedure `CountVowels()` will:

- be called with a string containing alphanumeric characters as its parameter
- count and output the number of occurrences of each vowel (a, e, i, o, u) in the string
- count and output the number of occurrences of the other alphabetic characters (as a single total).

The string may contain both upper and lower case characters.

Each count value will be stored in a unique element of a global 1D array `CharCount` of type `INTEGER`. The array will contain six elements.

Write pseudocode for the procedure `CountVowels()`.

This image shows a single sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

3. Question part 2/2 | 9618 May-June 2021 P2 V2 Q6

.....

.....

.....

.....

..... [8]

- 8 A program is needed to take a string containing a full name and to produce a new string of initials.

Some words in the full name will be ignored. For example, “the”, “and”, “of”, “for” and “to” may all be ignored.

Each letter of the new string must be upper case.

For example:

Full name	Initials
Integrated Development Environment	IDE
The American Standard Code for Information Interchange	ASCII

The programmer has decided to use the following global variables:

- a ten element 1D array `IgnoreList` of type `STRING` to store the ignored words
- a string `FNString` to store the full name string.

Assume that:

- each alphabetic character in the full name string may be either upper or lower case
- the full name string contains at least one word.

The programmer has started to define program modules as follows:

Module	Description
<code>GetStart()</code>	• Called with an <code>INTEGER</code> as its parameter, representing the number of a word in <code>FNString</code> • Returns the character start position of that word in <code>FNString</code> or returns <code>-1</code> if that word does not exist • For example: <code>GetStart(3)</code> applied to "hot and cold" returns 9
<code>GetWord()</code>	• Called with the position of the first character of a word in <code>FNString</code> as its parameter • Returns the word from <code>FNString</code> • For example: if <code>FNString</code> contains the string "hot and cold", <code>GetWord(9)</code> returns "cold"
<code>IgnoreWord()</code>	• Called with a <code>STRING</code> parameter representing a word • Searches for the word in the <code>IgnoreList</code> array • Returns <code>TRUE</code> if the word is found, otherwise returns <code>FALSE</code>
<code>GetInitials()</code>	• Processes the sequence of words in the full name one word at a time • Calls <code>GetStart()</code>, <code>GetWord()</code> and <code>IgnoreWord()</code> to process each word to form the new string • Outputs the new string

[8]

Topical questions | 96 questions

5 A global 2D array `Result` of type `INTEGER` is used to store a list of exam candidate numbers together with their marks. The array contains 2000 elements, organised as 1000 rows and 2 columns.

Column 1 contains the candidate number and column 2 contains the mark for the corresponding candidate. All elements contain valid exam result data.

A procedure `Sort()` is needed to sort `Result` into ascending order of mark using an efficient bubble sort algorithm.

Write pseudocode for the procedure `Sort()`.

[illegible]

Topical questions | 96 questions

[8]

- 5 The following data items will be recorded each time a student successfully logs on to the school network:

Data item	Example data
Student ID	"CJL404"
Host ID	"Lib01"
Time and date	"08:30, June 01, 2021"

The Student ID is six characters long. The other two data items are of variable length.

A single string will be formed by concatenating the three data items. A separator character will need to be inserted between items two and three.

For example:

"CJL404Lib01<separator>08:30, June 01, 2021"

Each string represents one log entry.

A programmer decides to store the concatenated strings in a 1D array `LogArray` that contains 2000 elements. Unused array elements will contain an empty string.

- (a) Suggest a suitable separator character **and** give a reason for your choice.

Character

Reason

..... [2]

- (b) The choice of data structure was made during one stage of the program development life cycle.

Identify this stage.

..... [1]

- Write pseudocode for the function `LogEvents()`.

[7]

5 A company has several departments. Each department stores the name, email address and the status of each employee in that department in its own text file. All text files have the same format.

Employee details are stored as three separate data strings on three consecutive lines of the file. An example of the first six lines of one of the files is as follows:

File line	Comment
1	First employee name
2	First email address
3	First employee status
4	Second employee name
5	Second email address
6	Second employee status

A procedure `MakeNewFile()` will:

- take three parameters as strings:
 - an existing file name
 - a new file name
 - a search status value
- create a new text file using the new file name
- write all employee details to the new file where the employee status is **not** equal to the search status value
- count the number of sets of employee details that were in the original file
- count the number of sets of employee details that were written to the new file
- produce a summary output.

An example summary output is as follows:

```
File Marketing contained 54 employee details
52 employee sets of details were written to file NewMarketingList
```

(a) Write pseudocode for the procedure `MakeNewFile()`.

.....

.....

.....

.....

.....

.....

.....

.....

9618 Computer Science Paper 2 - Programming

Topical questions | 96 questions

7. Question part 2/4 | 9618 Oct-Nov 2021 P2 V2 Q5

..... [7]

[7]

(b) An alternative format could be used for storing the data.

A text file will still be used.

(i) Describe the alternative format.

.....
..... [1]

(ii) State **one** advantage **and one** disadvantage of the alternative format.

Advantage
.....
Disadvantage
..... [2]

BLANK PAGE

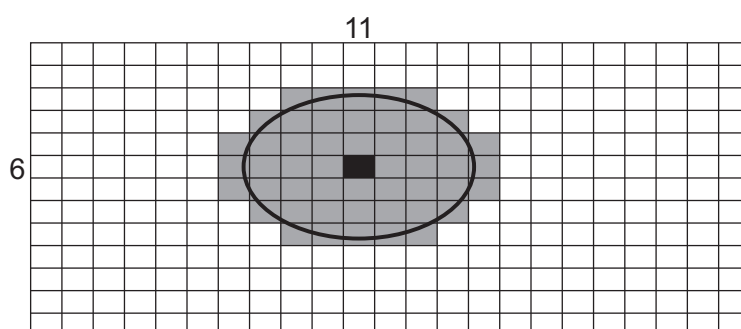
- 6** A mobile phone has a touchscreen. The screen is represented by a grid, divided into 800 rows and 1280 columns.

The grid is represented by a 2D array `Screen` of type `INTEGER`. An array element will be set to 0 unless the user touches that part of the screen.

Many array elements will be set to 1 by a single touch of a finger or a stylus.

The following diagram shows a simplified touchscreen. The dark line represents a touch on the screen. All grid elements that are wholly or partly inside the outline will be set to 1. These elements are shaded.

The element shaded in black represents the centre point of the touch.



A program is needed to find the coordinates (the row and column) of the centre point. The centre point on the diagram shown is row 6, column 11.

Assume:

- the user may only touch one area at a time
- screen rotation does not affect the touchscreen.

The programmer has decided to use global values `CentreRow` and `CentreCol` as coordinate values for the centre point.

The programmer has started to define program modules as follows:

Module	Description
<code>FirstRowSet()</code>	- Searches for the first row that has an array element set to 1 - Returns the index of that row (1 is the first row) - Returns -1 if there are no elements set to 1
<code>LastRowSet()</code>	- Searches for the last row that has an array element set to 1 - Returns the index of that row - Returns -1 if there are no elements set to 1
<code>FirstColSet()</code>	- Searches for the first column that has an array element set to 1 - Returns the index of that column (1 is the first column) - Returns -1 if there are no elements set to 1
<code>LastColSet()</code>	- Searches for the last column that has an array element set to 1 - Returns the index of that column - Returns -1 if there are no elements set to 1

[7]

(b) Describe a feature of your solution to **part (a)** that indicates the pseudocode represents an efficient algorithm.

.....

.....

.....

..... [2]

(c) The programmer decides to produce a **single** search module `FindSet()`, which will be able to perform each of the individual searches performed by the first four modules in the table.

(i) Outline the changes needed to convert one of the existing modules into this single module.

.....

.....

.....

.....

.....

..... [2]

(ii) Give one possible advantage **and** one possible disadvantage of combining the four searches into a single module.

Advantage

.....

Disadvantage

..... [2]

Write pseudocode for the module `GetCentre()`.

[6]

- 5 The following data items will be recorded each time a student successfully logs on to the school network:

Data item	Example data
Student ID	"CJL404"
Host ID	"Lib01"
Time and date	"08:30, June 01, 2021"

The Student ID is six characters long. The other two data items are of variable length.

A single string will be formed by concatenating the three data items. A separator character will need to be inserted between items two and three.

For example:

"CJL404Lib01<separator>08:30, June 01, 2021"

Each string represents one log entry.

A programmer decides to store the concatenated strings in a 1D array `LogArray` that contains 2000 elements. Unused array elements will contain an empty string.

- (a) Suggest a suitable separator character **and** give a reason for your choice.

Character

Reason

.....

[2]

- (b) The choice of data structure was made during one stage of the program development life cycle.

Identify this stage.

..... [1]

- Write pseudocode for the function `LogEvents()`.

..... [7]

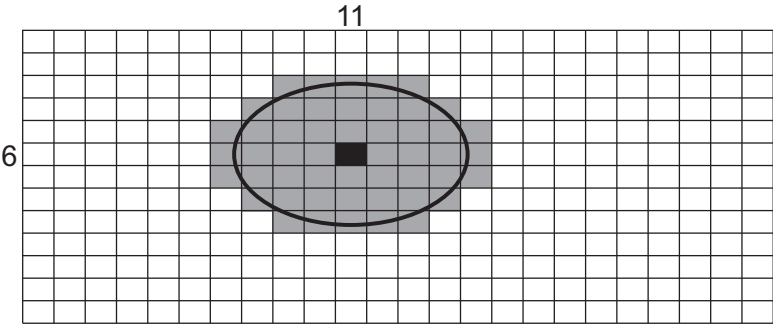
10. Question part 1/4 | 9618 Oct-Nov 2021 P2 V3 Q6

6 A mobile phone has a touchscreen. The screen is represented by a grid, divided into 800 rows and 1280 columns.

The grid is represented by a 2D array `Screen` of type `INTEGER`. An array element will be set to 0 unless the user touches that part of the screen.

Many array elements are set to 1 by a single touch of a finger or a stylus.

The following diagram shows a simplified touchscreen. The dark line represents a touch to the screen. All grid elements that are wholly or partly inside the outline will be set to 1. These elements are shaded. The element shaded in black represents the centre point.



A program is needed to find the coordinates (the row and column) of the centre point. The centre point on the diagram above is row 6, column 11.

Assume:

- the user may only touch one area at a time
- screen rotation does not affect the touchscreen.

The programmer has started to define program modules as follows:

Module	Description
<code>SetRow()</code> (generates test data)	• Called with three parameters of type <code>INTEGER</code>: ◦ a row number ◦ the number of pixels to be skipped starting from column 1 ◦ the number of pixels that should be set to 1 • Sets the required number of pixels to 1 For example, <code>SetRow(3, 8, 5)</code> will give row 3 as in the diagram shown.
<code>SearchInRow()</code>	• Takes two parameters of type <code>INTEGER</code>: ◦ a row number ◦ a start column (1 or 1280) • Searches the given row from the start column (either left to right or right to left) for the first column that contains an element set to 1 • Returns the column number of the first element in the given row that is set to 1 • Returns -1 if no element is set to 1
<code>SearchInCol()</code>	• Takes two parameters of type <code>INTEGER</code>: ◦ a column number ◦ a start row (1 or 800) • Searches the given column from the start row (either up or down) for the first row that contains an element set to 1 • Returns the row number of the first element in the given column that is set to 1 • Returns -1 if no element is set to 1

Topical questions | 96 questions

(b) The module description of `SearchInRow()` is provided here for reference.

Module	Description
SearchInRow()	• Takes two parameters of type <code>INTEGER</code>: ◦ a row number ◦ a start column (1 or 1280) • Searches the given row from the start column (either left to right or right to left) for the first column that contains an element set to 1 • Returns the column number of the first element in the given row that is set to 1 • Returns -1 if no element is set to 1

Write pseudocode to implement the module `SearchInRow()`.

This image shows a full page of white paper with horizontal dotted lines. The lines are evenly spaced and run across the width of the page, providing a guide for handwriting practice. There are no margins, text, or other markings on the page.

Module	Description
GetCentreCol ()	• Called with a row number as an <code>INTEGER</code> • Uses <code>SearchInRow()</code> to find the first and last columns in the given row which have an array element set to 1 • Returns the index of the column midway between the first and last columns • Returns <code>-1</code> if no element is set to 1

.....[6]

11. Question part 1/2 | 9618 Specimen 2021 P2 Q4

4 (a) Parameter `x` is used to pass data to procedure `MyProc` in the following pseudocode:

```
x ← 4
CALL MyProc(x)
OUTPUT x
```



```
PROCEDURE MyProc(x : INTEGER)
  DECLARE z : INTEGER
  x ← x + 1
  z ← x + 3
ENDPROCEDURE
```

There are two parameter passing methods that could be used.

Complete the following table for each of the two methods.

Name of parameter passing method	Value output	Explanation
.....		
.....		

[6]

(b) The pseudocode includes the use of parameters.

State **two** other features in the pseudocode that support a modular approach to programming.

- 1
-
- 2
-

[2]

BLANK PAGE

- 1 At least two lower-case alphabetic characters
- 2 At least two upper-case alphabetic characters
- 3 At least three numeric characters
- 4 Alpha-numeric characters only

- TRUE if it is a valid password
- FALSE otherwise

Refer to the **Insert** for the list of pseudocode functions and operators.

This image shows a full page of white paper with horizontal dotted lines. The lines are evenly spaced and run across the width of the page, providing a guide for handwriting practice. There are no margins, text, or other markings on the page.

Topical questions | 96 questions

[illegible]

(b) The `ValidatePassword` function will be tested.

- (i) Give a valid password that can be used to check that the function returns `TRUE` under the correct conditions.

Password1: [1]

- (ii) Password1 is modified to test each rule separately. Give **four** modified passwords and justify your choice.

Password to test rule 1:

Reason:

.....

.....

Password to test rule 2:

Reason:

.....

.....

Password to test rule 3:

Reason:

.....

.....

Password to test rule 4:

Reason:

.....

.....

[4]

- (iii) When testing the `ValidatePassword` function a module it is necessary to test all possible paths through the code.

State the name given to this type of validation testing.

..... [1]

(iv) A program consisting of several functions can be tested using a process known as ‘stub testing’

Explain this process.

.....

.....

.....

..... [2]

9618 Computer Science Paper 2 - Programming

Topical questions | 96 questions

13. Question | 9618 Specimen 2021 P2 Q7

7 LogArray is a 1D array containing 500 elements of type STRING.

A procedure, `LogEvents`, is required to add data from the array to the end of the existing text file `LoginFile.txt`

Unused array elements are assigned the value "Empty". These can occur anywhere in the array and should **not** be added to the file.

Write pseudocode for the procedure `LogEvents`.

Refer to the **Insert** for the list of pseudocode functions and operators.

[8]

9618 Computer Science Paper 2 - Programming

Topical questions | 96 questions

14. Question part 1/2 | 9618 May-June 2022 P2 V1 Q6

6 (a) An algorithm will:

- output **each** integer value between 100 and 200 that ends with the digit 7, for example, 107
- output a final count of the number of values that are output.

Write pseudocode for this algorithm.

Any variables used must be declared.

[5]

[5]

Write pseudocode with the same functionality **without** using a CASE structure.

..... [4]

Topical questions | 96 questions

7 A string is a palindrome if it reads the same forwards as backwards.

"Racecar"
"madam"
"12344321"

(a) A function `IsPalindrome()` will take a string parameter. The function will return `TRUE` if the string is a palindrome and will return `FALSE` if the string is not a palindrome.

Write pseudocode for `IsPalindrome()`.

This image shows a full page of white paper with horizontal dotted lines. The lines are evenly spaced and run across the width of the page, providing a guide for handwriting practice. There are no margins, text, or other markings on the page.

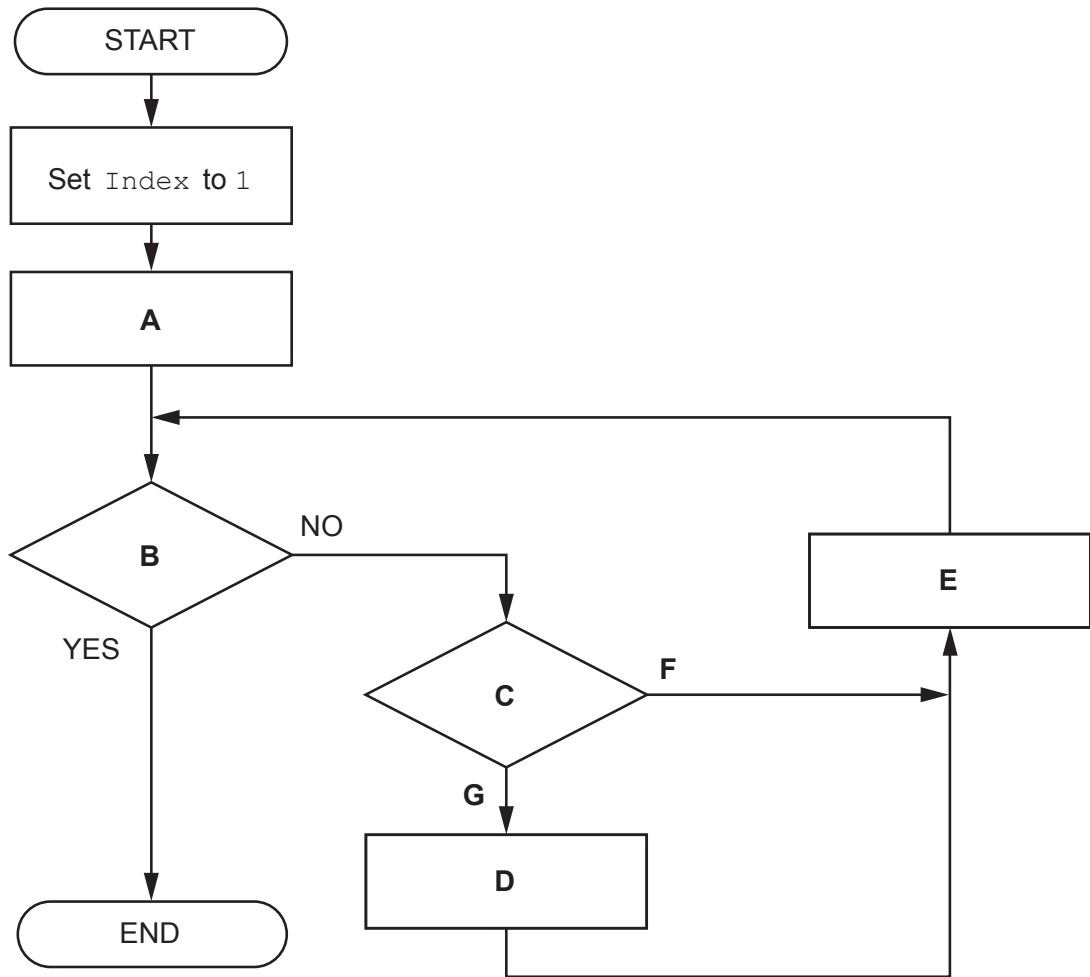
Topical questions | 96 questions

[7]

(b) Strings may consist of several words separated by spaces.

