w22 qp 21:

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]

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]

(ii) `Calculate()` is changed to a function that returns the value of the evaluated expression.

Write the header for the function in pseudocode.

.....

..... [1]

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

Further modules are defined as follows:

Module	Description
`CheckLine()`	• takes a line from a project file as a parameter of type string • returns zero if the line is blank or contains no syntax error, otherwise returns an error number as an integer

`CountErrors()`	• takes two parameters:
	  ○ the name of a project file as a string
	  ○ the maximum number of errors as an integer
	• uses `CheckFile()` to test the project file. Outputs an error message and ends if the project file is not valid
	• calls `CheckLine()` for each line in the project file
	• counts the number of errors
	• outputs the number of errors or a warning message if the maximum number of errors is exceeded

(b) `CountErrors()` is called to check the project file `Jim01Prog.txt` and to stop if more than 20 errors are found.

Write the pseudocode statement for this call.

.....

..... [2]

(c) Write pseudocode for module `CountErrors()`. Assume `CheckFile()` and `CheckLine()` have been written and can be used in your solution.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

w22 qp 22:

1 **(a)** A programmer is developing an algorithm to solve a problem. Part of the algorithm would be appropriate to implement as a subroutine (a procedure or a function).

(i) State **two** reasons why the programmer may decide to use a subroutine.

1 .....

.....

2 .....

.....

[2]

(ii) A procedure header is shown in pseudocode:

```
PROCEDURE MyProc(Count : INTEGER, Message : STRING)
```

Give the correct term for the identifiers `Count` and `Message` **and** explain their use.

Term .....

Use .....

.....

.....

.....

[2]

6 **(a)** The factorial of an integer number is the product of all the integers from that number down to 1.

In general, the factorial of n is $n \times (n-1) \times ... \times 2 \times 1$

For example, the factorial of 5 is $5 \times 4 \times 3 \times 2 \times 1 = 120$

In this question, n will be referred to as the `BaseNumber`.

A function `FindBaseNumber()` will:

- be called with a positive, non-zero integer value as a parameter
- return `BaseNumber` if the parameter value is the factorial of the `BaseNumber`
- return −1 if the parameter value **is not** a factorial.

For example:

Parameter value	Value returned
120	5
12	−1
6	3
1	1

`FindBaseNumber(12)` will return −1 because 12 is not a factorial.

Write pseudocode for the function `FindBaseNumber()`.

[7]

w22 qp 23:

5 **(a)** A text string contains three data items concatenated as shown:

<StockID><Description><Cost>

Item lengths are:

Item	Length
StockID	5
Description	32
Cost	the remainder of the string

A procedure `Unpack()` takes four parameters of type string. One parameter is the original text string. The other three parameters are used to represent the three data items shown in the table and are assigned values within the procedure. These values will be used by the calling program after the procedure ends.

(c) `Unpack()` is part of a program made up of several modules. During the design stage, it is important to follow good programming practice. One example of good practice is the use of meaningful identifier names.

Give the reason why this is good practice. Give **two other** examples of good practice.

Reason .....

.....

.....

Example 1 .....

.....

.....

Example 2 .....

.....

.....

[3]

(i) Write pseudocode for the procedure `Unpack()`.

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [6]

(ii) Explain the term **procedure interface** with reference to procedure `Unpack()`.

.....

.....

.....

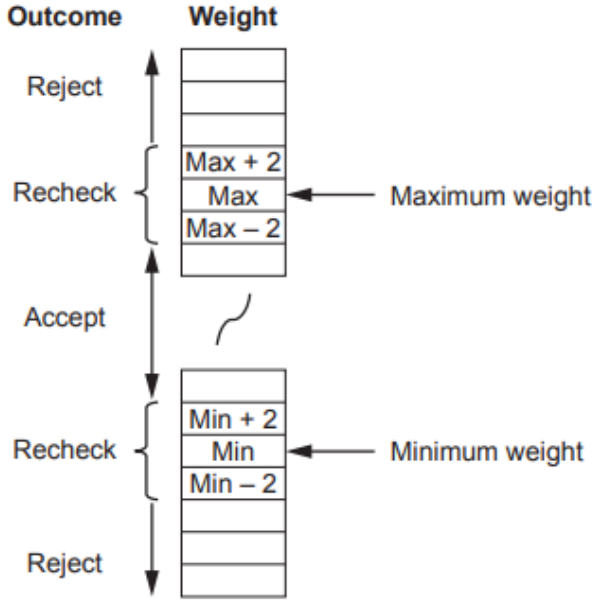
..... [2]

6 Components are weighed during manufacture. Weights are measured to the nearest whole gram.

Components that weigh at least 3 grams more than the maximum weight, or at least 3 grams less than the minimum weight, are rejected.

