w22 qp 21:

(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 .....

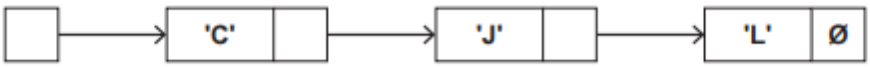
.....

Data type .....

[2]

4 **(a)** The following diagram shows an Abstract Data Type (ADT) representation of an ordered linked list. The data item stored in each node is a single character. The data will be accessed in alphabetical order.

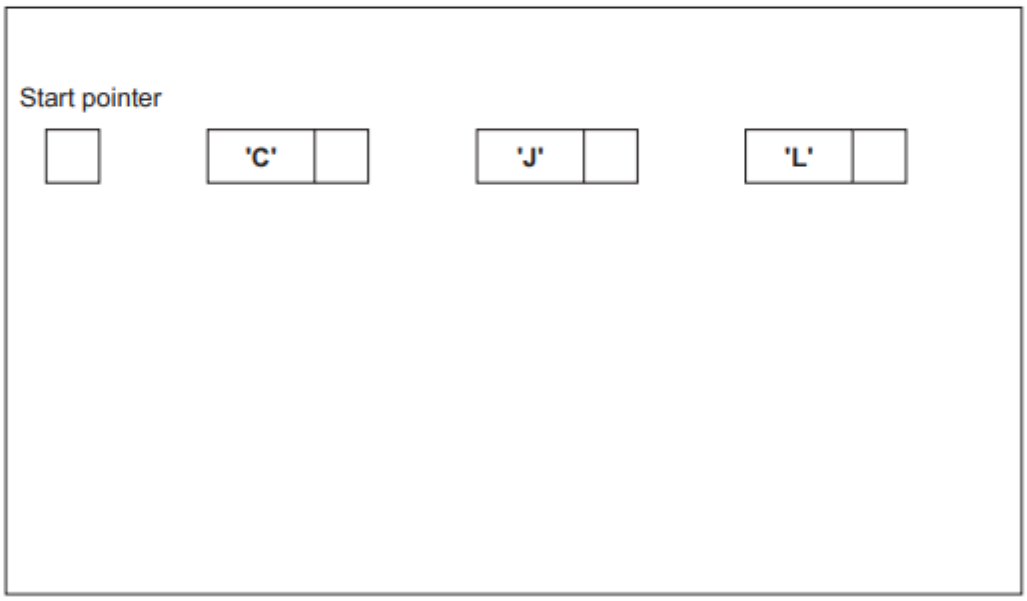
The symbol Ø represents a null pointer.



(i) Nodes with data 'A' and 'K' are added to the linked list. Nodes with data 'J' and 'L' are deleted.

After the changes, the data items still need to be accessed in alphabetical order.

Complete the diagram to show the new state of the linked list.



[4]

(ii) The original data could have been stored in a 1D array in which each element stores a character.

For example:

'C'	'J'	'L'		

Explain the advantages of making the changes described in **part (a)(i)** when the data is stored in the linked list instead of an array.

.....

.....

.....

..... [2]

(iii) Explain the disadvantages of making the changes described in **part (a)(i)** when the data is stored in the linked list instead of an array.

.....

.....

.....

..... [2]

(b) A program will store data using a linked list like the one shown in **part (a)**.

Explain how the linked list can be implemented.

.....

.....

.....

.....

.....

.....

.....

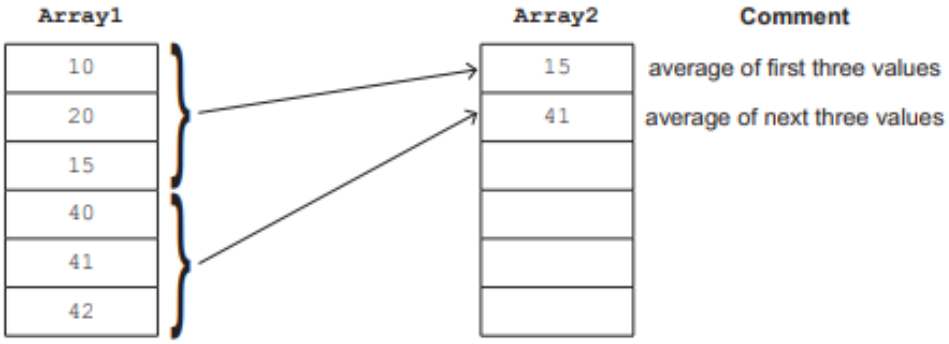
..... [4]