For example, the string "never odd or even" becomes a palindrome if the spaces are removed.

The program flowchart represents an algorithm to produce a string `OutString` by removing all spaces from a string `InString`.



Complete the table by writing the text that should replace each of the labels **B**, **C**, **D**, **F** and **G**.

Note: the text may be written as a pseudocode statement.

Label	Text
A	Set <code>OutString</code> to ""
B	
C	
D	
E	Set <code>Index</code> to <code>Index</code> + 1
F	
G	

- 8** A program allows a user to save passwords used to login to websites. A stored password is inserted automatically when the user logs into the corresponding website.

A student is developing a program to generate a password. The password will be of a fixed format, consisting of **three groups of four** alphanumeric characters. The groups are separated by the hyphen character '-'.
 An example of a password is: "FxAf-3haV-Tq49"

A global 2D array `Secret` of type `STRING` stores the passwords together with the website domain name where they are used. `Secret` contains 1000 elements organised as 500 rows by 2 columns.

Unused elements contain the empty string (""). These may occur anywhere in the array.

An example of a part of the array is:

Array element	Value
<code>Secret[27, 1]</code>	"www.thiswebsite.com"
<code>Secret[27, 2]</code>	"....."
<code>Secret[28, 1]</code>	"www.thatwebsite.com"
<code>Secret[28, 2]</code>	"....."

Note:

- For security, passwords are stored in an encrypted form, shown as "....." in the example.
- The passwords cannot be used without being decrypted.
- Assume that the encrypted form of a password will **not** be an empty string.

The programmer has started to define program modules as follows:

Module	Description
<code>RandomChar()</code>	- Generates a single random character from within one of the following ranges: 'a' to 'z' 'A' to 'Z' '0' to '9' - Returns the character
<code>Encrypt()</code>	- Takes a password as a parameter of type string - Returns the encrypted form of the password as a string
<code>Decrypt()</code>	- Takes an encrypted password as a parameter of type string - Returns the decrypted form of the password as a string

For reference, relevant ASCII values are as follows:

Character range	ASCII range
'a' to 'z'	97 to 122
'A' to 'Z'	65 to 90
'0' to '9'	48 to 57

..... 16

[6]

[7]

(c) The modules `Encrypt()` and `Decrypt()` are called from several places in the main program.

Identify a method that could have been used to test the main program before these modules were completed. Describe how this would work.

Method

Description

.....

.....

..... [3]

(d) A validation function is written to check that the passwords generated are valid.

To be valid, each password must:

- be 14 characters long
- be organised as three groups of four case-sensitive alphanumeric characters. The groups are separated by hyphen characters
- not include any duplicated characters, except for the hyphen characters.

Note: lower-case and upper-case characters are not the same. For example, 'a' is not the same as 'A'.

Give **two** password strings that could be used to test different areas of the validation rules.

Password 1

Password 2 [2]

(e) The `RandomChar()` module is to be modified so that alphabetic characters are generated twice as often as numeric characters.

Describe how this might be achieved.

.....

.....

.....

.....

.....

..... [3]

9618 Computer Science Paper 2 - Programming

Topical questions | 96 questions

17. Question part 1/2 | 9618 May-June 2022 P2 V2 Q4

4 (a) A procedure `LastLines()` will:

- take the name of a text file as a parameter
- output the **last** three lines from that file, in the same order as they appear in the file.

Note:

- Use local variables `LineX`, `LineY` and `LineZ` to store the three lines from the file.
- You may assume the file exists and contains **at least** three lines.

Write pseudocode for the procedure `LastLines()`.

[6]

(b) The algorithm in **part (a)** is to be amended. The calling program will pass the number of lines to be output as well as the name of the text file.

The number of lines could be any value from 1 to 30.

It can be assumed that the file contains **at least** the number of lines passed.

Outline **three** changes that would be needed.

- 1
-
-
- 2
-
-
- 3
-
-

[3]

5 Study the following pseudocode. Line numbers are for reference only.

```

10 PROCEDURE Encode()
11   DECLARE CountA, CountB, ThisNum : INTEGER
12   DECLARE ThisChar : CHAR
13   DECLARE Flag : BOOLEAN
14   CountA ← 0
15   CountB ← 10
16   Flag ← TRUE
17   INPUT ThisNum
18   WHILE ThisNum <> 0
19       ThisChar ← LEFT(NUM_TO_STR(ThisNum), 1)
20       IF Flag = TRUE THEN
21           CASE OF ThisChar
22               '1' : CountA ← CountA + 1
23               '2' : IF CountB < 10 THEN
24                   CountA ← CountA + 1
25               ENDIF
26               '3' : CountB ← CountB - 1
27               '4' : CountB ← CountB - 1
28                   Flag ← FALSE
29               OTHERWISE : OUTPUT "Ignored"
30           ENDCASE
31       ELSE
32           IF CountA > 2 THEN
33               Flag ← NOT Flag
34               OUTPUT "Flip"
35           ELSE
36               CountA ← 4
37           ENDIF
38       ENDIF
39       INPUT ThisNum
40   ENDWHILE
41   OUTPUT CountA
42 ENDPROCEDURE

```

(a) Procedure `Encode()` contains a loop structure.

Identify the type of loop **and** state the condition that ends the loop.

Do **not** include pseudocode statements in your answer.

Type

Condition

.....

18. Question part 2/2 | 9618 May-June 2022 P2 V2 Q5

(b) Complete the trace table below by dry running the procedure `Encode()` when the following values are input:

12, 24, 57, 43, 56, 22, 31, 32, 47, 99, 0

The first row is already complete.

ThisNum	ThisChar	CountA	CountB	Flag	OUTPUT
		0	10	TRUE	

[6]

(c) Procedure `Encode()` is part of a modular program. Integration testing is to be carried out on the program.

Describe **integration testing**.

.....

.....

.....

..... [2]

9618 Computer Science Paper 2 - Programming

Topical questions | 96 questions

19. Question part 1/2 | 9618 May-June 2022 P2 V2 Q6

6 A string represents a series of whole numbers, separated by commas.

For example:

"12,13,451,22"

Assume that:

- the comma character ',' is used as a separator
- the string contains only the characters '0' to '9' and the comma character ','.

A procedure `Parse` will:

- take the string as a parameter
- extract each number in turn
- calculate the total value and average value of all the numbers
- output the total and average values with a suitable message.

Write pseudocode for the procedure.

```
PROCEDURE Parse(InString : STRING)
```

[illegible]

19. Question part 2/2 | 9618 May-June 2022 P2 V2 Q6

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

ENDPROCEDURE [7]

The language includes a `LEFT ()` and a `RIGHT ()` function, but it does **not** have a `MID ()` function.

(a) Write pseudocode for an algorithm to implement your own version of the `MID()` function which will operate in the same way as that shown in the **insert**.

Do **not** use the `MID()` function given in the **insert**, but you may use any of the other functions.

Assume that the values passed to the function will be correct.

..... [4]

(b) The values passed to your `MID()` function in **part (a)** need to be validated.

Assume that the values are of the correct data type.

State **two** checks that could be applied to the values passed to the function.

1

2

[2]

BLANK PAGE

- 8 A program allows a user to save passwords used to log in to websites. A stored password is then inserted automatically when the user logs in to the corresponding website.

A global 2D array `Secret` of type `STRING` stores the passwords together with the website domain name where they are used. `Secret` contains 1000 elements organised as 500 rows by 2 columns.

Unused elements contain the empty string (`""`). These may occur anywhere in the array.

An example of a part of the array is:

Array element	Value
<code>Secret[27, 1]</code>	<code>"thiswebsite.com"</code>
<code>Secret[27, 2]</code>	<code>"....."</code>
<code>Secret[28, 1]</code>	<code>"thatwebsite.com"</code>
<code>Secret[28, 2]</code>	<code>"....."</code>

Note:

- For security, the passwords are stored in an encrypted form, shown as `"....."` in the example.
- The passwords cannot be used without being decrypted.
- You may assume that the encrypted form of a password will **NOT** be an empty string.

The programmer has started to define program modules as follows:

Module	Description
<code>Exists()</code>	- Takes two parameters: - a string - a character - Performs a case-sensitive search for the character in the string - Returns <code>TRUE</code> if the character occurs in the string, otherwise returns <code>FALSE</code>
<code>Encrypt()</code>	- Takes a password as a parameter of type string - Returns the encrypted form of the password as a string
<code>Decrypt()</code>	- Takes an encrypted password as a parameter of type string - Returns the decrypted form of the password as a string

Note: in a case-sensitive comparison, 'a' is not the same as 'A'.

Topical questions | 96 questions

(b) A new module `SearchDuplicates()` will:

- For example, if the same password was used for the three websites ThisWebsite.com, website27.net and websiteZ99.org, then the following messages will be output:

```
"Password for ThisWebsite.com also used for website27.net"
"Password for ThisWebsite.com also used for websiteZ99.org"
```

- The module will output a message if no duplicates are found.
For example:

```
"No duplicate passwords found"
```

Write efficient pseudocode for the module `SearchDuplicates()`. `Encrypt()` and `Decrypt()` functions have been written.

Note: It is necessary to decrypt each password before checking its value.

[illegible]

- (c) A password has a fixed format, consisting of **three groups of four** alphanumeric characters, separated by the hyphen character '-'.

An example of a password is:

"FxAf-3haV-Tq49"

Each password must:

- be 14 characters long
- be organised as three groups of four alphanumeric characters. The groups are separated by hyphen characters
- not include any duplicated characters, except for the hyphen characters.

An algorithm is needed for a new function `GeneratePassword()`, which will generate and return a password in this format.

Assume that the following modules have already been written:

Module	Description
<code>Exists()</code>	• Takes two parameters: ◦ a string ◦ a character • Performs a case-sensitive search for the character in the string • Returns <code>TRUE</code> if the character occurs in the string, otherwise returns <code>FALSE</code>
<code>RandomChar()</code>	• Generates a single random character from within one of the following ranges: ◦ 'a' to 'z' ◦ 'A' to 'Z' ◦ '0' to '9' • Returns the character

Note: in a case-sensitive comparison, 'a' is not the same as 'A'.

Do **not** use pseudocode statements in your answer.

[6]

- 1 (a) The following table contains pseudocode examples.

Each example may include all or part of:

- selection
- iteration (repetition)
- assignment.

Complete the table by placing **one or more** ticks (✓) in each row.

Pseudocode example	Selection	Iteration	Assignment
FOR Index ← 1 TO 3 Safe[Index] ← GetResult() NEXT Index			
OTHERWISE : OUTPUT "ERROR 1202"			
REPEAT UNTIL Index = 27			
INPUT MyName			
IF Mark > 74 THEN Grade ← 'A' ENDIF			

[5]

- (b) (i) Program variables have values as follows:

Variable	Value
AAA	TRUE
BBB	FALSE
Count	99

Complete the table by evaluating each expression.

Expression	Evaluation
AAA AND (Count > 99)	
AAA AND (NOT BBB)	
(Count <= 99) AND (AAA OR BBB)	
(BBB AND Count > 50) OR NOT AAA	

[2]

- (ii) Give an example of when a variable of type Boolean would be used.

.....

..... [1]

- 9 A program allows a user to save passwords used to log in to websites. A stored password is then inserted automatically when the user logs in to the corresponding website.

A student is developing a program to generate a strong password. The password will be of a fixed format, consisting of **three groups of four** alphanumeric characters, separated by the hyphen character '-'.

An example of a password is: "FxAf-3hzV-Aq49"

A valid password:

- must be 14 characters long
- must be organised as three groups of four alphanumeric characters. The groups are separated by hyphen characters
- may include duplicated characters, **provided** these appear in different groups.

The programmer has started to define program modules as follows:

Module	Description
<code>RandomChar()</code>	• Generates a single random character from within one of the following ranges: ◦ 'a' to 'z' ◦ 'A' to 'Z' ◦ '0' to '9' • Returns the character
<code>Exists()</code>	• Takes two parameters: ◦ a string ◦ a character • Performs a case-sensitive search for the character in the string • Returns <code>TRUE</code> if the character occurs in the string, otherwise returns <code>FALSE</code>
<code>Generate()</code>	• Generates a valid password • Uses <code>RandomChar()</code> and <code>Exists()</code> • Returns the password

Note: in a case-sensitive comparison, 'a' is not the same as 'A'.

- (b) A global 2D array `Secret` of type `STRING` stores the passwords together with the website domain name where they are used. `Secret` contains 1000 elements organised as 500 rows by 2 columns.

Unused elements contain the empty string (`""`). These may occur anywhere in the array.

An example of part of the array is:

Array element	Value
<code>Secret[27, 1]</code>	<code>"www.thiswebsite.com"</code>
<code>Secret[27, 2]</code>	<code>"●●●●●●●●●●"</code>
<code>Secret[28, 1]</code>	<code>"www.thatwebsite.com"</code>
<code>Secret[28, 2]</code>	<code>"●●●●●●●●●●"</code>

Note:

- For security, the passwords are stored in an encrypted form, shown as `"●●●●●●●●●●"` in the example.
- The passwords cannot be used without being decrypted.
- You may assume that the encrypted form of a password will **not** be an empty string.

Additional modules are defined as follows:

Module	Description
<code>Encrypt()</code>	- Takes a password as a string - Returns the encrypted form of the password as a string
<code>Decrypt()</code>	- Takes an encrypted password as a string - Returns the decrypted form of the password as a string
<code>FindPassword()</code>	- Takes a website domain name as a string - Searches for the website domain name in the array <code>Secret</code> - If the website domain name is found, the decrypted password is returned - If the website domain name is not found, an empty string is returned
<code>AddPassword()</code>	- Takes two parameters as strings: a website domain name and a password - Searches for the website domain name in the array <code>Secret</code> and if not found, adds the website domain name and the encrypted password to the array - Returns <code>TRUE</code> if the website domain name and encrypted password are added to the array, otherwise returns <code>FALSE</code>

The first three modules have been written.

9618 Computer Science Paper 2 - Programming

Topical questions | 96 questions

23. Question part 4/5 | 9618 May-June 2022 P2 V3 Q9

Write pseudocode for the module `AddPassword()`.

[6]

[6]

(c) The content of the array `Secret` is to be stored in a text file for backup.

It **must** be possible to read the data back from the file and extract the website domain name and the encrypted password.

Both the website domain name and encrypted password are stored in the array as strings of characters.

The encrypted password may contain any character from the character set used and the length of both the encrypted password and the website domain name is variable.

Explain how a single line of the text file can be used to store the website domain name and the encrypted password.

.....

.....

.....

.....

.....

..... [3]

- 1 (a) An algorithm includes a number of complex calculations. A programmer is writing a program to implement the algorithm and decides to use library routines to provide part of the solution.

State **three** possible benefits of using library routines in the development of the program.

1

.....

2

.....

3

.....

[3]

- (b) The following pseudocode is part of a program that stores names and test marks for use in other parts of the program.

```
DECLARE Name1, Name2, Name3 : STRING
DECLARE Mark1, Mark2, Mark3 : INTEGER
INPUT Name1
INPUT Mark1
INPUT Name2
INPUT Mark2
INPUT Name3
INPUT Mark3
```

- (i) The pseudocode needs to be changed to allow for data to be stored for up to 30 students.

Explain why it would be good practice to use arrays to store the data.

.....

.....

.....

.....

.....

..... [3]

(ii) The following pseudocode statement includes array references:

```
OUTPUT "Student ", Name[Count], " scored ", Mark[Count]
```

State the purpose of the variable `Count` and give its data type.

Purpose

.....

Data type

[2]

(c) The pseudocode statements in the following table may contain errors.

State the error in each case or write 'NO ERROR' if the statement contains no error.

Assume that any variables used are of the correct type for the given function.

Statement	Error
IF EMPTY ← "" THEN	
Status ← IS_NUM(-23.4)	
X ← STR_TO_NUM("37") + 5	
Y ← STR_TO_NUM("37" + "5")	

[4]

A procedure `Summarise()` will calculate the average of three consecutive values from `Array1` and write the result to `Array2`. This will be repeated for all values in `Array1`.