A component is rechecked if it weighs within 2 grams of either the maximum or minimum weight.

The final outcome of weighing each component is shown below:



A function `Status()` will be called with three parameters. These are integers representing the weight of an individual component together with the minimum and maximum weights.

The value returned from the function will be as follows:

Outcome	Return value
Accept	`'A'`
Reject	`'R'`
Recheck	`'C'`

(b) Write pseudocode for `Status()`.

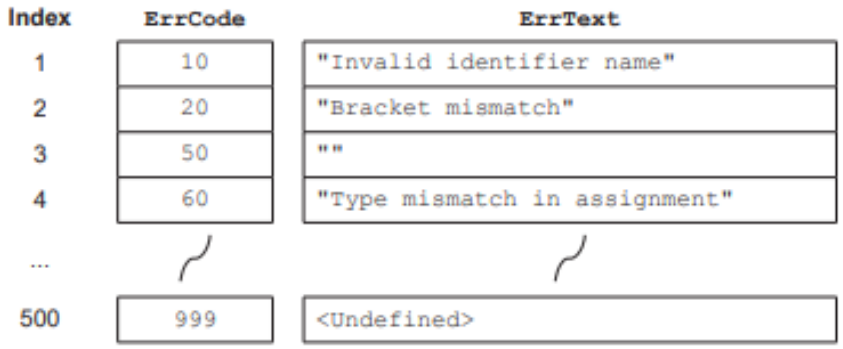
[6]

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.



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

`OutputRange()`	• 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:
	
	`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`

(a) Write pseudocode for module `OutputRange()`. Assume that the two numbers input represent a valid error number range.

s22 qp 21:

3 The manager of a cinema wants a program to allow users to book seats. The cinema has several screens. Each screen shows a different film.

(b) Two types of program modules may be used in the design of the program.

Identify the type of program module that should be used to return a value.

..... [1]

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]

(b) Study the following pseudocode.

```
CASE OF MySwitch
   1: ThisChar ← 'a'
   2: ThisChar ← 'y'
   3: ThisChar ← '7'
   OTHERWISE: ThisChar ← '*'
ENDCASE
```

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

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [4]

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

The following strings are examples of palindromes:

"Racecar"
"madam"
"12344321"

Upper-case and lower-case characters need to be treated the same. For example, 'A' is equivalent to 'a'.

(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()`.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

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
`Secret[27, 1]`	"www.thiswebsite.com"
`Secret[27, 2]`	"••••••••••••••"
`Secret[28, 1]`	"www.thatwebsite.com"
`Secret[28, 2]`	"••••••••••••••"

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

`RandomChar()`	• Generates a single random character from within one of the following ranges:
	  ○ 'a' to 'z'
	  ○ 'A' to 'Z'
	  ○ '0' to '9'
	• Returns the character

`Encrypt()`	• Takes a password as a parameter of type string • Returns the encrypted form of the password as a string
`Decrypt()`	• 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

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

You may wish to refer to the **insert** for a description of the `CHR()` function. Other functions may also be required.

(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]

s22 qp 22:

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

For example:

"12,13,151,22"