5 A program uses two 1D arrays of type integer. `Array1` contains 600 elements and `Array2` contains 200 elements.

`Array1` contains sample values read from a sensor. The sensor always takes three consecutive samples and all of these values are stored in `Array1`.

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`.

The diagram below illustrates the process for the first six entries in `Array1`.



Write pseudocode for the procedure `Summarise()`.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [5]

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`

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

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

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).

(c) Part of an identifier table is shown:

Variable	Type	Example value
FlagDay	DATE	23/04/2004
CharList	STRING	"ABCDEF"
Count	INTEGER	29

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

Expression	Evaluation
MID(CharList, MONTH(FlagDay), 1)	
INT(Count / LENGTH(CharList))	
(Count >= 29) AND (DAY(FlagDay) > 23)	

[3]

2 **(a)** An algorithm will process data from a test taken by a group of students. The algorithm will prompt and input the name and test mark for each of the 35 students.

The algorithm will add the names of all the students with a test mark of less than 20 to an existing text file `Support_List.txt`, which already contains data from other group tests.

(i) Describe the steps that the algorithm should perform.

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

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [5]

(ii) Explain why it may be better to store the names of the students in a file rather than in an array.

.....

.....

..... [1]

(iii) Explain why `WRITE` mode cannot be used in the answer to **part 2(a)(i)**.

.....

.....

..... [1]

3 A stack is used in a program to store string data which needs to be accessed in several modules.

(a) A stack is an example of an Abstract Data Type (ADT).

Identify **one other** example of an ADT **and** describe its main features.

Example .....

Features .....

.....

.....

.....

.....

.....

[3]

(b) Explain how the stack can be implemented using an array.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [5]

(c) A second stack is used in the program. The diagram below shows the initial state of this stack. Value X is at the top of the stack and was the last item added.

Upper-case letters are used to represent different data values.

Stack operations are performed in three groups as follows:

Group 1:

```
PUSH D
PUSH E
```

Group 2:

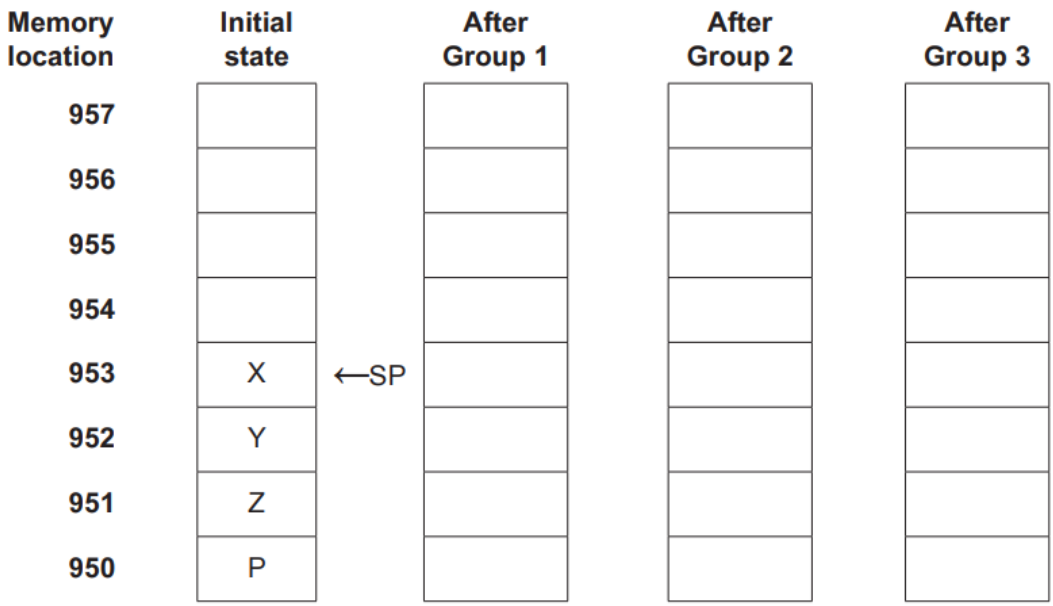
```
POP
POP
POP
```

Group 3:

```
PUSH A
PUSH B
POP
PUSH C
```

Complete the diagram to show the state of the stack **after** each group of operations has been performed.

Include the current stack pointer (SP) **after** each group.


Memory location
Initial state
After Group 1
After Group 2
After Group 3
957
956
955
954
953
952
951
950
X
Y
Z
P
←SP

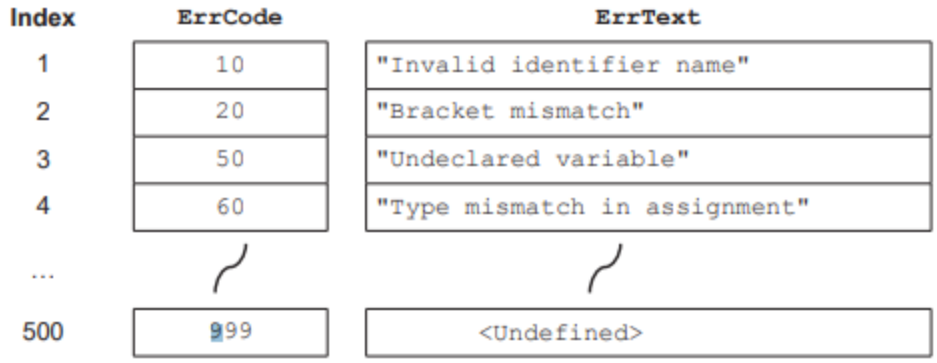
[5]

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 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 `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`.

The teacher has defined two program modules as follows:

Module	Description

`OutputError()`	• takes two parameters as integers:
	  ○ a line number in the student's program
	  ○ an error number
	• searches for the error number in the `ErrCode` 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"`

`SortArrays()`	sorts the arrays into ascending order of `ErrCode`

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

[6]

(b) Write an **efficient** bubble sort algorithm in pseudocode for module `SortArrays()`.

(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]

w22 qp 23:

1 A program is required for a shopping website.

(a) Part of the program requires four variables. The following table describes the use of each variable.

Complete the table by adding the most appropriate data type for each variable.

Variable use	Data type
Store the number of days in the current month	
Store the first letter of the customer's first name	
Store an indication of whether a year is a leap year	
Store the average amount spent per customer visit	

[4]

4 **(a)** A program contains a 1D array `DataItem` with 100 elements.

State the **one additional** piece of information required before the array can be declared.

.....

..... [1]

(b) A programmer decides to implement a queue Abstract Data Type (ADT) in order to store characters received from the keyboard. The queue will need to store at least 10 characters and will be implemented using an array.

(i) Describe **two** operations that are typically required when implementing a queue. State the check that must be carried out before each operation can be completed.

Operation 1 .....

.....

Check 1 .....

.....

Operation 2 .....

.....

Check 2 .....

.....

[4]

(ii) Describe the declaration and initialisation of the variables and data structures used to implement the queue.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [5]

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.

(b) The design changes and a record structure is defined to store the three data items.

A user-defined data type `StockItem` is created as shown:

```
TYPE StockItem
   DECLARE StockID : STRING
   DECLARE Description : STRING
   DECLARE Cost : REAL
ENDTYPE
```

(i) A variable `LineData` of type `StockItem` is declared.

Write the pseudocode statement to assign the value `12.99` to the `Cost` field of `LineData`.