Array1		Array2	Comment
10	}	15	average of first three values
20		41	average of next three values
15			
40	}		
41			
42			

[5]

BLANK PAGE

- 6** The following pseudocode algorithm attempts to check whether a string is a valid email address.

```

FUNCTION IsValid(InString : STRING) RETURNS BOOLEAN
  DECLARE Index, Dots, Ats, Others : INTEGER
  DECLARE NextChar : CHAR
  DECLARE Valid : BOOLEAN

  Index ← 1
  Dots ← 0
  Ats ← 0
  Others ← 0
  Valid ← TRUE

  REPEAT
    NextChar ← MID(InString, Index, 1)
    CASE OF NextChar
      '.' : Dots ← Dots + 1
      '@' : Ats ← Ats + 1
            IF Ats > 1 THEN
              Valid ← FALSE
            ENDIF
      OTHERWISE : Others ← Others + 1
    ENDCASE

    IF Dots > 1 AND Ats = 0 THEN
      Valid ← FALSE
    ELSE
      Index ← Index + 1
    ENDIF

  UNTIL Index > LENGTH(InString) OR Valid = FALSE

  IF NOT (Dots >= 1 AND Ats = 1 AND Others > 8) THEN
    Valid ← FALSE
  ENDIF

  RETURN Valid

ENDFUNCTION

```

- (a)** Part of the validation is implemented by the line:

```
IF NOT (Dots >= 1 AND Ats = 1 AND Others > 8) THEN
```

State the values that would result in the condition evaluating to TRUE.

.....

.....

..... [1]

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

```
Result ← IsValid("Liz.123@big@net")
```

Index	NextChar	Dots	Ats	Others	Valid

[5]

(ii) State the value returned when IsValid() is called using the expression shown in part (b)(i).

..... [1]

An operator character is one of the following: '+' '-' '*' '/'

"803+1904"
"34/7"

- take an arithmetic expression string as a parameter
- evaluate the expression
- output the result.

- the string contains only numeric digits and a single operator character
- Value1 and Value2 represent integer values
- Value1 and Value2 are unsigned (they will not be preceded by '+' or '-').

[illegible]

Write the header for the function in pseudocode.

..... [1]

Give an example of a correctly formatted string and explain why evaluation would be impossible.

Example string

Explanation

.....

.....

[2]

Topical questions | 96 questions

28. Question part 1/4 | 9618 Oct-Nov 2022 P2 V1 Q8

- 8 A teacher is designing a program to perform simple syntax checks on programs written by students. Student programs are submitted as text files, which are known as project files.

A project file may contain blank lines.

The teacher has defined the first program module as follows:

Module	Description
<code>CheckFile()</code>	• takes the name of an existing project file as a parameter of type string • returns <code>TRUE</code> if the file is valid (it contains at least 10 non-blank lines), otherwise returns <code>FALSE</code>

- (a)** Write pseudocode for module `CheckFile()`.

[illegible]

.....

.....

.....

..... [7]

[illegible]

Two examples of syntax error that **cannot** be detected from examining a **single** line are those involving selection and iteration.

1

.....

.....

2

.....

.....

[2]

5 Examine the following pseudocode.

```
IF A = TRUE THEN
  IF B = TRUE THEN
    IF C = TRUE THEN
      CALL Sub1 ()
    ELSE
      CALL Sub2 ()
    ENDIF
  ENDIF
ELSE
  IF B = TRUE THEN
    IF C = TRUE THEN
      CALL Sub4 ()
    ELSE
      CALL Sub3 ()
    ENDIF
  ELSE
    IF C = FALSE THEN
      CALL Sub3 ()
    ELSE
      CALL Sub4 ()
    ENDIF
  ENDIF
ENDIF
```

A programmer wants to re-write the pseudocode as **four** separate IF...THEN...ENDIF statements, each containing a single CALL statement. This involves writing a single, simplified logic expression as the condition in each statement.

Write the amended pseudocode.

1

.....

.....

2

.....

.....

3

.....

.....

4

.....

.....

BLANK PAGE

7 A teacher is designing a program to perform simple syntax checks on programs written by students.

Two global 1D arrays are used to store the syntax error data. Both arrays contain 500 elements.

- Array `ErrCode` contains integer values that represent an error number in the range 1 to 800.
- Array `ErrMsgText` contains string values that represent an error description.

The following diagram shows an example of the arrays.

Index	ErrCode	ErrMsgText
1	10	"Invalid identifier name"
2	20	"Bracket mismatch"
3	50	"Undeclared variable"
4	60	"Type mismatch in assignment"
...		
500	999	<Undefined>

Note:

- There may be less than 500 error numbers so corresponding elements in both arrays may be unused. Unused elements in `ErrCode` have the value 999. The value of unused elements in `ErrMsgText` is undefined.
- Values in the `ErrCode` array are stored in ascending order but not all values may be present, for example, there may be no error code 31.

The teacher has defined two program modules as follows:

Module	Description
<code>OutputError()</code>	• takes two parameters as integers: ◦ a line number in the student's program ◦ an error number • searches for the error number in the <code>ErrCode</code> array: ◦ if found, outputs the corresponding error description and the line number, for example: "Bracket mismatch on line 34" ◦ if not found, outputs the line number and a warning, for example: "Unknown error on line 34"
<code>SortArrays()</code>	sorts the arrays into ascending order of <code>ErrCode</code>

9618 Computer Science Paper 2 - Programming

Topical questions | 96 questions

30. Question part 2/4 | 9618 Oct-Nov 2022 P2 V2 Q7

(a) Write **efficient** pseudocode for module `OutputError()`.

[6]

[6]

[8]

- (c) Two 1D arrays were described at the beginning of the question. Both arrays contain 500 elements.
- Array `ErrCode` contains integer values that represent an error number in the range 1 to 800.
 - Array `ErrText` contains string values that represent an error description.

The two arrays will be replaced by a single array. A user-defined data type (record structure) has been declared as follows:

```
TYPE ErrorRec
    DECLARE ErrCode : STRING
    DECLARE ErrText : STRING
ENDTYPE
```

- (i) State the error in the record declaration.

.....

..... [1]

- (ii) State **two** benefits of using the single array of the user-defined data type.

1

.....

2

..... [2]

- (iii) Write the declaration for the single array in pseudocode.

..... [1]

31. Question part 1/6 | 9618 Oct-Nov 2022 P2 V3 Q7

7 A teacher is designing a program to perform simple syntax checks on programs written by students.

Two global 1D arrays are used to store the syntax error data. Both arrays contain 500 elements.

- Array `ErrCode` contains integer values that represent an error number in the range 1 to 800.
- Array `ErrText` contains string values that represent an error description.

The following diagram shows an example of the arrays.

Index	ErrCode	ErrText
1	10	"Invalid identifier name"
2	20	"Bracket mismatch"
3	50	" "
4	60	"Type mismatch in assignment"
...		
500	999	<Undefined>

Note:

- There are less than 500 error codes so corresponding elements in both arrays may be unused. Unused elements in `ErrCode` have the value 999. These will occur at the end of the array. The value of unused elements in `ErrText` is undefined.
- Values in the `ErrCode` array are stored in ascending order but not all values may be present. For example, there may be no error code 31.
- Some error numbers are undefined. In these instances, the `ErrCode` array will contain a valid error number but the corresponding `ErrText` element will contain an empty string.

The teacher has defined one program module as follows:

Module	Description
<code>OutputRange()</code>	• Prompts for input of two error numbers • Outputs a list of error numbers between the two numbers input (inclusive) together with the corresponding error description • Outputs a warning message when the error description is missing as for error number 50 in the example • Outputs a suitable header and a final count of error numbers found Output based on the example array data above: <pre>List of error numbers from 1 to 60 10 : Invalid identifier name 20 : Bracket mismatch 50 : Error Text Missing 60 : Type mismatch in assignment 4 error numbers output</pre>

9618 Computer Science Paper 2 - Programming

Topical questions | 96 questions

31. Question part 3/6 | 9618 Oct-Nov 2022 P2 V3 Q7

[8]

Question 7 continues on the next page.

Write pseudocode for the module `AddError()`. Assume that the error code is **not** already in the `ErrCode` array.

This image shows a full page of white paper with horizontal dotted lines, resembling notebook paper. The lines are evenly spaced and run across the width of the page. There is no handwriting or other markings on the paper.

.....

.....

.....

.....

.....

.....

.....

..... [6]

(ii) A new Module `RemoveError()` will remove a given error number from the array.

Describe the algorithm that would be required. Do **not** include pseudocode statements in your answer.

.....

.....

.....

.....

.....

.....

..... [3]

- Write pseudocode for function `Replace()`.

[6]

33. Question part 1/2 | 9618 May-June 2023 P2 V1 Q6

6 A video-conferencing program supports up to six users. Speech from each user is sampled and digitised (converted from analogue to digital). Digitised values are stored in array `Sample`.

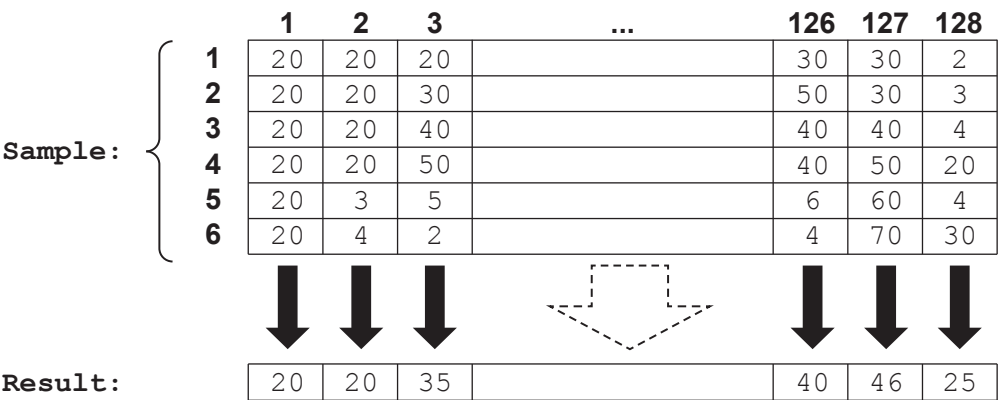
The array `Sample` consists of 6 rows by 128 columns and is of type integer. Each row contains 128 digitised sound samples from one user.

The digitised sound samples from each user are to be processed to produce a single value which will be stored in a 1D array `Result` of type integer. This process will be implemented by procedure `Mix()`.

A procedure `Mix()` will:

- calculate the average of each of the 6 sound samples in a column
- ignore sound sample values of 10 or less
- store the average value in the corresponding position in `Result`
- repeat for each column in array `Sample`

The diagram uses example values to illustrate the process:



34. Question part 1/4 | 9618 May-June 2023 P2 V1 Q8

8 A computer shop assembles computers using items bought from several suppliers. A text file `Stock.txt` contains information about each item.

Information for each item is stored as a single line in the `Stock.txt` file in the format:

```
<ItemNum><SupplierCode><Description>
```

Item information is as follows:

	Format	Comment
ItemNum	4 numeric characters	unique for each item in the range "0001" to "5999" inclusive
SupplierCode	5 alphabetic characters	to identify the supplier of the item
Description	a string	a minimum of 12 characters

The file is organised in ascending order of `ItemNum` and does **not** contain all possible values in the range.

A programmer has started to define program modules as follows:

Module	Description
<code>SuppExists()</code> (already written)	- called with a parameter of type string representing a supplier code - returns <code>TRUE</code> if the supplier code is already in use, otherwise returns <code>FALSE</code>
<code>IsNewSupp()</code>	- called with a parameter of type string representing a new supplier code - returns <code>TRUE</code> if the string only contains alphabetic characters (either upper or lower case) and the supplier code is not already in use, otherwise returns <code>FALSE</code>

Module `SuppExists()` has already been written and should be used as part of your solution. Module `SuppExists()` will generate a run-time error if the given parameter is not 5 characters in length.

..... [7]

Topical questions | 96 questions

34. Question part 3/4 | 9618 May-June 2023 P2 V1 Q8

(b) A new module has been defined:

Module	Description
CheckNewItem()	- called with a parameter of type string representing a line of item information - checks to see whether an item with the same <code>ItemNum</code> already exists in the file - returns <code>TRUE</code> if the <code>ItemNum</code> is not already in the file, otherwise returns <code>FALSE</code>

Write **efficient** pseudocode for module `CheckNewItem()`.

This image shows a full page of a handwriting practice worksheet. It consists of multiple sets of three horizontal dashed lines spaced evenly down the page, providing a guide for letter height and placement. The background is plain white, and there are no other markings or text present.

.....
.....
.....
..... [7]

(c) The program modules `SuppExists()`, `IsNewSupp()` and `CheckNewItem()` are part of a group of modules that are combined to create a complete stock control program.

Each module in the program is tested individually during development and is debugged as necessary. It is then added to the program and further testing performed.

(i) Identify this method of testing.
..... [1]

(ii) One of the modules does not work properly when it is added to the program.

Describe a testing method that can be used to address this problem so that testing can continue and other modules can be added.

.....
.....
.....
..... [2]

(d) A new module `AddItem()` will be used to add information to the `Stock.txt` file.

State the file mode that should be used for the algorithm within this module.
..... [1]

(e) A new module `FindItem()` searches for a given item in the `Stock.txt` file, which is already organised in ascending order of `ItemNum`.

Describe how this organisation may improve the efficiency of the algorithm.

.....
.....
.....
.....
.....
..... [3]

9618 Computer Science Paper 2 - Programming

Topical questions | 96 questions

35. Question part 1/2 | 9618 May-June 2023 P2 V2 Q4

4 A function `GetNum()` will:

1. take two parameters: a string and a character
2. count the number of times that the character occurs in the string
3. return the count.

Any comparison between characters needs to be case sensitive. For example, character 'a' and character 'A' are not identical.

Write pseudocode for function `GetNum()`.

[6]

[6]

BLANK PAGE

36. Question part 1/4 | 9618 May-June 2023 P2 V2 Q8

8 A computer shop assembles computers using items bought from several suppliers. A text file `Stock.txt` contains information about each item.

Information for each item is stored as a single line in the `Stock.txt` file in the format:

```
<ItemNum><SupplierCode><Description>
```

Valid item information is as follows:

	Format	Comment
ItemNum	4 numeric characters	unique number for each item in the range "0001" to "5999" inclusive
SupplierCode	3 alphabetic characters	to identify the supplier of the item
Description	a string	a minimum of 12 characters

The file is organised in ascending order of `ItemNum` and does **not** contain all possible values in the range.

A programmer has started to define program modules as follows:

Module	Description
<code>OnlyAlpha()</code> (already written)	- called with a parameter of type string - returns <code>TRUE</code> if the string contains only alphabetic characters, otherwise returns <code>FALSE</code>
<code>CheckInfo()</code>	- called with a parameter of type string representing a line of item information - checks to see whether the item information in the string is valid - returns <code>TRUE</code> if the item information is valid, otherwise returns <code>FALSE</code>

Topical questions | 96 questions

36. Question part 3/4 | 9618 May-June 2023 P2 V2 Q8

..... [7]

Module	Description
AddItem()	- called with a parameter of type string representing valid information for a new item that is not currently in the <code>Stock.txt</code> file - creates a new file <code>NewStock.txt</code> from the contents of the file <code>Stock.txt</code> and adds the new item information at the appropriate place in the <code>NewStock.txt</code> file

As a reminder, the file `Stock.txt` is organised in ascending order of `ItemNum` and does not contain all possible values in the range.

Write pseudocode for module `AddItem()`.

[illegible]

[7]

- Identify a method that can be used to continue testing the main program before the errors in these modules have been corrected **and** describe how this would work.

Method

Description

.....

.....

.....

[3]

2 A program stores a date of birth for a student using a variable, `MyDOB`, of type `DATE`.

(a) `MyDOB` has been assigned a valid value corresponding to Kevin’s date of birth.

Complete the pseudocode statement to test whether Kevin was born on a Thursday.

IF THEN [2]

(b) A function `CheckDate()` will take three integer parameters representing a day, month and year of a given date.

The function will validate the date of birth for a student that the parameters passed to it represent.

For a date to be valid, a student must be at least 18 in year 2020.

(i) Two of the parameter values can be checked without reference to the third parameter.

Describe these **two** checks.

Check 1

.....

.....

Check 2

.....

.....

[2]

(ii) Several values of the parameter representing the day can only be checked completely by referring to the value of **one other** parameter.

Describe this check.

.....

.....

.....

[2]

BLANK PAGE

- For example, the function call:

Write pseudocode for function `MakeString()`.

[6]

- (a) Write pseudocode for procedure `Select()`.

Parameter validation is **not** required.

[7]

(b) The check performed by procedure `Select()` on the last two digits is needed at several places in the program and will be implemented using a new function.

The new function `CheckNum()` will:

- allow the required sum to be specified (not just 6)
- check one number
- return an appropriate value.

Describe the function interface **and two** advantages of this modular approach.

Interface

.....

.....

.....

Advantage 1

.....

Advantage 2

.....

8 A computer shop assembles desktop computers, using items bought from several suppliers. A text file `Stock.txt` contains information about each item.

Information for each item is stored as a single line in the `Stock.txt` file in the format:

```
<ItemNum><SupplierCode><Description>
```

Item information is as follows:

	Format	Comment
ItemNum	4 numeric characters	unique number for each item in the range “0001” to “5999” inclusive
SupplierCode	3 alphabetic characters	code to identify the supplier of the item
Description	a string	a minimum of 12 characters

The file is organised in ascending order of `ItemNum` and does not contain all possible values in the range.

The programmer has defined the first program module as follows:

Module	Description
<code>ChangeSupp()</code>	- called with two parameters <code>Code1</code> and <code>Code2</code> of type string that represent valid supplier codes - creates a new file <code>NewStock.txt</code> from the contents of the file <code>Stock.txt</code> where any reference to <code>Code1</code> is replaced by <code>Code2</code> - returns a count of the number of items that have had their supplier code changed

9618 Computer Science Paper 2 - Programming

Topical questions | 96 questions

40. Question part 2/5 | 9618 May-June 2023 P2 V3 Q8

(a) Write pseudocode for module `ChangeSupp()`.

[8]

Module	Description
Report_1()	• takes a parameter of type string that represents a <code>SupplierCode</code> • searches the <code>Stock.txt</code> file for each line of item information that contains the given <code>SupplierCode</code> • produces a formatted report of items for the given <code>SupplierCode</code>, for example, for supplier DRG, the output could be: <pre> Report for Supplier: DRG Item Description 1234 USB Printer Cable 3m 1273 32GB USB Flash Drive 1350 Mouse Mat 320 x 240mm Number of items listed: 3 </pre>

Write pseudocode for module `Report_1()`.