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)
```

7 A programming language has string functions equivalent to those given in the **insert**.

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]

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
`Secret[27, 1]`	"thiswebsite.com"
`Secret[27, 2]`	"••••••••••••"
`Secret[28, 1]`	"thatwebsite.com"
`Secret[28, 2]`	"••••••••••••"

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

`Exists()`	• Takes two parameters:
	  ○ a string
	  ○ a character
	• Performs a case-sensitive search for the character in the string
	• Returns `TRUE` if the character occurs in the string, otherwise returns `FALSE`

`Encrypt()`	• Takes a password as a parameter of type string • Returns the encrypted form of the password as a string
`Decrypt()`	• 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'.

(a) Write pseudocode for the module `Exists()`.

[5]

s22 qp 23:

6 **(a)** The factorial of a number is the product of all the integers from 1 to that number.

For example:

factorial of 5 is given by $1 \times 2 \times 3 \times 4 \times 5 = 120$
factorial of 7 is given by $1 \times 2 \times 3 \times 4 \times 5 \times 6 \times 7 = 5040$
factorial of 1 = 1

Note: factorial of 0 = 1

A function `Factorial()` will:

- be called with an integer number as a parameter
- calculate and return the factorial of the number
- return −1 if the number is negative.

Write pseudocode for the function `Factorial()`.

[6]

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

`RandomChar()`	• Generates a single random character from within one of the following ranges:
	  ○ 'a' to 'z'
	  ○ 'A' to 'Z'
	  ○ '0' to '9'
	• Returns the character

`Exists()`	• Takes two parameters:
	  ○ a string
	  ○ a character
	• Performs a case-sensitive search for the character in the string
	• Returns `TRUE` if the character occurs in the string, otherwise returns `FALSE`

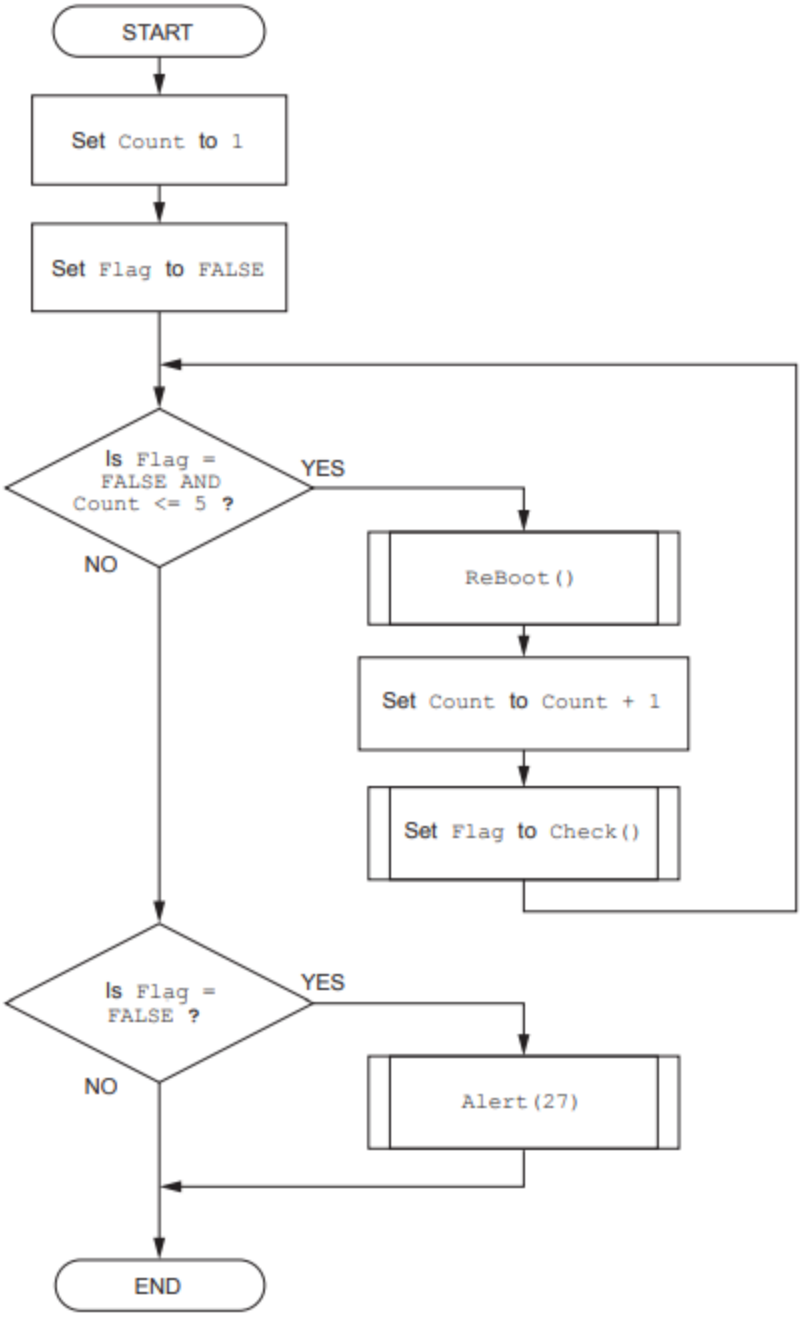
`Generate()`	• Generates a valid password • Uses `RandomChar()` and `Exists()` • Returns the password

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

(a) Write pseudocode for the module `Generate()`.

w21 qp 21:

(b) The program flowchart shown describes a simple algorithm.



Write pseudocode for the simple algorithm shown on page 6.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [6]

w21 qp 22:

4 The following is a procedure design in pseudocode.

Line numbers are given for reference only.