..... [1]

(ii) Procedure `Unpack()` is modified and converted to a function which takes the original text string as the only parameter.

Explain the other changes that need to be made to convert the procedure into a function.

.....

.....

.....

.....

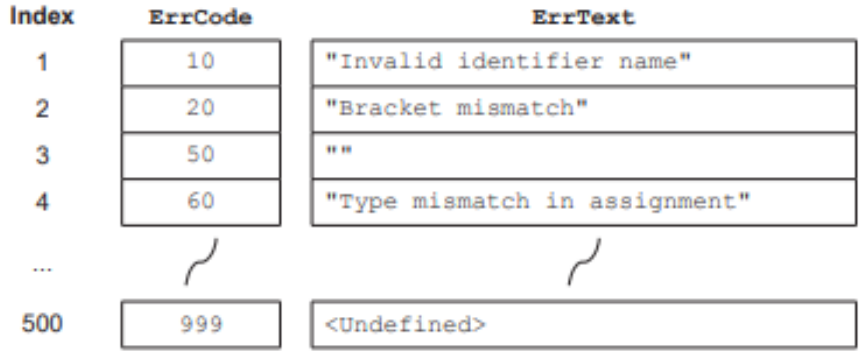
..... [2]

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`

(b) (i) Two additional modules are defined:

Module	Description
`SortArrays()`	• Sorts the arrays into ascending order of `ErrCode`
`AddError()`	• Takes two parameters: ◦ an error number as an integer ◦ an error description as a string • Writes the error number and error description to the first unused element of the two arrays. Ensures the `ErrCode` array is still in ascending order • Returns the number of unused elements **after** the new error number has been added • Returns −1 if the new error number could not be added

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

(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]

s22 qp 21:

1 **(a)** A programmer draws a program flowchart to show the sequence of steps required to solve a problem.

(b) The table lists some of the variables used in a program.

(i) Complete the table by writing the most appropriate data type for each variable.

Variable	Use of variable	Data type
`Temp`	Stores the average temperature	
`PetName`	Stores the name of my pet	
`MyDOB`	To calculate the number of days until my next birthday	
`LightOn`	Stores state of light; light is only on or off	

[4]

(ii) One of the names used for a variable in the table in part **1(b)(i)** is not an example of good practice.

Identify the variable and give a reason why it is **not** good practice to use that name.

Variable .....

Reason .....

.....

.....

[2]

4 A stack is created using a high-level language. Memory locations 200 to 207 are to be used to store the stack.

The following diagram represents the current state of the stack.

`TopOfStack` points to the last value added to the stack.

Stack		Pointer
Memory location	**Value**	
200		
201		
202		
203	'F'	← `TopOfStack`
204	'C'	
205	'D'	
206	'E'	
207	'H'	

(b) The following diagram shows the current state of the stack:

Stack **Pointer**

Memory location	Value	
200		
201		
202	'W'	← TopOfStack
203	'Y'	
204	'X'	
205	'Z'	
206	'N'	
207	'P'	

The following operations are performed:

POP
POP
PUSH 'A'
PUSH 'B'
POP
PUSH 'C'
PUSH 'D'

Complete the diagram to show the state of the stack **after** the operations have been performed.

Stack **Pointer**

Memory location	Value
200	
201	
202	
203	
204	
205	
206	
207	

[4]

5 Each line of a text file contains data organised into fixed-length fields as shown:

<Field 1><Field 2><Field 3>

An algorithm is required to search for the first instance of a given value of Field 2 and, if found, to output the corresponding values for Field 1 and Field 3.

Describe the algorithm needed.

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

[6]

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

(b) A new module is defined as follows:

Module	Description
`FindPassword()`	• Takes a website domain name as a parameter of type string • Searches for the website domain name in the array `Secret` • If the website domain name is found, the decrypted password is returned • If the website domain name is not found, a warning message is output, and an empty string is returned

Write pseudocode for module `FindPassword()`.

Assume that modules `Encrypt()` and `Decrypt()` have already been written.

..... [7]

s22 qp 22:

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()`.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

(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]

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'.