[illegible]

Topical questions | 96 questions

[6]

(c) The format of the output from module `Report_1()` from **part (b)** is changed. The number of items listed is moved to the top of the report as shown in the example:

```
Report for Supplier: DRG
Number of items listed: 3

Item          Description

1234          USB Printer Cable 3m
1273          32GB USB Flash Drive
1350          Mouse Mat 320 x 240mm
```

(i) Explain why this new layout would increase the complexity of the algorithm.

.....

.....

.....

..... [2]

(ii) The algorithm will be modified to produce the report in the new format. The modified algorithm will be implemented so that the file `Stock.txt` is only read once.

Describe the modified algorithm.

.....

.....

.....

.....

.....

.....

..... [3]

41. Question part 1/2 | 9618 Oct-Nov 2023 P2 V1 Q1

1 The following pseudocode represents part of the algorithm for a program:

```
CASE OF ThisValue
  < 30      : Level ← "Low"           // less than 30
              Check ← 1
  < 20      : Level ← "Very Low"     // less than 20
              Check ← ThisValue / 2
  30 TO 40  : Level ← "Medium"       // between 30 and 40
              Check ← ThisValue / 3
              Data[ThisValue] ← Data[ThisValue] + 1
  > 40      : Level ← "High"
ENDCASE
```

(a) Complete the table by writing the answer for each row:

	Answer
The value assigned to <code>Level</code> when <code>ThisValue</code> is 40	
The value assigned to <code>Check</code> when <code>ThisValue</code> is 36	
The value assigned to <code>Level</code> when <code>ThisValue</code> is 18	
The number of elements in array <code>Data</code> that may be incremented	

[4]

(b) The pseudocode contains four assignments to variable `Level`. One of these assignments will never be performed.

Identify this assignment **and** explain why this is the case.

.....

.....

.....

.....

.....

.....

..... [3]

(c) The following line is added immediately before the `ENDCASE` statement:

```
OTHERWISE : Level ← "Undefined"
```

State why this assignment is never performed.

.....

..... [1]

(d) Give the appropriate data types for the variables `ThisValue`, `Check` and `Level`.

`ThisValue`

`Check`

`Level`

[3]

A function `TooMany()` will:

- (a)** Write pseudocode for the function `TooMany()`.

[6]

(b) The global array is changed to a 2D array, organised as 150 rows by 2 columns. It is declared in pseudocode as follows:

```
DECLARE Data : ARRAY[1:150, 1:2] OF STRING
```

The algorithm for the function in **part (a)** is changed. Strings will only be counted if **both** of the following conditions are true:

- The current row is an even number.
- The search string exactly matches the value in **either** column.

Write pseudocode to check these conditions.

Assume that the row index is contained in variable `Row` and the search string in variable `Search`.

.....

.....

.....

.....

.....

..... [3]

43. Question part 1/2 | 9618 Oct-Nov 2023 P2 V1 Q5

- 5 An algorithm is designed to find the smallest numeric value from an input sequence and count how many numeric values have been input.
An example of an input sequence is:

23, AB56, 17, 23ZW, 4, 10, END

Numeric input values are all integers and non-numeric input is ignored, except for the string "END" which is used to terminate the sequence.

The algorithm is expressed in pseudocode as shown:

```
DECLARE NextInput : STRING
DECLARE Min, Count, Num : INTEGER

Min ← 999
Count ← 0

REPEAT
  INPUT NextInput
  IF IS_NUM(NextInput) = TRUE THEN
    Num ← STR_TO_NUM(NextInput)
    IF Num > Min THEN
      Min ← Num
    ENDIF
    Count ← Count & 1
  ENDIF
UNTIL NextInput ← "END"

OUTPUT "The minimum value is ", Min, " and the count was ", Count
```

(a) The pseudocode contains three errors due to the incorrect use of **operators**.

Identify each error **and** state the correction required.

1

.....

2

.....

3

.....

(b) The operator errors are corrected and the algorithm is tested as follows:

The input sequence:

18, 4, ONE, 27, 189, ERIC, 3, 65, END

produces the output:

The minimum value is 3 and the count was 6

The algorithm is tested with a different test data sequence. The sequence contains a mix of integer and non-numeric values. It is terminated correctly but the algorithm produces unexpected results.

(i) Explain the problem with the algorithm.

.....

.....

.....

..... [2]

(ii) Give a sequence of **four** test data values that could be input to demonstrate the problem.

Value 1

Value 2

Value 3

Value 4 [2]

6 The pseudocode `OUTPUT` command starts each output on a new line.

- (a)** A new procedure `MyOutput()` will take a string and a Boolean parameter.
`MyOutput()` may be called repeatedly and will use concatenation to build a string using a global variable `MyString`, up to a maximum length of 255 characters.

`MyString` will be output in either of these two cases:

1. The Boolean parameter value is `TRUE`
2. The resulting string (after concatenation) would be longer than 255 characters.

If `MyString` is not output, the string is concatenated with `MyString`.

For example, the calls to `MyOutput()` given below would result in the output as shown:

```
MyOutput("Hello ", FALSE)
MyOutput("ginger ", FALSE)
MyOutput("cat", TRUE)
MyOutput("How are you?", TRUE)
```

Resulting output:

```
Hello ginger cat
How are you?
```

Notes:

- `MyString` is initialised to an empty string before `MyOutput()` is called for the first time.
- No string passed to `MyOutput()` will be longer than 255 characters.

(b) The design of the procedure given in **part (a)** is modified and `MyString` is changed from a global to a local variable declared in `MyOutput()`.

Explain why it does not work as expected.

..... [2]

8 A class of students are developing a program to send data between computers. Many computers are connected together to form a wired network. Serial ports are used to connect one computer to another.

Each computer:

- is assigned a unique three-digit ID
- has three ports, each identified by an integer value
- is connected to between one and three other computers.

Messages are sent between computers as a string of characters organised into fields as shown:

```
<STX><DestinationID><SourceID><Data><ETX>
```

Field number	Field name	Description
n/a	STX	a single character marking the start of the message (ASCII value 02)
1	DestinationID	three numeric characters that identify the destination computer
2	SourceID	three numeric characters that identify the source computer
3	Data	a variable length string containing the data being sent (Minimum length is 1 character)
n/a	ETX	a single character marking the end of the message (ASCII value 03)

For example, the following message contains the data "Hello Kevin" being sent from computer "101" to computer "232":

```
<STX>"232101Hello Kevin"<ETX>
```

Each computer will run a copy of the same program. Each program will contain a global variable, `MyID` of type string, that contains the unique ID of the computer in which the program is running.

The programmer has defined the first two program modules as follows:

Module	Description
<code>Transmit()</code> (already written)	• takes two parameters: ◦ a string containing a message ◦ an integer containing a port number • transmits the message using the given port
<code>SendFile()</code>	• takes three parameters: ◦ a string containing a text file name ◦ a string containing a Destination ID ◦ an integer containing a Port number • transmits the file one line at a time • transmits a final message with data string "****"

Assume:

[7]

(b) Module `SendFile()` is used to copy a file from one computer to another.

A module within the program running on the destination computer will receive the data and write it to a new file.

Explain why module `SendFile()` transmits the message with data string "*****" after the last line of the file.

.....

.....

.....

..... [2]

(c) One of the text files to be sent contains several blank lines (lines that do not contain any text).

(i) Explain why this is a problem.

.....

.....

.....

..... [2]

(ii) Explain how the message format could be changed to allow a blank line to be sent.

.....

.....

.....

..... [2]

BLANK PAGE

Question 8(d) starts on page 18.

(d) A new module has been defined:

Module	Description
GetField()	• takes two parameters: ◦ a string containing a message ◦ an integer containing a field number • If the field number is valid (in the range 1 to 3, inclusive), it returns a string containing the required field, otherwise it returns an empty string.

As a reminder, a message is defined as follows:

```
<STX><DestinationID><SourceID><Data><ETX>
```

Field number	Field name	Description
Not applicable	STX	a single character marking the start of the message (ASCII value 02)
1	DestinationID	three numeric characters that identify the destination computer
2	SourceID	three numeric characters that identify the source computer
3	Data	a variable length string containing the data being sent (Minimum length is 1 character)
Not applicable	ETX	a single character marking the end of the message (ASCII value 03)

- The value 99 must not be counted.

(a) Write pseudocode for the procedure `Count()`.

[6]

(b) The procedure `Count ()` is to be tested.

Typical test data would consist of odd and even values, for example:

23, 5, 64, 100, 2002, 1, 8, 900, 99

The purpose of this test would be to test a typical mix of even and odd values and check the totals.

Give **three** test data sequences that may be used to test **different** aspects of the procedure.

Do **not** include invalid data.

Sequence 1:

Test data

Purpose of test.

.....

Sequence 2:

Test data

Purpose of test.

.....

Sequence 3:

Test data

Purpose of test.

.....

47. Question part 1/2 | 9618 Oct-Nov 2023 P2 V2 Q5

- 5** A global 1D array of integers contains four elements, which are assigned values as shown:

```
Mix[1] ← 1  
Mix[2] ← 3  
Mix[3] ← 4  
Mix[4] ← 2
```

A procedure `Process()` manipulates the values in the array.

The procedure is written in pseudocode:

```
PROCEDURE Process(Start : INTEGER)  
    DECLARE Value, Index, Count : INTEGER  
  
    Index ← Start  
    Count ← 0  
  
    REPEAT  
        Value ← Mix[Index]  
        Mix[Index] ← Mix[Index] - 1  
        Index ← Value  
        Count ← Count + 1  
    UNTIL Count = 5  
  
    Mix[4] ← Count * Index  
  
ENDPROCEDURE
```

Complete the trace table on the opposite page by dry running the procedure when it is called as follows:

```
CALL Process(3)
```


- The procedure will create the number of text files specified.

For example, the call `CreateFiles("TestData", 3)` would result in the creation of the three files, `TestData.001`, `TestData.002` and `TestData.003`.

This is File TestData.002

Write pseudocode for `CreateFiles()`.

Assume both parameters are valid and that the integer value is between 1 and 999, inclusive.

[illegible]

(b) A module `CheckFiles()` will count the number of files produced by `CreateFiles()` in part **(a)**.

`CheckFiles()` will take a string representing a file name and return the number of files found.

(i) Identify the type of module that should be used for `CheckFiles()`.
..... [1]

(ii) Write the module header for `CheckFiles()`.
.....
..... [1]

(iii) State the file mode that should be used in `CheckFiles()`.
..... [1]

8 A class of students are developing a program to send data between computers. Many computers are connected together to form a wired network. Serial ports are used to connect one computer to another.

Each computer:

- is assigned a unique three-digit ID
- has three ports, each identified by an integer value
- is connected to between one and three other computers.

Data is sent as individual message strings.
Each string contains the destination ID (the ID of the computer that is to receive the message) followed by the data:

```
<DestinationID><Data>
```

Messages may pass through several computers on the way to their destination.
When a message arrives at a computer, that is **not** the destination, the program needs to forward it on to another computer using one of its serial ports.

The port to use is obtained from information that is stored in an array `RouteTable`.

`RouteTable` is a global 2D array of integers. It is declared in pseudocode as follows:

```
DECLARE RouteTable : ARRAY[1:6,1:3] OF INTEGER
```

The values in the first two columns of `RouteTable` define a range of ID values.
Column 3 gives the corresponding port number to use when forwarding the message to a computer with an ID within this range.

For example, the contents of `RouteTable` could be:

	Column 1	Column 2	Column 3
Row 1	100	199	1
Row 2	200	259	2
Row 3	-1	<undefined>	<undefined>
Row 4	260	399	2
Row 5	400	599	3
Row 6	600	999	1

In this example, a message that arrives with a `DestinationID` of "283" will be forwarded using port 2.

Row 3 in the example shows an unused row. These may occur anywhere. Unused rows have the column 1 element set to -1. The value of unused elements in the other two columns is undefined.

Module	Description
GetPort()	• takes a DestinationID as a parameter of type string • searches for the range corresponding to the DestinationID in the array • returns the port number, or returns -1 if no corresponding range is found

Assume `DestinationID` contains a valid three-digit string.

[7]

(b) Copies of the same program will run on each computer. The program contains a global variable `MyID` of type string, which contains the unique ID of the computer in which the program is running.

When messages are received, they are placed on one of two stacks. Stack 1 is used for messages that have reached their destination and stack 2 is used for messages that will be forwarded on to another computer.

Additional modules are defined:

Module	Description
<code>StackMsg()</code> (already written)	• takes two parameters: ◦ a string representing a message ◦ an integer representing the stack number • adds the message to the required stack • returns <code>TRUE</code> if the message is added to the required stack, otherwise returns <code>FALSE</code>
<code>ProcessMsg()</code>	• takes a message as a parameter of type string • ignores any message with a zero-length data field • extract the <code>DestinationID</code> from the message • checks whether the <code>DestinationID</code> is this computer or whether the message is to be forwarded • uses <code>StackMsg()</code> to add the message to the appropriate stack • outputs an error if the message could not be added to the stack

- (c) The program contains a module `GetFile()` which receives text files sent from another computer.

Lines from the file are sent one at a time. Each message contains one line and `ProcessMsg()` from part (b) adds each message as it is received onto stack 1.

Module `GetFile()` removes messages from stack 1 and writes the data to a text file.

There is a problem. Under certain circumstances, the received file does not appear as expected.

Assume that while a file is being received `ProcessMsg()` receives only messages containing lines from the file.

- (i) Describe the circumstances and explain the problem.

Circumstances

.....

Explanation

.....

.....

.....

[3]

- (ii) Suggest a more appropriate Abstract Data Type that could be used to store the messages that would not have the same problem.

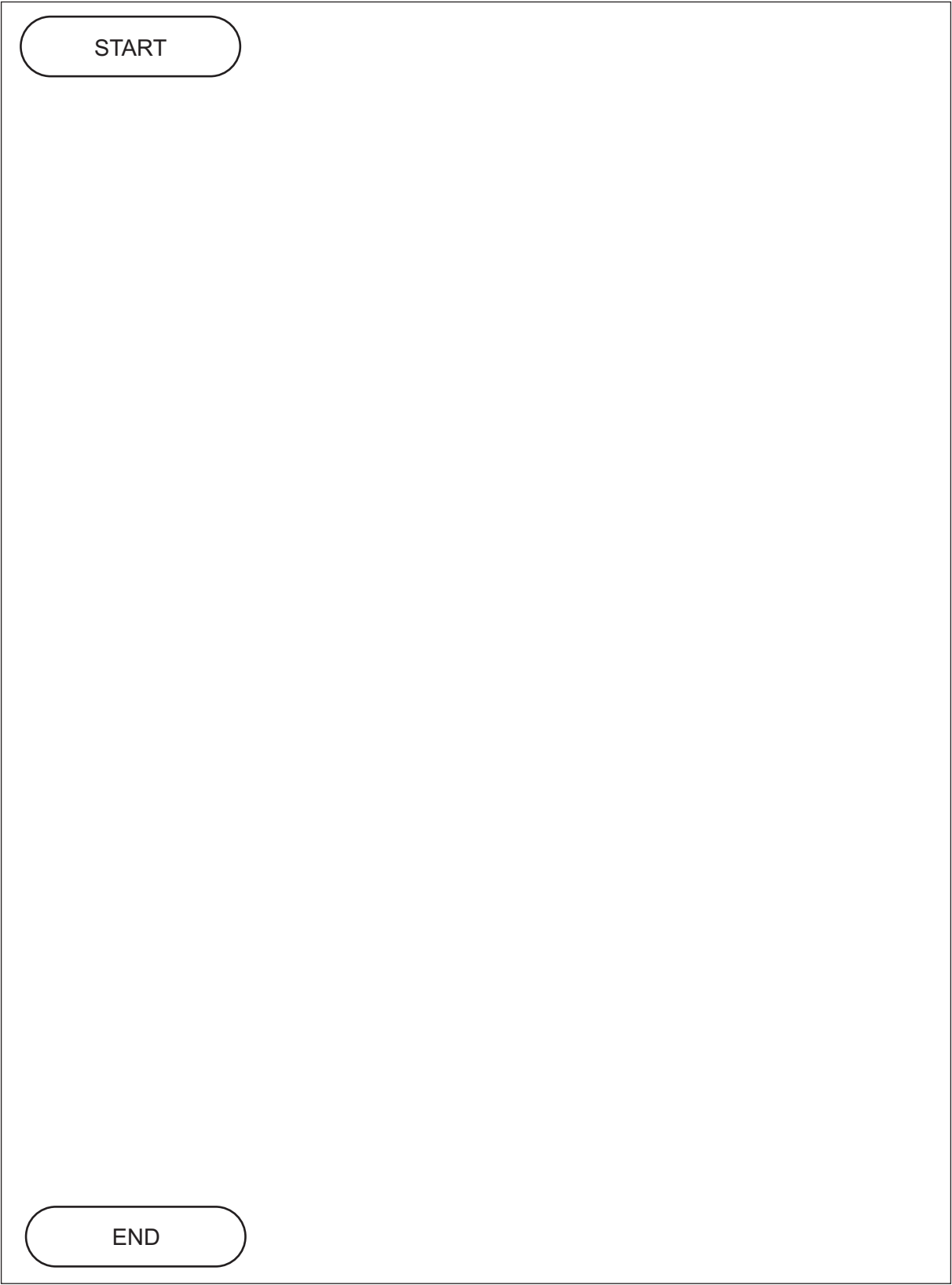
..... [1]

50. Question part 1/2 | 9618 Oct-Nov 2023 P2 V3 Q2

2 `Data` is a 1D array of integers, containing 30 elements. All element values are unique.

(a) An algorithm will output the index of the element with the smallest value.

Draw a program flowchart to represent the algorithm.



(b) The 30 data values could have been stored in separate variables rather than in an array.

Explain the benefits of using an array when designing a solution to part **(a)**.