```
10  PROCEDURE Check(InString : STRING)
11     DECLARE Odds, Evens, Index : INTEGER
12
13     Odds ← 0
14     Evens ← 0
15     Index ← 1
16
17     WHILE Index <= LENGTH(InString)
18        IF STR_TO_NUM(MID(InString, Index, 1)) MOD 2 <> 0 THEN
19           Odds ← Odds + 1
20        ELSE
21           Evens ← Evens + 1
22        ENDIF
23        Index ← Index + 1
24     ENDWHILE
25
26     CALL Result(Odds, Evens)
27  ENDPROCEDURE
```

(a) Complete the following table by giving the answers, using the given pseudocode.

	Answer
A line number containing a variable being incremented	
The type of loop structure	
The number of functions used	
The number of parameters passed to `STR_TO_NUM()`	
The name of a procedure other than `Check()`	

[5]

(b) The pseudocode includes several features that make it easier to read and understand.

Identify **three** of these features.

1 .....

2 .....

3 .....

[3]

(c) **(i)** The loop structure used in the pseudocode is not the most appropriate.

State a more appropriate loop structure **and** justify your choice.

Loop structure .....

Justification .....

.....

.....

[2]

(ii) The appropriate loop structure is now used. Two lines of pseudocode are changed and two lines are removed.

Write the line numbers of the two lines that are removed.

.....

..... [1]

s21 qp 21:

2 **(a)** Four program modules form part of a program for a library.

(b) The definition for module `LoanReturn()` is amended as follows:

Module name	Description
`LoanReturn()`	Called with parameters `LoanID`, `BookID` and `Fine` The module code checks whether the book has been returned on time and then assigns a new value to `Fine`

- `LoanID` and `BookID` are of type `STRING`
- `Fine` is of type `REAL`

Write the pseudocode header for the **amended** module `LoanReturn()`.

.....

..... [2]

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
```

(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]

7 A program is needed to take a string containing a full name and 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 abbreviated 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 a global variable `FNString` of type `STRING` to store the full name.

It is assumed that:

- words in the full name string are separated by a single space character
- space characters will not occur at the beginning or the end of the full name string
- the full name string contains at least one word.

The programmer has started to define program modules as follows:

Module	Description
`GetStart()`	• Called with an `INTEGER` as a parameter, representing the number of a word in `FNString`. • Returns the character start position of that word in `FNString` or returns `-1` if that word does not exist • For example: if `FNString` contains the string `"hot and cold"`, `GetStart(3)` returns `9`
`GetWord()`	• Called with a parameter representing the position of the first character of a word in `FNString` • Returns the word from `FNString` • For example: if `FNString` contains the string `"hot and cold"`, `GetWord(9)` returns `"cold"`

(a) Write pseudocode for the module `GetStart()`.

Write pseudocode for the module `GetWord()`.

s21 qp 22:

5 **(a)** A student is learning about arrays.

(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]

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()`.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

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

(b) Write pseudocode for the module `GetInitials()`.

specimen paper:

(c) Write in pseudocode a **post-condition loop** to output all the odd numbers between 100 and 200.

.....

.....

.....

.....

.....

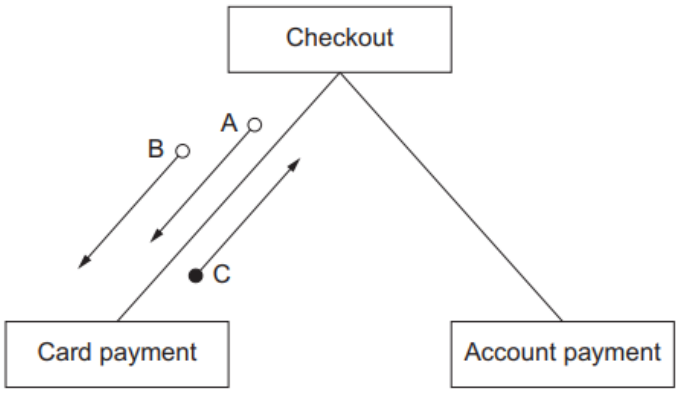
.....

.....

.....

..... [4]

2 Roberta downloads music from an online music store. The diagram shows part of a structure chart for the online music store program.



(b) Examples of the data items that correspond to the arrows are given in the table.

Arrow	Data item
A	`234.56`
B	`"Ms Roberta Smith"`
C	`TRUE`

Use pseudocode to write the function header for the Card payment module.

.....

..... [3]

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]

6 Members of a family use the same laptop computer. Each family member has their own password.

To be valid, a password must comply with the following rules:

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

A function, `ValidatePassword`, is needed to check that a given password follows these rules. This function takes a string, `Pass`, as a parameter and returns a boolean value:

- TRUE if it is a valid password
- FALSE otherwise

(a) Write pseudocode to implement the function `ValidatePassword`.

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