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

- search for the **first** password that occurs more than once in the array and output a message each time a duplicate is found.

  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"
```

- end once all messages have been output.

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.

s22 qp 23:

5 A program will store attendance data about each employee of a company.

The data will be held in a record structure of type `Employee`. The fields that will be needed are as shown:

Field	Typical value	Comment
`EmployeeNumber`	`123`	A numeric value starting from 1
`Name`	`"Smith,Eric"`	Format: <last name>','<first name>
`Department`	`"1B"`	May contain letters and numbers
`Born`	`13/02/2006`	Must not be before 04/02/1957
`Attendance`	`97.40`	Represents a percentage

(a) (i) Write pseudocode to declare the record structure for type `Employee`.

.....

.....

.....

.....

.....

.....

..... [4]

(ii) A 1D array `Staff` containing 500 elements will be used to store the employee records.

Write pseudocode to declare the `Staff` array.

.....

..... [2]

(b) There may be more records in the array than there are employees in the company. In this case, some records of the array will be unused.

(i) State why it is good practice to have a standard way to indicate unused array elements.

.....

..... [1]

(ii) Give **one** way of indicating an unused record in the `Staff` array.

.....

..... [1]

(c) A procedure `Absentees()` will output the `EmployeeNumber` and the `Name` of all employees who have an `Attendance` value of 90.00 or less.

Write pseudocode for the procedure `Absentees()`.

Assume that the `Staff` array is global.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

7 The following pseudocode represents an algorithm intended to output the last three lines as they appear in a text file. Line numbers are provided for reference only.