.....
..... [2]

(c) The requirement changes. Array `Data` needs to hold 120 elements and each value may include a decimal place.

Write a pseudocode statement to declare the modified array.

.....
..... [2]

[6]

(b) Procedure `RandList()` is modified so that the random numbers are also written into a 1D array `Result`.

A new module is written to confirm that the numbers in the array are in ascending order.

This module contains an `IF` statement that performs a comparison between elements:

```
IF (Result[x + 1] = Result[x]) OR (Result[x] > Result[x + 1]) THEN
    Sequence ← FALSE
ENDIF
```

Write a simplified version of the conditional clause.

.....

.....

..... [1]

- 5** A global 1D array of integers contains four elements, which are assigned values as shown:

```
Mix[1] ← 4  
Mix[2] ← 2  
Mix[3] ← 3  
Mix[4] ← 5
```

A procedure `Process()` manipulates the values in the array.

The procedure is written in pseudocode as follows:

```
PROCEDURE Process(Start : INTEGER)  
    DECLARE Value, Index, Total : INTEGER  
  
    Index ← Start  
    Total ← 0  
  
    WHILE Total < 20  
        Value ← Mix[Index]  
        Total ← Total + Value  
  
        IF Index < 4 THEN  
            Mix[Index] ← Mix[Index] + Mix[Index+1]  
        ELSE  
            Mix[Index] ← Mix[Index] + Mix[1]  
        ENDIF  
        Index ← (Value MOD 4) + 1  
  
    ENDWHILE  
  
    Mix[1] ← Total * Index  
  
ENDPROCEDURE
```

Complete the trace table on the opposite page by dry running the procedure when it is called as follows:

```
CALL Process(2)
```


6 A function `TestNum()` will take a six-digit string as a parameter.

The function will test whether the string meets certain conditions and will return an integer value as follows:

Return value	Condition	Example
1	The last three digits are the same but non-zero.	"253444"
2	The last three digits are zero.	"253000"
3	The first three and last three digits are the same.	"410410"

The function will return the highest possible value for the given string.

If the string does not meet any of the conditions, zero is returned.

- 8 A class of students are developing a program to send data between computers. Many computers are connected together to form a wired network. Serial ports are used to connect one computer to another.

Each computer:

- is assigned a unique three-digit ID
- has three ports, each identified by an integer value
- is connected to between one and three other computers.

Messages are sent between computers as a string of characters organised into fields as shown:

```
<STX><DestinationID><SourceID><Data><ETX>
```

Field name	Description
STX	a single character marking the start of the message (ASCII value 02)
DestinationID	three numeric characters identifying the destination computer
SourceID	three numeric characters identifying the source computer
Data	a variable length string containing the data being sent (Minimum length is 1 character)
ETX	a single character marking the end of the message (ASCII value 03)

For example, the following message contains the data "Hello Jack" being sent from computer "202" to computer "454":

```
<STX>"454202Hello Jack"<ETX>
```

Each computer will run a copy of the same program. Each program will contain a global variable `MyID` of type string which contains the unique ID of the computer in which the program is running.

The first two program modules are defined as follows:

Module	Description
<code>GetData()</code> (already written)	• returns the data field from a message that has been received • If no message is available, the module waits until one has been received.
<code>ReceiveFile()</code>	• takes a file name as a parameter of type string • creates a text file with the given file name (no checking required) • writes the data field returned by <code>GetData()</code> to the file • repeats until the data field is "****", which is not written to the file • outputs a final message giving the total number of characters written to the file, for example: 132456 characters were written to newfile.txt

Module `GetData ()` has already been written and must be used.

[7]

[7]

(b) The use of the string "*****" as explained in the module description for `ReceiveFile()` may cause a problem.

Explain the problem and suggest a solution.

Problem

.....

.....

.....

Solution

.....

.....

.....

[3]

(c) Two new modules are defined, which will allow two users to exchange messages.

Module	Description
<code>Transmit()</code> (already written)	- takes two parameters: - a string representing a message - an integer representing a port number - transmits the message using the given port
<code>Chat()</code>	- takes two parameters: - a string representing a Destination ID - an integer representing a port number - extracts data from a received message using <code>GetData()</code> and outputs it - forms a message using data input by the user and sends it using <code>Transmit()</code> - repeats until either the output string or the sent string is "Bye"

Reminders:

- Each program contains a global variable `MyID` of type string which contains the unique ID of the computer in which the program is running.
- Messages are sent between computers as a string of characters organised into fields as shown:

<STX><DestinationID><SourceID><Data><ETX>

(d) Module `GetData()` returns the data field from a message that has been received. If no message is available, the module waits until one has been received.

Explain the limitation of this on module `Chat()` from part (c).

Describe a modification to `GetData()` to address this limitation.

Limitation

.....

.....

.....

.....

Modification

.....

.....

.....

[3]

55. Question part 1/2 | 9618 May-June 2024 P2 V1 Q1

1 An algorithm is developed in pseudocode before being coded in a programming language.

(a) The following table shows four valid pseudocode assignment statements.

Complete the table by giving an appropriate data type to declare each of the variables A, B, C and D.

Assignment statement	Data type
A ← LEFT(MyName, 1)	
B ← Total * 2	
C ← INT(ItemCost) / 3	
D ← "Odd OR Even"	

[4]

(b) Other variables in the program have example values as shown:

Variable	Value
Sorted	False
Tries	9
ID	"ZGAC001"

Complete the table by evaluating each expression, using the example values.

Expression	Evaluates to
Tries < 10 AND NOT Sorted	
Tries MOD 4	
TO_LOWER(MID(ID, 3, 1))	
LENGTH(ID & "xx") >= Tries	

[4]

(c) The variable names A, B, C and D in part (a) are **not** good programming practice.

(i) State why these variable names are **not** suitable.

.....
..... [1]

(ii) Identify **one** problem that these variable names might cause.

.....
..... [1]

(iii) The choice of suitable variable names is one example of good programming practice.

Give **one other** example.

.....
..... [1]

- total the element values in odd index locations (1, 3, 5 ... 97, 99)
- total the element values in even index locations (2, 4, 6 ... 98, 100)
- return one of three strings 'Odd', 'Even' or 'Same' to indicate which total is the greater, or whether the totals are the same.

[6]

BLANK PAGE

57. Question part 1/2 | 9618 May-June 2024 P2 V1 Q5

- 5** A global 1D array of strings contains three elements which are assigned values as shown:

```
Data[1] ← "aaaaaa"  
Data[2] ← "bbbbbb"  
Data[3] ← "cccccc"
```

Procedure `Process()` manipulates the values in the array.

The procedure is written in pseudocode as follows:

```
PROCEDURE Process(Format : STRING)  
    DECLARE Count, Index, L : INTEGER  
    DECLARE Result : STRING  
    DECLARE C : CHAR  
  
    Result ← "*****"  
  
    FOR Count ← 1 TO LENGTH(Format) STEP 2  
        C ← MID(Format, Count, 1)  
        L ← STR_TO_NUM(MID(Format, Count + 1, 1))  
  
        Index ← (Count + 1) DIV 2  
  
        CASE OF C  
            'X' : Result ← TO_UPPER(Data[Index])  
            'Y' : Result ← TO_LOWER(Data[Index])  
            'Z' : Result ← "***" & Data[Index]  
        ENDCASE  
  
        Data[Index] ← LEFT(Result, L)  
    NEXT Count  
  
ENDPROCEDURE
```

(a) Complete the trace table by dry running the procedure when it is called as follows:

```
CALL Process ("X3Y2W4")
```

Count	C	L	Index	Result	Data[1]	Data[2]	Data[3]

[6]

(b) The procedure is to be modified. If variable C is assigned a value other than 'X', 'Y' or 'Z', then procedure Error() is called and passed the value of variable C as a parameter.

This modification can be implemented by adding a **single line** of pseudocode.

(i) Write the single line of pseudocode.

..... [1]

(ii) State where this new line should be placed.

..... [1]

8 A teacher is designing a program to process pseudocode projects written by her students.

Each student project is stored in a text file.

The process is split into a number of stages. Each stage performs a different task and creates a new file named as shown:

File name	Comment
MichaelAday_src.txt	student project file produced by student Michael Aday
MichaelAday_S1.txt	file produced by stage 1
MichaelAday_S2.txt	file produced by stage 2

The teacher has defined the first program module as follows:

Module	Description
DeleteComment()	- called with a parameter of type string representing a line of pseudocode from a student’s project file - returns the line after removing any comments Note on comments: A comment starts with two forward slash characters and includes all the remaining characters on the line. The following example shows a string before and after the comment has been removed: Before: IF X2 > 13 THEN //check if limit exceeded After: IF X2 > 13 THEN

Topical questions | 96 questions

(a) Complete the pseudocode for module `DeleteComment()`.

This image shows a full page of a worksheet designed for handwriting practice. It features approximately 20 evenly spaced, horizontal dotted lines across the entire width of the page. The background is plain white, providing a clear guide for letter height and placement. There are no margins, text, or other markings present.

[8]

Module	Description
Stage_1 ()	- called with a parameter of type string representing a student name - creates a new stage 1 file - copies each line from the student's project file to the stage 1 file after removing any comment from each line - does not write blank lines to the stage 1 file - returns the number of lines written to the stage 1 file

Write pseudocode for module `Stage_1()`.

Module DeleteComment() must be used in your solution.

This image shows a full page of a document template designed for handwritten notes or essays. It features approximately 28 evenly spaced, thin grey horizontal lines extending across the entire width of the page. The margins are consistent on all sides, providing ample space for writing. There are no vertical lines, headers, footers, or other markings present on the page.

1 (a) The following table contains pseudocode examples.

Each example may contain statements that relate to one or more of the following:

- selection
- iteration (repetition)
- input/output.

Complete the table by placing **one or more** ticks (✓) in each row.

Pseudocode example	Selection	Iteration	Input/Output
FOR Index ← 1 TO 10 Data[Index] ← 0 NEXT Index			
WRITEFILE ThisFile, "*****"			
UNTIL Level > 25			
IF Mark > 74 THEN READFILE OldFile, Data ENDIF			

[4]

(b) Program variables have data types as follows:

Variable	Data type
MyChar	CHAR
MyString	STRING
MyInt	INTEGER

Complete the table by filling in each gap with a function (from the **insert**) so that each expression is valid.

Expression
MyInt ← (3.1415926)
MyChar ← ("Elwood", 3, 1)
MyString ← (..... (27.509))
MyInt ← (..... ("ABC123", 3))

[4]

(c) The variables given in part **(b)** are chosen during the design stage of the program development life cycle.

The choices are to be documented to simplify program maintenance.

State a suitable way of documenting the variables **and** give **one** piece of information that should be recorded, in addition to the data type.

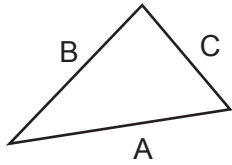
.....

.....

..... [2]

60. Question part 1/2 | 9618 May-June 2024 P2 V2 Q4

- 4 A triangle has sides of length A, B and C.



In this example, A is the length of the longest side.

This triangle is said to be right-angled if the following equation is true:

$$A \times A = (B \times B) + (C \times C)$$

A procedure will be written to check whether three lengths represent a right-angled triangle. The lengths will be input in any sequence.

The procedure `IsRA()` will:

- prompt and input three integer values representing the three lengths
- test whether the three lengths correspond to the sides of a right-angled triangle
- output a suitable message.

The length of the longest side may **not** be the first value input.

5 A program is being designed in pseudocode.

The program contains a global 1D array `Data` of type string containing 200 elements.

The first element has the index value 1.

A procedure `Process()` is written to initialise the values in the array:

```
PROCEDURE Process(Label : STRING)
  DECLARE Index : INTEGER
  Index ← 0
  INPUT Data[Index]
  WHILE Index < 200
    Index ← Index + 1
    CASE OF (Index MOD 2)
      0 : Data[Index] ← TO_UPPER(Label)
      1 : Data[Index] ← TO_LOWER(Label)
      OTHERWISE : OUTPUT "Alarm 1201"
    ENDCASE
  NEXT Index
  OUTPUT "Completed " & Index & " times"
ENDPROCEDURE
```

(a) (i) The pseudocode contains **two** syntax errors and **one** other error.

Identify the errors.

Syntax error 1

.....

Syntax error 2

.....

Other error

.....

[3]

(ii) The procedure contains a statement that is **not** needed.

Identify the pseudocode statement **and** explain why it is **not** needed.

Statement

Explanation

.....

[2]

- (b) After correcting all syntax errors, the pseudocode is translated into program code which compiles without generating any errors.

When the program is executed it unexpectedly stops responding.

Identify the type of error that has occurred.

..... [1]

- 6** A music player stores music in a digital form and has a display which shows the track being played.
- (a)** Up to 16 characters can be displayed. Track titles longer than 16 characters will need to be trimmed as follows:
- Words must be removed from the end of the track title until the resulting title is less than 14 characters.
 - When a word is removed, the space in front of that word is also removed.
 - Three dots are added to the end of the last word displayed when one or more words have been removed.

The table below shows some examples:

Original title	Display string															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Bat out of Hull	B	a	t		o	u	t		o	f		H	u	l	l	
Bohemian Symphony	B	o	h	e	m	i	a	n	.	.	.					
Paperbook Writer	P	a	p	e	r	b	o	o	k		W	r	i	t	e	r
Chris Sings the Blues	C	h	r	i	s		S	i	n	g	s	.	.	.		
Green Home Alabama	G	r	e	e	n		H	o	m	e	.	.	.			

- A function `Trim()` will:
- take a string representing the original title
 - return the string to be displayed.

- Assume:
- Words in the original title are separated by a single space character.
 - There are no spaces before the first word or after the last word of the original title.
 - The first word of the original title is less than 14 characters.

(b) Music is stored as a sequence of digital samples.

Each digital sample is a denary value in the range 0 to 99999999 (8 digits).

The samples are to be stored in a text file. Each sample is converted to a numeric string and 32 samples are concatenated (joined) to form a single line of the text file.

Each numeric string is 8 characters in length; leading '0' characters are added as required.

Example:

Sample	Denary value	String
1	456	"00000456"
2	48	"00000048"
3	37652	"00037652"
.....		
32	673	"00000673"

The example samples will be stored in the text file as a single line:

"000004560000004800037652...00000673"

(i) Identify one drawback of adding leading '0' characters to each numeric string.

.....
..... [1]

(ii) Suggest an alternative method of storing the samples which does **not** involve adding leading '0' characters but which would still allow each individual sample to be extracted.

.....
.....
..... [1]

(iii) State **one** drawback of the alternative method given in part (b)(ii).

.....
..... [1]

BLANK PAGE

8 A teacher is designing a program to process pseudocode projects written by her students.

Each student project is stored in a text file.

The process is split into a number of stages. Each stage performs a different task and creates a new file.

For example:

File name	Comment
MichaelAday_src.txt	Student project file produced by student Michael Aday
MichaelAday_S1.txt	File produced by stage 1
MichaelAday_S2.txt	File produced by stage 2

(a) Suggest a reason why the teacher’s program has been split into a number of stages **and** give the benefit of producing a different file from each stage.

Reason

.....

Benefit

.....

[2]

(b) The teacher has defined the first program module as follows:

Module	Description
DeleteSpaces()	- called with a parameter of type string representing a line of pseudocode from a student’s project file - returns the line after removing any leading space characters The following example shows a string before and after the leading spaces have been removed: Before: " IF X2 > 13 THEN" After: "IF X2 > 13 THEN"

(c) Two modules are defined:

Module	Description
DeleteComment() (already written)	- called with a parameter of type string representing a line of pseudocode from a student's project file - returns the line after removing any comment
Stage_2()	- called with two parameters: - a string representing an input file name - a string representing an output file name - copies each line from the input file to the existing output file having first removed all leading spaces and comments from that line - does not write blank lines to the output file - outputs a final message giving the number of blank lines removed

The procedure will:

- Values input in the first part are totalled using global variable `TotalA` and those input in the second part are totalled using global variable `TotalB`.

(a) Write pseudocode for the procedure `TwoParts()`.

[6]



(b) The value zero denotes the split between the two parts of the sequence.

The requirement changes and now there may be up to 20 parts.

(i) Identify a suitable data structure that could be used to store the different total values.

..... [2]

(ii) Describe **three** benefits of using the data structure given in part (b)(i).

1

.....

2

.....

3

.....

[3]

The program contains the following declaration:

```
DECLARE Data : ARRAY[1:1000] OF STRING
```

```
PROCEDURE ArrayInitialise(Label : STRING)
  DECLARE Index : INTEGER
  Index  $\leftarrow$  1
  WHILE Index <= 1000
    CASE OF (Index MOD 2)
      0 : Data[Index]  $\leftarrow$  FormatA(Label)
          Index  $\leftarrow$  Index + 1
      1 : Data[Index]  $\leftarrow$  FormatB(Label)
          Index  $\leftarrow$  Index + 1
    ENDCASE
  ENDWHILE
ENDPROCEDURE
```

Functions `FormatA()` and `FormatB()` apply fixed format case changes to the parameter string.

(a) The design of the procedure does **not** use the most appropriate loop construct.

Suggest a **more appropriate** construct that could be used **and** explain your choice.

Construct

Explanation

[2]

(b) The algorithm calls one of the functions `FormatA()` and `FormatB()` each time within the loop.

Explain why this is **not** efficient **and** suggest a more efficient solution.

[4]

[4]



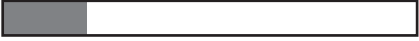
BLANK PAGE

6 A program displays a progress bar to inform the user of the progress of tasks that take a significant time to complete, such as those involving file transfer operations.

Task progress is divided into 11 steps. Each step represents the amount of progress as a percentage. An image is associated with each step and each image is stored in a different file.

Different progress bar images may be selected. For a given image, files all have the same filename root, with a different suffix.

The table illustrates the process for using the image with filename root `BargraphA`

Step	Percentage progress	Image filename	Image
1	< 10	BargraphA-1.bmp	
2	>= 10 and < 20	BargraphA-2.bmp	
3	>= 20 and < 30	BargraphA-3.bmp	
   			
9	>= 80 and < 90	BargraphA-9.bmp	
10	>= 90 and < 100	BargraphA-10.bmp	
11	100	BargraphA-11.bmp	

A procedure `Progress()` will:

- be called with two parameters:
 - an integer representing the percentage progress (0 to 100 inclusive)
 - a string representing the image filename root
- generate the full image filename
- call a procedure `Display()` using the full image filename as the parameter.



[6]

[6]



(b) The definition of procedure `Progress()` is provided here for reference:

- A procedure `Progress()` will:
- be called with two parameters:
 - an integer representing the percentage progress (0 to 100 inclusive)
 - a string representing the image filename root
 - generate the full image filename
 - call a procedure `Display()` using the full image filename as the parameter.

`Progress()` will be rewritten and a new module `Progress2()` produced with these requirements:

- an additional parameter of type integer will specify the total number of steps
- the image filename will be returned (procedure `Display()` will **not** be called from within `Progress2()`).

(i) Write pseudocode for the new module header.

.....

.....

..... [2]

(ii) State **one** benefit of increasing the number of steps.

.....

..... [1]



8 A teacher is designing a program to process pseudocode projects written by her students.

The program analyses a student project and extracts information about each module that is defined (each procedure or function). This information is stored in a global 2D array `ModInfo` of type string.

A module header is the first line of a module definition and starts with either of the keywords `PROCEDURE` or `FUNCTION`.

An example of part of the array is given below. Row 10 of the array shows that a procedure header occurs on line 27 and row 11 shows that a function header occurs on line 35. "P" represents a procedure and "F" represents a function:

	x = 1	x = 2	x = 3
ModInfo[10, x]	"27"	"P"	"MyProc(Z : CHAR)"
ModInfo[11, x]	"35"	"F"	"MyFun(Y : CHAR) RETURNS BOOLEAN"

The string stored in column 3 is called the module description. This is the module header **without** the keyword.

A valid module header will:

- be at least 13 characters long
- start with the keyword `PROCEDURE` or `FUNCTION`. The keyword may appear in either upper or lower case (or a mix of both) and **must** be followed by a space character.

The teacher has defined the first program module as follows:

Module	Description
Header()	• called with a parameter of type string representing a line of pseudocode • if the line is a valid procedure header, returns a string: "P<Module description>" • if the line is a valid function header, returns a string: "F<Module description>" • otherwise, returns an empty string For example, given the string: "FUNCTION Zap(X : INTEGER) RETURNS CHAR" Header() returns the string: "FZap(X : INTEGER) RETURNS CHAR"



..... [7]

[7]



Module	Description
<code>FindModules()</code>	- called with a parameter of type string representing a student project file name - uses module <code>Header()</code> to check each line of the project - assigns values to the <code>ModInfo</code> array for each module declaration in the student project

	x = 1	x = 2	x = 3
ModInfo[10, x]	"27"	"P"	"MyProc(Z : CHAR)"
ModInfo[11, x]	"35"	"F"	"MyFun(Y : CHAR) RETURNS BOOLEAN"

Assume that the array contains enough rows for the number of modules in each project.

[illegible]

67. Question part 4/4 | 9618 May-June 2024 P2 V3 Q8



21



.....

.....

.....

.....

.....

.....

.....

..... [8]

1 A program will calculate the tax payable based on the cost of an item.

Calculations will occur at many places in the program and these involve the use of one of three tax rates.

Tax rate values represent a percentage. For example, a tax rate value of 5.23 represents 5.23%. In this case, the tax payable on an item costing \$100 would be \$5.23.

Tax rate values are used at several places within the program. One example is given in pseudocode as follows:

```
HighRate ← FALSE
CASE OF ItemCost
  <= 50 : TaxRate ← 3.75 // tax rate of 3.75%
  <= 200 : TaxRate ← 5.23 // tax rate of 5.23%
  > 200 : TaxRate ← 6.25 // tax rate of 6.25%
        HighRate ← TRUE
ENDCASE
TaxPayable ← ItemCost * TaxRate // tax payable
```

(a) The pseudocode contains a logical error.

Identify the error **and** suggest a correction.

Error

.....

Correction

.....

[2]

(b) During the design of the program, tax rate values have been used wherever they are needed as shown in the pseudocode example above. Tax rates do not change while the program runs.

(i) Identify a more appropriate way of representing the tax rate values in the final program.

..... [1]

(ii) Describe the benefits of your answer to part (b)(i) with reference to this program.

.....

.....

.....

.....

.....

..... [3]



(c) Give the **appropriate** data type for the variables in the following table, as used in the pseudocode:

Variable name	Data type
HighRate	
TaxPayable	

[2]

(d) The final CASE condition (> 200) in the pseudocode example could be replaced with a keyword.

Give the keyword.

..... [1]

2 A program uses three global integer variables `HH`, `MM` and `SS` to represent the current time in hours, minutes and seconds using the 24-hour clock notation.

Midnight would be represented as 00:00:00 (HH:MM:SS). If the variables HH, MM and SS contained the values 16, 30 and 10 respectively, then the time would be 16:30:10 or just after 4.30 in the afternoon.

A procedure `Tick()` will be called every second.

The procedure `Tick()` will:

- update the value in `SS` each time it is called
- update the values in `HH` and `MM` as appropriate
- call a procedure `CheckAlarm()` at the start of each minute
- call a procedure `NewDay()` whenever the time reaches midnight.

Complete the pseudocode for procedure `Tick()`.

```
PROCEDURE Tick()
```

[illegible]

ENDPROCEDURE

69. Question part 2/2 | 9618 Oct-Nov 2024 P2 V1 Q2



BLANK PAGE

4 A program includes the following assignment statement:

```
Result ← STR_TO_NUM(x) / STR_TO_NUM(y)
```

When the program evaluates the expression in the statement, it performs a calculation.

Variable `Result` is of type real and variables `x` and `y` are of type string.

Two checks are required before the calculation is performed:

- 1. The two strings represent valid numeric values.
- 2. The numeric value of string `y` is not zero.

(a) Identify the type of error that could occur if these checks are **not** carried out **and** state a cause of this error.

Type

.....

Cause

.....

[2]

(b) The designer considers implementing the **checks and calculation** as a module (a procedure or a function). One reason for this is that the same checks and calculations are performed at several places in the program.

Give **another** reason why this is a suitable approach **and** state what is avoided by this approach.

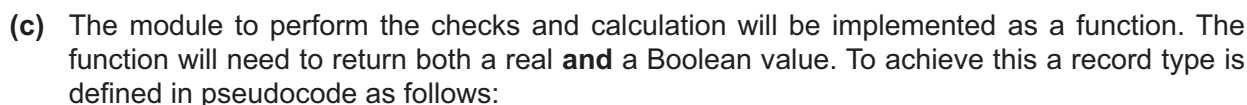
Reason

.....

Avoided

.....

[2]



The function `Evaluate()` will:

- Write pseudocode for the function `Evaluate()`.

[6]

71. Question part 1/4 | 9618 Oct-Nov 2024 P2 V1 Q8

8 An exam paper has a maximum of 75 marks. One of five pass grades (A to E) is assigned, depending on the mark obtained. The lowest mark for a given grade is known as the grade boundary. For example, if the grade boundary for an A grade is 65 marks, then any candidate who achieves a mark of 65 or above will be awarded an A. A grade of U is awarded for marks below the E grade boundary.

The five grade boundaries are stored in a global 1D array `GradeBoundary` of type integer.

For example:

Element	Value	Comment
<code>GradeBoundary[1]</code>	65	The minimum mark for an A grade.
<code>GradeBoundary[2]</code>	57	The minimum mark for a B grade.
<code>GradeBoundary[3]</code>	43	The minimum mark for a C grade.
<code>GradeBoundary[4]</code>	35	The minimum mark for a D grade.
<code>GradeBoundary[5]</code>	27	The minimum mark for an E grade.

A global 2D array `Result` of type integer contains candidate marks for the exam. Each row relates to one candidate. Column 1 contains the candidate mark and column 2 contains the unique candidate ID.

For example, for the fourth and fifth candidates:

Element	Mark	Element	ID
<code>Result[4, 1]</code>	56	<code>Result[4, 2]</code>	1074832
<code>Result[5, 1]</code>	54	<code>Result[5, 2]</code>	2573839

There are more rows in the array than candidates who sit the exam. Any unused rows will be at the end of the array.

Candidate papers that are given a mark within two marks of any grade boundary must be checked.

For example, given the values in the example grade boundaries above, any paper that was awarded between 41 and 45 marks (inclusive) would need to be checked.

A program is being written to identify papers that need to be checked.

The programmer has defined the first program module as follows:

Module	Description
<code>CheckMark()</code>	- called with a parameter of type integer representing a candidate mark - returns <code>TRUE</code> if the mark is within 2 of any of the five grade boundaries, otherwise returns <code>FALSE</code>

9618 Computer Science Paper 2 - Programming

Topical questions | 96 questions

71. Question part 2/4 | 9618 Oct-Nov 2024 P2 V1 Q8



(a) Write pseudocode for module `CheckMark()`.

[6]

[6]

Topical questions | 96 questions

71. Question part 3/4 | 9618 Oct-Nov 2024 P2 V1 Q8



(b) A second module is defined:

Module	Description
CheckAll()	- called with a parameter of type integer representing the number of candidate marks in the <code>Result</code> array - uses <code>CheckMark()</code> to check each candidate mark - for each paper that needs to be checked, write the corresponding candidate ID on a separate line in a new file named <code>GRLIST.txt</code> - outputs a message with a count of how many papers need to be checked

Write pseudocode for module `CheckAll()`.

CheckMark () must be used to check each individual mark.

[illegible]

71. Question part 4/4 | 9618 Oct-Nov 2024 P2 V1 Q8



.....

.....

.....

.....

.....

.....

.....

.....

..... [8]

(c) The requirement changes. Instead of a new file, the module described in part (b) needs to add the corresponding candidate ID for each paper that needs to be checked to an **existing** file.

Explain the change that will need to be made to `CheckAll()`.

.....

..... [1]

When a card number is displayed, all the characters **except** the last four are replaced with the asterisk character ' * '.

The function `Conceal()` will take a string representing a card number and return a modified string.

Original string	Modified string
"1234567890"	"*****7890"
"1234567897652"	"*****7652"
"1234567890123456"	"*****3456"

- take a numeric string as a parameter representing the card number
- return a string in which the asterisk character replaces all except the last four characters of the card number parameter.

[6]



(b) The requirements have been changed. `Conceal()` will now be written as a procedure which will process 100 card numbers each time it is called.

The card numbers will be stored in a 2D array `CardNumber`. The original string will be stored in column one and the modified string in column two.

(i) Write pseudocode to declare the array.

.....
..... [2]

(ii) The new procedure `Conceal()` will write the modified string to the corresponding element in column two.

The array `CardNumber` is passed as a parameter to the new procedure `Conceal()`.

Identify how this parameter should be specified in the new procedure header.

.....
..... [1]

9618 Computer Science Paper 2 - Programming

Topical questions | 96 questions

73. Question part 1/2 | 9618 Oct-Nov 2024 P2 V2 Q4

- 4 A global integer variable `Tick` is always incremented every millisecond (1000 times per second) regardless of the other programs running.

The value of `Tick` can be read by any program but the value should not be changed.

Assume that the value of `Tick` does not overflow.

As an example, the following pseudocode algorithm would output "Goodbye" 40 seconds after outputting "Hello".

```
DECLARE Start : INTEGER

OUTPUT "Hello"
Start ← Tick

REPEAT
    //do nothing
UNTIL Tick = Start + 40000

OUTPUT "Goodbye"
```

A program is needed to help a user to time an event such as boiling an egg.

The time taken for the event is known as the elapsed time.

The program contains a procedure `Timer()` which will:

- take two integer values representing an elapsed time in minutes and seconds
- use the value of variable `Tick` to calculate the elapsed time
- output a warning message 30 seconds before the elapsed time is up
- output a final message when the total time has elapsed.

For example, to set an alarm for 5 minutes and 45 seconds the program makes the following call:

```
CALL Timer(5, 45)
```

When 5 minutes and 15 seconds have elapsed, the program will output:

```
"30 seconds to go"
```

When 5 minutes and 45 seconds have elapsed, the program will output:

```
"The time is up!"
```


74. Question part 1/2 | 9618 Oct-Nov 2024 P2 V2 Q5

5 A program contains a global 1D array `Data` with 100 elements of type `INTEGER`.

The program contains a function `Process()` expressed in pseudocode as follows:

```
FUNCTION Process(Number : INTEGER, Label : STRING) RETURNS STRING
  DECLARE Index, Count : INTEGER
  DECLARE ReturnValue : STRING

  Count ← INT(100 / Number)
  Index ← Data[Number]

  CASE OF (Index MOD 2)
    0 : ReturnValue ← TO_UPPER(RIGHT(Label, Count))
    1 : ReturnValue ← "*****"
  ENDCASE

  RETURN RetVal
ENDFUNCTION
```

(a) Run-time errors can be generated in different ways. For example, a run-time error will be generated if a function is called with invalid parameters.

The pseudocode contains three statements that could generate a run-time error.

Write the **three** statements **and** explain how each could generate a run-time error.

Statement 1

Explanation

.....

Statement 2

Explanation

.....

Statement 3

Explanation

.....

74. Question part 2/2 | 9618 Oct-Nov 2024 P2 V2 Q5



(b) One type of run-time error can cause a program to stop responding ('freezing').

Identify a particular type of programming construct that can generate this type of error **and** explain why it occurs.

Construct

Explanation

.....

.....

[2]

(c) The function `Process()` contains a selection construct using a `CASE` structure.

Write pseudocode using a single selection construct with the same functionality **without** using a `CASE` structure.

.....

.....

.....

.....

..... [2]

6 A shop sells sandwiches and snacks. The owner chooses a ‘daily special’ sandwich which is displayed on a board outside the shop. Each ‘daily special’ has two different fillings and is made with one type of bread.

The owner wants a program to randomly choose the ‘daily special’ sandwich.

The program designer decides to store the possible sandwich fillings in a 1D array of type string.

The array is declared in pseudocode as follows:

```
DECLARE Filling : ARRAY [1:35] OF STRING
```

Each element contains the name of one filling.

An example of the first five elements is as follows:

Index	Element value
1	"Cheese"
2	"Onion"
3	"Salmon"
4	"Anchovies"
5	"Peanut Butter"

A second 1D array stores the possible bread used:

```
DECLARE Bread : ARRAY [1:10] OF STRING
```

Each element contains the name of one type of bread.

An example of the first three elements is as follows:

Index	Element value
1	"White"
2	"Brown"
3	"Pitta"

Both arrays may contain unused elements. The value of these will be an empty string and they may occur anywhere in each array.

A procedure `Special()` will output a message giving the ‘daily special’ sandwich made from **two** randomly selected different fillings and **one** randomly selected bread.

Unused array elements must **not** be used when creating the ‘daily special’ sandwich.

Using the above examples, the output could be:

```
"The daily special is Cheese and Onion on Brown bread."
```

Assume that both arrays are global.

[illegible]

[7]

75. Question part 3/3 | 9618 Oct-Nov 2024 P2 V2 Q6



- (b) The owner decides that some combinations of fillings do not go well together. For example, anchovies and peanut butter.

Describe how the design could be changed to prevent certain combinations being selected.

.....

.....

.....

.....

.....

.....

.....

..... [2]

8 A program is being developed to implement a game for up to six players.

During the game, each player assembles a team of characters. At the start of the game there are 45 characters available.

Each character has four attributes, as follows:

Attribute	Examples	Comment
Player	0, 1, 3	The player the character is assigned to.
Role	Builder, Teacher, Doctor	The job that the character will perform in the game.
Name	Bill, Lee, Farah, Mo	The name of the character. Several characters may perform the same role, but they will each have a unique name.
Level	14, 23, 76	The skill level of the character. An integer in the range 0 to 100 inclusive.

The programmer has defined a record type to define each character.

The record type definition is shown in pseudocode as follows:

```
TYPE CharacterType
    DECLARE Player : INTEGER
    DECLARE Role : STRING
    DECLARE Name : STRING
    DECLARE Level : INTEGER
ENDTYPE
```

The `Player` field indicates the player to which the character is assigned (1 to 6). The field value is 0 if the character is **not** assigned to any player.

The programmer has defined a global array to store the character data as follows:

```
DECLARE Character : ARRAY[1:45] OF CharacterType
```

At the start of the game all record fields are initialised, and all `Player` field values are set to 0

The programmer has defined a program module as follows:

Module	Description
<code>Assign()</code>	- called with two parameters: - an integer representing a player - a string representing a character role - search the <code>Character</code> array for an unassigned character with the required role - If found, assign the character to the given player and output a confirmation message, for example: "Bill the Builder has been assigned to player 3" - If no unassigned character with the required role is found, output a suitable message.

9618 Computer Science Paper 2 - Programming

Topical questions | 96 questions

76. Question part 2/4 | 9618 Oct-Nov 2024 P2 V2 Q8



(a) Write pseudocode for module `Assign()`.

[7]



- The module is defined as follows:

Module	Description
Save ()	- form a string from each record with fields separated by the character '^' - write each string to a separate line of the new file named SaveFile.txt

```
PROCEDURE Save ()
```

This image shows a full page of a handwriting practice worksheet. It consists of multiple sets of three horizontal dashed lines, providing a guide for letter height and placement. The lines are evenly spaced across the entire page, leaving ample room for writing practice. There is no text or other markings on the page.

[71]



(c) The program is changed and the record type definition is modified as follows:

```
TYPE CharacterType
  DECLARE Player : INTEGER
  DECLARE Role : STRING
  DECLARE Name : STRING
  DECLARE Level : INTEGER
  DECLARE Status : BOOLEAN
ENDTYPE
```

Describe how the additional Boolean field may be stored with the rest of the fields on one line of a text file.

.....

.....

.....

..... [2]

(d) The save operation is to be extended so that multiple files may be saved as the game progresses. This will allow the user to restore the game from any saved position. The filename must reflect the sequence in which the files are saved.

Describe a method that would allow multiple files to be saved **and** give an example of **two** consecutive filenames.

Method

.....

Example

..... [2]

1 (a) The following table contains pseudocode examples.

Each example may contain statements that relate to one or more of the following:

- selection
- iteration (repetition)
- subroutine (procedure or function).

Complete the table by placing **one or more** ticks ('✓') in each row.

Pseudocode example	Selection	Iteration	Subroutine
FOR Index ← 1 TO 3 IF Safe[Index] = TRUE THEN Flap[Index] ← 0 ENDIF NEXT Index			
CASE OF Compound(3)			
REPEAT UNTIL AllDone() = TRUE			
WHILE Result[3] <> FALSE			

[4]

(b) Complete the table by giving the appropriate data type in each case.

Variable	Example data value	Data type
Available	TRUE	
Received	"18/04/2021"	
Index	100	

[3]

(c) Evaluate each expression in the table by using the data values shown in part (b).

Write 'ERROR' if the expression contains an error.

Expression	Evaluates to
Available AND NOT(Index > 100)	
Index MOD 30	
NUM_TO_STR(Index + "33")	

[3]

3 The implementation of a linked list uses an integer variable and a 1D array `List` of type `Node`.

Record type `Node` is declared in pseudocode as follows:

```
TYPE Node
  DECLARE Data : STRING
  DECLARE Pointer : INTEGER
ENDTYPE
```

The array `List` is declared in pseudocode as follows:

```
DECLARE List : ARRAY[1:200] OF Node
```

The linked list will operate as follows:

- Integer variable `HeadPointer` will store the array index for the first node in the linked list.
- The `Pointer` field of a node contains the index value of the next array element (the next node) in the linked list.
- The value 0 is used as a null pointer.

(a) (i) State why the value 0 has been selected as the null pointer.

.....

..... [1]

(ii) Give the range of valid values that could be assigned to variable `HeadPointer`.

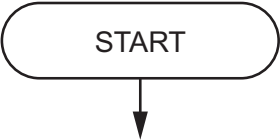
..... [1]

78. Question part 2/4 | 9618 Oct-Nov 2024 P2 V3 Q3



- (b) The array `List` will be initialised so that each node points to the following node.
The last node will contain a null pointer.

Complete the program flowchart to represent the algorithm for this operation.



78. Question part 3/4 | 9618 Oct-Nov 2024 P2 V3 Q3



- (c) An algorithm outputs the `Data` field from all nodes in the array `List`. The order the `Data` is output should be the same order it is stored in the linked list.

Describe the algorithm in **four** steps.

Do **not** use pseudocode statements in your answer.

Step 1

.....

.....

Step 2

.....

.....

Step 3

.....

.....

Step 4

.....

.....



BLANK PAGE

79. Question part 1/2 | 9618 Oct-Nov 2024 P2 V3 Q4

- 4 An examination paper has a maximum of 75 marks. One of five pass grades (A to E) is assigned, depending on the mark obtained. The lowest mark for a given grade is known as the grade boundary.

A program is being written to process examination marks.

The five grade boundaries are stored in a global 1D array `GB` of type `INTEGER`, for example:

Index	Value	Comment
1	65	The minimum mark for an A grade.
2	57	The minimum mark for a B grade.
3	43	The minimum mark for a C grade.
4	35	The minimum mark for a D grade.
5	27	The minimum mark for an E grade.

Any paper that achieves a mark within 2 marks of a grade boundary must be checked. Using the given table, a paper with 45 marks would need to be checked.

- (a) The pseudocode algorithm to determine whether a paper should be checked is as shown. The mark for the paper is stored in variable `Mark`. Global variables `Mark`, `Index`, `Upper` and `Lower` are declared as integers.

Complete the pseudocode.

```
FOR Index ← 1 TO .....
    Lower ← GB[Index] - 2
    Upper ← .....
    IF Mark ..... AND Mark ..... THEN
        OUTPUT "Check this paper"
    ENDIF
NEXT Index
```

[4]

- (b) An alternative algorithm to determine if a paper needs to be checked uses a global 1D array `Check`, containing 76 elements of type `BOOLEAN`. The indices of the array are from 0 to 75 (inclusive), corresponding to the range of possible marks.

An element value in `Check` is `TRUE` if the index is within 2 marks of a grade boundary. For example, in the case where the C grade boundary is 43 the corresponding part of the `Check` array would be as follows:

Index	Value
40	FALSE
41	TRUE
42	TRUE
43	TRUE
44	TRUE
45	TRUE
46	FALSE

← The grade boundary for a C grade

Topical questions | 96 questions

79. Question part 2/2 | 9618 Oct-Nov 2024 P2 V3 Q4



- Describe how an algorithm would use the `Check` array to determine whether this paper should be checked.

[2]

- Note it can be assumed that the maximum grade boundary value for A is 70 and the minimum value for E is 15.

Write pseudocode for the procedure.

[6]

6 In some countries, on the third Sunday in March, daylight saving time begins when clocks move forward by one hour.

A module `AdjustClock()` will take an integer parameter representing a year. The module will return an integer value representing the number of the day in March on which the clocks move forward.

For example, the following line of pseudocode would assign `DayNumber` the value 20:

```
DayNumber ← AdjustClock(2022)
```

Write pseudocode for the function `AdjustClock()`.

Date functions from the **insert** should be used in your solution.

[illegible]

8 A program is being developed to implement a game for up to six players.

During the game, each player assembles a team of characters. At the start of the game there are 45 characters available.

Each character has four attributes, as follows:

Attribute	Examples	Comment
Player	0, 1, 3	The player the character is assigned to.
Role	Builder, Teacher, Doctor	The job that the character will perform in the game.
Name	Bill, Lee, Farah, Mo	The name of the character. Several characters may perform the same role, but they will each have a unique name.
Skill level	14, 23, 76	An integer in the range 0 to 100, inclusive.

The programmer has defined a record type to define each character. The record type definition is shown in pseudocode as follows:

```
TYPE CharacterType
    DECLARE Player : INTEGER
    DECLARE Role : STRING
    DECLARE Name : STRING
    DECLARE SkillLevel : INTEGER
ENDTYPE
```

The `Player` field indicates the player to which the character is assigned (1 to 6). This field value is 0 if the character is **not** assigned to any player.

The programmer has defined a global array to store the character data, as follows:

```
DECLARE Character : ARRAY[1:45] OF CharacterType
```

At the start of the game all record fields are initialised, and all `Player` fields are set to 0

The programmer has defined a program module as follows:

Module	Description
<code>Count()</code>	- called with two parameters: - an integer representing a player - a string representing a character role - searches the <code>Character</code> array for characters with the given role that are assigned to the given player - counts the number of assigned characters and sums their total skill level - outputs the result of the search if characters with the given role are found, for example: "Player 3 has 4 characters with the role of Teacher and the total skill level is 65" - if no characters with the given role are found, outputs: "No characters with that role are assigned to this player"

Topical questions | 96 questions

81. Question part 2/4 | 9618 Oct-Nov 2024 P2 V3 Q8



(a) Complete the pseudocode for module `Count()`.

```
PROCEDURE Count(ThisPlayer : INTEGER, ThisRole : STRING)
```

This image shows a single sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

ENDPROCEDURE

Topical questions | 96 questions

81. Question part 3/4 | 9618 Oct-Nov 2024 P2 V3 Q8



- (b)** The `Character` array data has been saved in the text file `SaveFile.txt`. Each line of the file contains one element of the array (one record).

New modules are defined:

Module	Description
<code>Extract()</code> (already written)	- called with two parameters: - a string representing a complete line from the text file - an integer representing a field number (see structure below) - returns a string representing the required field
<code>Restore()</code>	- opens the text file <code>SaveFile.txt</code> - reads lines from the file and assigns values to each record in the <code>Character</code> array using data from each line of the file

As a reminder, the record structure is repeated here:

```

TYPE CharacterType
  DECLARE Player : INTEGER           //Field number 1
  DECLARE Role : STRING              //Field number 2
  DECLARE Name : STRING              //Field number 3
  DECLARE SkillLevel : INTEGER       //Field number 4
ENDTYPE

```

Write pseudocode for module `Restore()`.

You must use the module `Extract()`.

This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

81. Question part 4/4 | 9618 Oct-Nov 2024 P2 V3 Q8



.....

.....

.....

.....

..... [7]

(c) The game can last for several days and users often find that they have to close and rerun the game program many times in order to complete it.

Describe the benefit of using the file `SaveFile.txt` as described in part (b).

.....

.....

.....

..... [2]

82. Question part 1/3 | 9618 May-June 2025 P2 V1 Q2

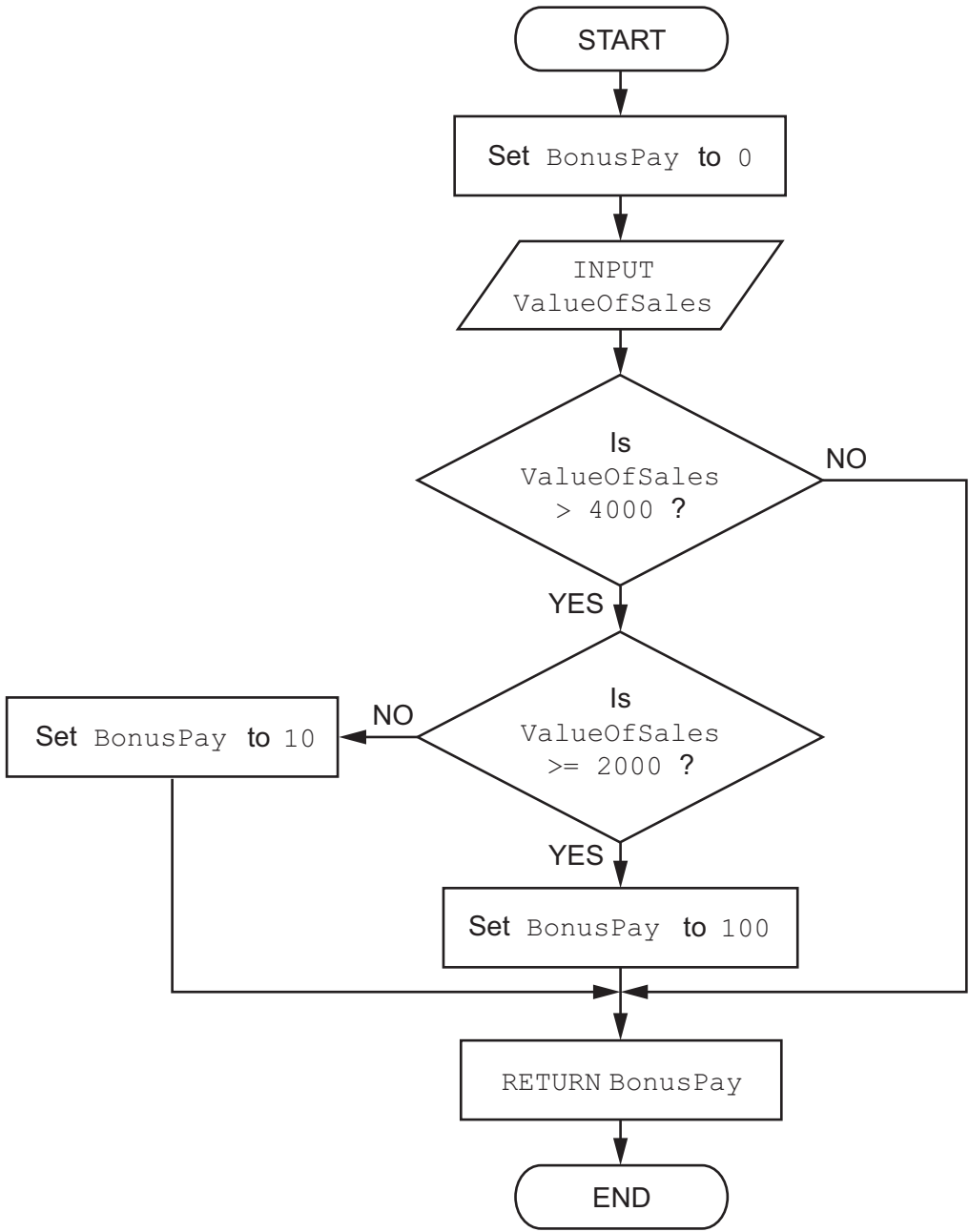
2 A program is being developed to calculate the pay of employees working for a company.

A function `CalculateBonus()` calculates bonus pay based on the value of sales.

Bonus pay is calculated as shown in the table.

Value of sales (in dollars)	Bonus pay (in dollars)
below 2000	0
between 2000 and 4000 inclusive	10
above 4000	100

A flowchart for the function `CalculateBonus()` has been designed.



82. Question part 2/3 | 9618 May-June 2025 P2 V1 Q2



(a) The flowchart contains logic errors.

One logic error is that `BonusPay` will **not** be set to 10 although the `ValueOfSales` input is between 2000 and 4000 inclusive.

(i) Explain why this error occurs.

.....

..... [1]

(ii) Explain how this error could be corrected.

.....

..... [1]

(b) There are different ways to reduce the risk of errors when developing the new program, such as the use of constants.

(i) State a value that could be replaced by a constant in the function `CalculateBonus()`

..... [1]

(ii) Explain how the use of constants helps to reduce the risk of programming errors.

.....

.....

.....

..... [2]

(iii) One other way that can reduce the risk of errors when writing the program is the use of library routines.

Explain how library routines can reduce the risk of programming errors.

.....

..... [1]

82. Question part 3/3 | 9618 May-June 2025 P2 V1 Q2



(c) All the logic errors have been corrected, and the program has been coded.

Identify and describe **two** other types of error that the program could contain.

Type of error

Description

.....

.....

Type of error

Description

.....

.....

[4]

- step 1 – randomly generate an integer value between 1 and 100 inclusive
- step 2 – prompt the user to input an integer value
- step 3 – output an appropriate message if the value input was too high; then repeat from **step 2**
- step 4 – output an appropriate message if the value input was too low; then repeat from **step 2**
- step 5 – output an appropriate message if the value input was the same value that was randomly generated; then end the program.

Assume no input validation is needed.

[illegible]

84. Question part 1/2 | 9618 May-June 2025 P2 V1 Q6

6 A program monitors the speed of vehicles as they move around a large building site.

Each vehicle contains a sensor which reads an integer value that represents the speed of the vehicle. The value is expected to be in the range 0 to 60 inclusive.

The sensors cannot read values less than 0.

A program module has been written to validate the values read by the sensors.

(a) A test plan is needed to fully test the module.

Complete the table. The first line has been completed for you.

Assume the sensors generate only integer values.

Type of test data	Test data value	Expected outcome
normal	36	data item is accepted

[4]

(b) Each sensor used in the system has a unique sensor ID number in the range 1 to 50.

The program that is used to monitor the speed of each vehicle reads each sensor value every second and stores this value along with the sensor’s unique ID into a global 2D array `Reading`

The global array `Reading` has been declared as follows:

```
DECLARE Reading : ARRAY[1:2000, 1:2] OF INTEGER
```

The array contains 4000 elements organised as 2000 rows and 2 columns.

Column 1 contains the sensor value, and column 2 contains the sensor ID.

When 2000 sensor readings have been taken, the array is full and the system stops taking any more sensor readings until the array has been processed.

A procedure `Sort()` is needed to sort the array into ascending order of sensor value using an efficient bubble sort algorithm.

7 A program is being developed to implement a customer loyalty scheme for a coffee shop.

The programmer has decided that the following data items need to be stored for each customer:

Data item	Description
customer ID	a unique six-digit string
points	an integer value that is increased by one for every cup of coffee the customer orders

When a customer visits the shop and orders coffee, the scheme operates as follows:

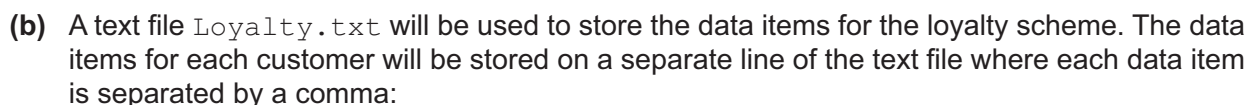
- The total number of points is increased by the number of coffees ordered.
- If just one cup of coffee is ordered and the number of points goes above 10, then:
 - the cup of coffee they have just ordered is given to them free of charge
 - the number of points is reduced by 11.
- If the order is for multiple coffees and the number of points goes above 10, then:
 - they get one coffee free of charge for every 11 points
 - the number of points is reduced by 11 for each free coffee.

For example, the:

- customer currently has 9 points
- customer orders 16 cups of coffee
- total number of points now becomes 25
- customer gets 2 free coffees and now has 3 points left.

The programmer has defined a program module that is called every time a customer places an order:

Module	Description
<code>CustomerOrder()</code>	• called with two integer parameters: ◦ the number of coffees ordered ◦ the current number of points • output a suitable message giving the number of free coffees • return a value for the new points total.



<CustomerID>, <Points>

The contents of the text file `Loyalty.txt` will always be stored in ascending order by customer ID.

When the data items are read from or written to the text file `Loyalty.txt`, they may need to be converted to the appropriate data type.

Each customer has a unique customer ID starting at "100001" with this value increasing by one each time a new customer joins the loyalty scheme.

When a customer joins the loyalty scheme, they are assigned the next customer ID and value of points is set to 0.

For example, if the loyalty scheme has 204 customers and a new customer joins the loyalty scheme, the following line is added to the text file `Loyalty.txt`:

"100205,0"

You can assume that the number of customers in the loyalty scheme will never be more than 9000.

The programmer has defined a program module as follows:

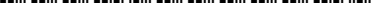
Module	Description
AddNewCustomers()	- called with an integer parameter representing the number of new customers to be added to the loyalty scheme - adds a new line, containing the required information, to the text file <code>Loyalty.txt</code> for each customer added to the loyalty scheme - outputs each new customer ID added to the loyalty scheme

(i) Write pseudocode for module `AddNewCustomers()`

Assume that there is at least **one** customer already in the loyalty scheme.

[illegible]

Topical questions | 96 questions



[8]

- Explain the changes that will need to be made to the module `AddNewCustomers()`

..... [4]

[4]

- 1 A program is being developed for the management of a sports centre.
- (a) The programmer developing the software decides to use modules (procedures or functions) and local variables. They also decide not to use global variables.

(i) State **two** reasons why the programmer decides to use modules.

1

.....

2

.....

[2]

(ii) State **one** difference between local and global variables.

.....

.....

[1]

(iii) State **two** benefits of using local variables.

1

.....

2

.....

[2]
- (b) The pseudocode design contains a number of expressions.

Complete each pseudocode expression so that it evaluates to the value shown.

Refer to the **insert** for the list of pseudocode functions and operators.
- | Expression | Evaluates to |
|--|--------------|
| (68) | 'D' |
| (04/02/2025) | 2 |
| TRUE | FALSE |
| (..... ("Court1.4Upper" , ..., ...)) | 1.4 |
- [5]

86. Question part 2/2 | 9618 May-June 2025 P2 V3 Q1



(c) The table lists some of the variables used in the program.

Complete the table by writing the most appropriate data type for each variable.

Variable	Use of variable	Data type
MemberCount	stores how many members have used the sports centre each day	
TotalTakings	stores the total amount of money taken each day	
BookingConfirmed	stores the state of a booking for an exercise class; a booking is either confirmed or not confirmed	
MemberDOB	to calculate when to send an email with a birthday message to a member	

[4]

- The module will:

step 1 – take a positive integer parameter representing the number of times the dice will be rolled

step 2 – simulate **one** roll of the dice by generating a random integer between 1 and 6 inclusive

step 3 – output the integer generated

step 4 – repeat, as required from **step 2**

step 5 – calculate the average value of the random integers generated

step 6 – return the average value.

Write pseudocode for the module `RollDice()`

This image shows a full page of blank handwriting practice paper. It features multiple sets of horizontal lines, each consisting of a solid top line, a dashed midline, and a solid bottom line, providing a guide for letter height and placement. The lines are evenly spaced across the entire page.

Topical questions | 96 questions

88. Question part 1/2 | 9618 May-June 2025 P2 V3 Q5

5 A module `Parity()` takes a string as a parameter. The parameter has the identifier `BitString` and it represents a binary value.

The module will concatenate a single character to the end of `BitString` by applying one of the two rules:

- '0' if Bitstring contains an even number of 1s
- '1' if Bitstring contains an odd number of 1s.

The modified value of `BitString` is then returned.

For example:

Parameter	String returned	Explanation
"0010010110"	"0010010110 0 "	there are an even number of 1s in the string passed to <code>Parity()</code> so a '0' is concatenated to the end of <code>BitString</code>
"101010"	"101010 1 "	there are an odd number of 1s in the string passed to <code>Parity()</code> so a '1' is concatenated to the end of <code>BitString</code>

Write pseudocode for the module `Parity()`

Assume the parameter `BitString` can only contain the characters '0' and '1'

[illegible]

7 A program is being developed to implement a customer loyalty scheme for a coffee shop.

The programmer has decided that the following data items need to be stored for each customer:

Data item	Description
customer ID	a unique six-digit string
loyalty points	an integer value that depends on how often the customer visits the coffee shop
last visit date	the date the customer last visited the coffee shop

The programmer has defined a program module:

Module	Description
UpdateVisit()	- called with two parameters: - loyalty points of type <code>INTEGER</code> - last visit date of type <code>DATE</code> - change the loyalty points: - increase by four if the last visit date and the current date are in the same month - otherwise increase by one - change the last visit date to the current date - the changed values must be available to the code that follows the call to <code>UpdateVisit()</code>

(a) (i) Write pseudocode for module `UpdateVisit()`

Assume the customer has made at least **one** previous visit.

[5]

89. Question part 2/5 | 9618 May-June 2025 P2 V3 Q7



(ii) The programmer decides to amend the `UpdateVisit()` module so that loyalty points are further increased if the customer visits the coffee shop the same day of the week as their last visit.

Write the pseudocode for this condition.

.....

..... [1]

89. Question part 3/5 | 9618 May-June 2025 P2 V3 Q7



(b) A text file `Loyalty.txt` will be used to store the data items for the loyalty scheme.

The text file `Loyalty.txt` contains only one line of data for each customer.

Each data item is separated by a comma:

`<CustomerID>,<LoyaltyPointsString>,<LastVisitDateString>`

`LastVisitDateString` is always eight characters in length in the format: DDMMYYYY.

An example of how a line of data will be stored in the text file `Loyalty.txt` is:

`"100123,132,05032025"`

This example shows that the customer with an ID of 100123 had 132 loyalty points following their last visit to the coffee shop on 05/03/2025.

If a customer visits the coffee shop on a Monday, the value of their loyalty points is increased by 10.

The programmer has defined **two** more program modules:

Module	Description
<code>FindCustomer()</code> (is already written)	- called with a parameter of type <code>STRING</code> that represents a customer ID - returns the line of data read from the text file <code>Loyalty.txt</code> containing the data items for that customer
<code>MondayCheck()</code>	- called with a parameter of type <code>STRING</code> that represents the customer ID of a customer - calls <code>FindCustomer()</code> - extracts the loyalty points from the string returned by <code>FindCustomer()</code> - increases the loyalty points for the current customer by 10 if the day visited is a Monday - returns an integer value representing the loyalty points

The module `FindCustomer()` must be used and assume it returns a valid string for the current customer.

[8]

89. Question part 5/5 | 9618 May-June 2025 P2 V3 Q7



(ii) A date is stored in `LastVisitDateString` in the format:

DDMMYYYY

- DD is a 2-digit string
- MM is a 2-digit string
- YYYY is a 4-digit string.

For example, the date 03/09/2024 is stored as "03092024"

An algorithm is needed to convert the string stored in `LastVisitDateString` to data type `DATE` which is then stored in the variable `LastVisitDate`.

Complete the pseudocode for this algorithm:

Assume that `LastVisitDateString` has been declared and contains valid data.

DECLARE DayInt, MonthInt, YearInt : INTEGER

DECLARE LastVisitDate : DATE

.....

.....

.....

.....

.....

.....

.....

.....

90. Question part 1/4 | 9618 Oct-Nov 2025 P2 V1 Q8

- 8
- A program is being developed to manage student book loans from a college library. Students may borrow up to five books at a time from the library.

The programmer has defined a record type to define each loan.

The record data items are:

Data item	Data type	Comment
StudentID	STRING	the unique ID of the student who has borrowed the book
BookID	STRING	the unique ID of the book being borrowed
OnLoan	BOOLEAN	TRUE if the book has not been returned

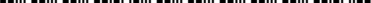
The programmer has defined a global array `Loan` to store 5000 loan records.

There are more elements in the array than books in the library. Unused elements have the `StudentID` set to an empty string. These may occur anywhere in the array.

The programmer has defined the first program module:

Module	Description
<code>OKToBorrow()</code>	- called with a parameter of type <code>STRING</code> representing a <code>StudentID</code> - search the array for loan records for the specified student - output a suitable message to say whether the student may, or may not, borrow another book

Topical questions | 96 questions



..... [7]

Topical questions | 96 questions

90. Question part 3/4 | 9618 Oct-Nov 2025 P2 V1 Q8



(b) A second module is defined:

Module	Description
ReturnBook()	- called with two parameters of type <code>STRING</code> representing a <code>StudentID</code> and a <code>BookID</code> - searches the array for the relevant loan record - when found, sets <code>OnLoan</code> to <code>FALSE</code> and returns <code>TRUE</code> - if a loan record is not found, or the book has already been returned, then returns <code>FALSE</code>

Write efficient pseudocode for the module `ReturnBook()`

Assume that each student is only allowed to borrow each book only once. That means that there will be no more than **one** loan record for a given combination of student and book.

[illegible]

90. Question part 4/4 | 9618 Oct-Nov 2025 P2 V1 Q8



.....

.....

.....

.....

.....

..... [8]

(c) It is decided to introduce a system of fines for books that have been borrowed for too long.

Two new requirements are defined:

- 1. Each loan has a maximum length, represented as a number of days.
- 2. Each book in the library will be assigned one of three categories. Each category has a different maximum loan length.

Record structure and program design changes are needed to meet these two requirements.

Outline the changes that are necessary to meet the **two** requirements.

.....

.....

.....

.....

.....

..... [2]

1 (a) The table contains pseudocode examples.

Each example may contain statements that relate to one or more of:

- selection
- iteration (repetition)
- subroutines (procedures or functions).

Complete the table by placing **one or more** ticks (✓) in each row.

Pseudocode example	Selection	Iteration	Subroutine
IF Status = FALSE THEN FOR Count ← 1 TO 20 CALL Reset(Count) NEXT Count ENDIF			
OTHERWISE : Status ← TRUE			
WHILE AllDone() = TRUE			
30 : NextChar ← 'X'			

[4]

(b) Complete the table by giving the appropriate data type.

Variable	Example data value	Data type
Result	5.42	
MonthLetter	"JFMAMJJASOND"	
Birthday	15/11/2009	

[3]

(c) Evaluate each expression in the table by using the data values shown in (b).

Write 'ERROR' if the expression contains an error.

Expression	Evaluates to
INT(Result) + 1 > 6	
NUM_TO_STR(LENGTH(MonthLetter))	
NUM_TO_STR(Result + "3.2")	
MID(MonthLetter, MONTH(Birthday) - 2, 1)	

[4]

92. Question part 1/2 | 9618 Oct-Nov 2025 P2 V2 Q3

- 3 A text file `OldFile.txt` contains IDs, names and email addresses for members of a club. Three information items are stored for each member and each item is stored on a separate line.

The example shows the information for the first two members in the first six lines of the file:

Line in file	Information item	Example data
1	member 1 ID	"AB1234"
2	member 1 name	"Freddie Jones"
3	member 1 email address	"FreddieJ909@Cambridge.org"
4	member 2 ID	"BC2345"
5	member 2 name	"Sue Smith"
6	member 2 email address	"Sue1024@Cambridge.org"

The member ID string is always two letters followed by four digits.

The file design is to be changed so that information for each user is stored as a single line of the file, with the character `'\'` used as a separator between data items.

For example, the single line for member 1 will be:

"AB1234\Freddie Jones\FreddieJ909@Cambridge.org"

An algorithm will produce a new file `NewFile.txt` from the contents of `OldFile.txt`

Assume:

- `LineX`, `LineY` and `LineZ` are of type `STRING` and are used to store the three items of information for each member
- `NewString` is a temporary variable of type `STRING`
- `OldFile.txt` exists and contains valid data.

- (a) The algorithm to create the new file is expressed in steps.

Complete the following numbered steps:

- Step 1 : open the file in
- Step 2 : open the file in
- Step 3 : read a line from and store in
- Step 4 : read a line from and store in
- Step 5 : read a line from and store in
- Step 6 : set `NewString` to
- Step 7 : write `NewString` to
- Step 8 : repeat from step until
- Step 9 : close both files.



- Explain why this is a suitable character.

..... [1]

- For example:

Information item	Example data
member 1 ID	"AB1234"
member 1 name	"Freddie Jones"
member 1 email address	"FreddieJ909@Cambridge.org"
member 1 new information 1	"En/98&*(?\\D7iP"
member 1 new information 2	"23\\CoboL"

- all the information items for each member are stored on a single line of `NewFile.txt`
- it is possible to extract each information item after the line is read.

Assume the ' \ ' character is not used in any email address.

[3]

8 A program is being developed to manage student book loans from a college library.

The programmer has defined a record type to define each loan.

The data items are:

Data item	Data type	Comment
StudentID	STRING	the unique ID of the student who has borrowed the book
BookID	STRING	the unique ID of the book being borrowed
OnLoan	BOOLEAN	TRUE if the book has not been returned

The programmer has defined a global array `Loan` to store 7000 loan records.

There are more elements in the array than books in the library. Unused elements have the `StudentID` set to an empty string. These may occur anywhere in the array.

The programmer has defined a program module:

Module	Description
<code>CountLoans()</code>	- called with a parameter of type <code>STRING</code> representing a <code>StudentID</code> - counts the number of books currently on loan to the specified student - counts the number of books that the student has already returned - output both counts together with a suitable message

93. Question part 3/5 | 9618 Oct-Nov 2025 P2 V2 Q8



(b) As a reminder, a global array `Loan` stores 7000 loan records with data items for each loan record:

Data item	Data type	Comment
<code>StudentID</code>	<code>STRING</code>	the unique ID of the student who has borrowed the book
<code>BookID</code>	<code>STRING</code>	the unique ID of the book being borrowed
<code>OnLoan</code>	<code>BOOLEAN</code>	<code>TRUE</code> if the book has not been returned

A new module is defined:

Module	Description
<code>NewLoan()</code>	- called with two parameters of type <code>STRING</code> representing a <code>StudentID</code> and a <code>BookID</code> - searches the array for an unused loan record - if found, updates the loan record and returns <code>TRUE</code>, otherwise returns <code>FALSE</code>

93. Question part 5/5 | 9618 Oct-Nov 2025 P2 V2 Q8



(c) When a book is returned, the loan record will have its `OnLoan` data set to `FALSE`

A new procedure `Archive()` will mark these records as unused after first saving them for future reference.

Explain how these records can be saved for future reference.

.....

.....

.....

..... [2]

(d) It is decided to extend the program so that a new module `Reminder()` will send an email to the student three days before the book is due to be returned.

Outline the changes that will need to be made to the data stored and how this data will be used to generate the reminder email.

Data

.....

.....

.....

Use

.....

.....

..... [2]

94. Question part 1/2 | 9618 Oct-Nov 2025 P2 V3 Q1

1 A user requires a program that implements complex data encryption to a recognised international standard. A programmer has started the design of the program.

(a) The programmer decides to use a library routine to provide part of the solution. One reason for this is that library routines may perform functions that the programmer is unable to program themselves.

State **three** other benefits of using library routines in the development of the program.

1

.....

2

.....

3

.....

[3]

(b) The programmer has identified a different part of the algorithm that would be appropriate to implement as a subroutine (a procedure or a function); no suitable library routine exists for this part.

(i) State **two** reasons why the programmer may decide to use a subroutine.

1

.....

2

.....

[2]

(ii) A function header in pseudocode is defined as:

```
FUNCTION Pass2 (Count : INTEGER) RETURNS BOOLEAN
```

Complete the table by describing the terms used in the function header.

The first row has been completed.

Term	Description
Pass2	the name of the function
Count	
BOOLEAN	

[2]

94. Question part 2/2 | 9618 Oct-Nov 2025 P2 V3 Q1



(c) Variables in the program have example values:

Variable	Example value
DoB	23/6/2011
Multiplier	2.5
AddressLine[1]	"35 Lincoln Avenue"

Complete the table by evaluating each expression using the example values:

Expression	Evaluates to
LENGTH(NUM_TO_STR(Multiplier))	
MONTH(DoB) > 4	
15 + STR_TO_NUM(MID(AddressLine[1], 2, 1))	

[3]

"101,456,219,754,328"

A procedure `Store()` will:

- Complete the pseudocode for `Store()`

```
PROCEDURE Store()
```

[illegible]

[6]

95. Question part 2/2 | 9618 Oct-Nov 2025 P2 V3 Q4



BLANK PAGE

96. Question part 1/4 | 9618 Oct-Nov 2025 P2 V3 Q5

- 5 A global 1D array `Num` of integers contains four elements. A program assigns values to these elements as shown:

```
Num[1] ← 1
Num[2] ← 2
Num[3] ← 5
Num[4] ← 3
```

A procedure `Process()` manipulates the values in the array.

The procedure is written in pseudocode:

```
PROCEDURE Process(Start : INTEGER)
    DECLARE CaseVar, Index, Count : INTEGER

    Index ← Start
    Count ← 0

    WHILE Count <= 20
        CaseVar ← Num[Index]
        CASE OF CaseVar
            1 : Num[Index] ← Num[Index] + Index //clause one
                Index ← Index + 1
                Count ← Count + 1

            2 : Num[Index] ← Num[Index] + Index //clause two
                Index ← Index + 2
                Count ← Count + 2

            3 : Num[Index] ← Num[Index] * 2      //clause three
                Index ← Index + 3
                Count ← Count - 1

            4 : Count ← Count + 4                //clause four
            OTHERWISE : Count ← 20
        ENDCASE

        Index ← (Index MOD 4) + 1

    ENDWHILE

ENDPROCEDURE
```

Topical questions | 96 questions



CALL Process (1)

[illegible]

[5]

96. Question part 3/4 | 9618 Oct-Nov 2025 P2 V3 Q5



(b) As a reminder, the CASE structure in the pseudocode is:

```
CASE OF CaseVar
  1 : Num[Index] ← Num[Index] + Index //clause one
      Index ← Index + 1
      Count ← Count + 1

  2 : Num[Index] ← Num[Index] + Index //clause two
      Index ← Index + 2
      Count ← Count + 2

  3 : Num[Index] ← Num[Index] * 2      //clause three
      Index ← Index + 3
      Count ← Count - 1

  4 : Count ← Count + 4                //clause four
  OTHERWISE : Count ← 20
ENDCASE
```

The CASE structure could be optimised by combining two existing clauses.

(i) Identify the CaseVar values for these **two** existing clauses.

..... [1]

(ii) Write pseudocode for a single clause to replace the **two** clauses identified in (b) (i).

.....
.....
..... [2]

96. Question part 4/4 | 9618 Oct-Nov 2025 P2 V3 Q5



BLANK PAGE