```
10  PROCEDURE LastLines(ThisFile : STRING)
11     DECLARE ThisLine : STRING
12     DECLARE Buffer : ARRAY[1:3] OF STRING
13     DECLARE LineNum : INTEGER
14     LineNum ← 1
15     OPENFILE ThisFile FOR READ
16
17     WHILE NOT EOF(ThisFile)
18        READFILE Thisfile, ThisLine // read a line
19        Buffer[LineNum] ← ThisLine
20        LineNum ← LineNum + 1
21        IF LineNum = 4 THEN
22           LineNum ← 1
23        ENDIF
24     ENDWHILE
25
26     CLOSEFILE ThisFile
27     FOR LineNum ← 1 TO 3
28        OUTPUT Buffer[LineNum]
29     NEXT LineNum
30  ENDPROCEDURE
```

(a) There is an error in the algorithm. In certain cases, a text file will have at least three lines but the output will be incorrect.

(i) State how the output may be incorrect.

.....

..... [1]

(ii) Describe the error in the algorithm **and** explain how it may be corrected.

Description .....

.....

.....

.....

Explanation .....

.....

.....

.....

[4]

(b) The original algorithm is implemented and sometimes the last three lines of the text file are output correctly.

State the condition that results in the correct output.

.....

..... [1]

(c) Lines 20 to 23 inclusive could be replaced with a single pseudocode statement.

Write the pseudocode statement.

.....

..... [2]

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'.

(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
`Secret[27, 1]`	`"www.thiswebsite.com"`
`Secret[27, 2]`	`"●●●●●●●●●●●●"`
`Secret[28, 1]`	`"www.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.

Additional modules are defined as follows:

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

The first three modules have been written.

Write pseudocode for the module `AddPassword()`.

[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]

w21 qp 21:

3 **(a)** The diagram below represents a queue Abstract Data Type (ADT) that can hold a maximum of eight items.

The operation of this queue may be summarised as follows:

- The front of queue pointer points to the next item to be removed.
- The end of queue pointer points to the last item added.
- The queue is circular so that empty storage elements can be reused.

0	Frog	← Front of queue pointer
1	Cat	
2	Fish	
3	Elk	← End of queue pointer
4		
5		
6		
7		

(i) Describe how "Octopus" is added to the given queue.

.....

.....

.....

..... [2]

(ii) Describe how the next item in the given queue is removed and stored in the variable `AnimalName`.

.....

.....

.....

..... [2]

(iii) Describe the state of the queue when the **front of queue** and the **end of queue** pointers have the same value.

.....

..... [1]

(b) Some operations are carried out on the original queue given in **part (a)**.

(i) The current state of the queue is:

0	Frog
1	Cat
2	Fish
3	Elk
4	
5	
6	
7	

Complete the diagram to show the state of the queue after the following operations:

Add "Wasp", "Bee" and "Mouse", and then remove two data items.

[3]

(ii) The state of the queue after other operations are carried out is shown:

0	Frog	
1	Cat	
2	Fish	
3	Elk	← Front of queue pointer
4	Wasp	
5	Bee	
6	Mouse	← End of queue pointer
7	Ant	

Complete the following diagram to show the state of the queue after the following operations:

Remove one item, and then add "Dolphin" and "Shark".

0	
1	
2	
3	
4	
5	
6	
7	

[2]

(c) The queue is implemented using a 1D array.

Describe the algorithm that should be used to modify the **end of queue pointer** when adding an item to the queue.

Your algorithm should detect any potential error conditions.

.....

.....

.....

.....

.....

..... [3]

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]

(c) A function `LogEvents()` will:

- take a Student ID as a parameter
- for each element in the array that matches the Student ID parameter:
  - add the value of the array element to the existing text file `LogFile`
  - assign an empty string to the array element
- count the number of lines added to the file
- return this count.

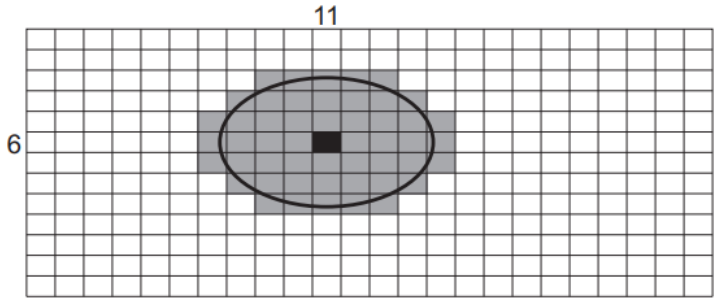
Write pseudocode for the function `LogEvents()`.

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.

Module	Description
`SetRow()` (generates test data)	• Called with three parameters of type `INTEGER`: ◦ 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, `SetRow(3, 8, 5)` will give row 3 as in the diagram shown.
`SearchInRow()`	• Takes two parameters of type `INTEGER`: ◦ 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
`SearchInCol()`	• Takes two parameters of type `INTEGER`: ◦ 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

(a) Write pseudocode to implement the module `SetRow()`.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [5]

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

Module	Description

`SearchInRow()`	• Takes two parameters of type `INTEGER`:
	  ◦ 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()`.

(c) The following new module is introduced:

Module	Description
`GetCentreCol()`	• Called with a row number as an `INTEGER` • Uses `SearchInRow()` 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 −1 if no element is set to 1

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

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....[6]

w21 qp 22:

1 **(a)** A programmer applies decomposition to a problem that she has been asked to solve.

(b) The following pseudocode assigns a value to an element of an array:

```
ThisArray[n] ← 42
```

Complete the following table by writing the answer for each row.

	Answer
The number of dimensions of `ThisArray`	
The technical terms for minimum and maximum values that the variable `n` may take	
The technical term for the variable `n` in the pseudocode statement	

[3]

3 A programmer is writing a program to help manage clubs in a school.

Data will be stored about each student in the school and each student may join up to three clubs.

The data will be held in a record structure of type `Student`.

The programmer has started to define the fields that will be needed as shown in the following table.

Field	Typical value	Comment
`StudentID`	`"CF1234"`	Unique to each student
`Email`	`"Carmen47@xyzmail.com"`	Contains letters, numbers and certain symbols
`Club_1`	`1`	Any value in the range 1 to 99 inclusive
`Club_2`	`14`	Any value in the range 1 to 99 inclusive
`Club_3`	`27`	Any value in the range 1 to 99 inclusive

(a) (i) Write pseudocode to declare the record structure for type `Student`.

.....

.....

.....

.....

.....

.....

..... [3]

(ii) A 1D array `Membership` containing 3000 elements will be used to store the student data.

Write pseudocode to declare the `Membership` array.

.....

..... [2]

(iii) Some of the elements of the array will be unused.

Give an **appropriate** way of indicating an unused array element.

.....

..... [1]

(iv) Some students are members of less than three clubs.

State **one** way of indicating an unused club field.

.....

..... [1]

(b) A procedure `GetIDs()` will:

- prompt and input the number of a club
- output the `StudentID` of all the students who are members of that club
- output a count of all students in the given club.

Write pseudocode for the procedure `GetIDs()`.

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()`.

[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 .....

.....

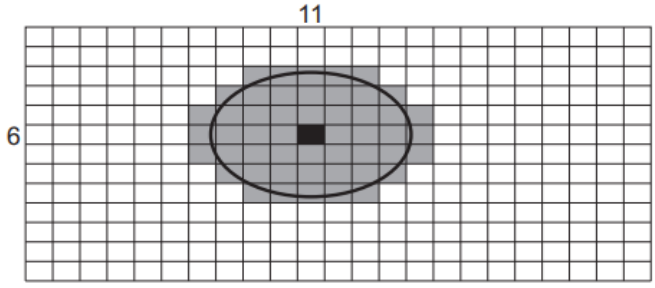
[2]

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 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
`FirstRowSet()`	• 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
`LastRowSet()`	• 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
`FirstColSet()`	• 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
`LastColSet()`	• 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

(a) Write efficient pseudocode for the module `FirstRowSet()`.

[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]

(d) An additional module `GetCentre()` is defined as follows:

Module	Description
`GetCentre()`	• Calculates the coordinates of the centre point • Uses the four modules: `FirstRowSet()`, `LastRowSet()`, `FirstColSet()`, `LastColSet()` • Assigns values to `CentreRow` and `CentreCol` • Sets `CentreRow` to −1 if no screen touch is detected

Write pseudocode for the module `GetCentre()`.

[6]

s21 qp 21:

(c) The following data items need to be stored for each student in a group:

- student name (a string)
- test score (an integer).

State a suitable data structure and justify your answer.

Structure .....

Justification .....

.....

.....

[3]

3 **(a)** A concert venue uses a program to calculate admission prices and store information about ticket sales.

A number of arrays are used to store data. The computer is switched off overnight and data has to be input again at the start of each day before any tickets can be sold. This process is very time consuming.

(i) Explain how the program could use text files to speed up the process.

.....

.....

.....

.....

.....

..... [2]

(ii) State the characteristic of text files that allow them to be used as explained in **part (a)(i)**.

.....

..... [1]

(iii) Information about ticket sales will be stored as a booking. The booking requires the following data:

- name of person booking
- number of people in the group (for example a family ticket or a school party)
- event type.

Suggest how data relating to each booking may be stored in a text file.

.....

.....

.....

..... [2]

(b) A procedure `Preview()` will:

- take the name of a text file as a parameter
- output a warning message if the file is empty
- otherwise output the first five lines from the file (or as many lines as there are in the file if this number is less than five).

Write pseudocode for the procedure `Preview()`.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [5]

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()`.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

6 The following diagram represents an Abstract Data Type (ADT) for a linked list.

The free list is as follows:

(a) Explain how a node containing data value B is added to the list in alphabetic sequence.

.....

.....

.....

.....

.....

.....

..... [4]

(b) Describe how the linked list in **part (a)** may be implemented using variables and arrays.

.....

.....

.....

..... [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"`

(b) The programmer has decided to use a global ten-element 1D array `IgnoreList` of type `STRING` to store the ignored words. Unused elements contain the empty string (`""`) and may occur anywhere in the array.

A new module `AddWord()` is needed as follows:

Module	Description
`AddWord()`	• Called with a parameter representing a word • Stores the word in an unused element of the `IgnoreList` array and returns `TRUE` • Returns `FALSE` if the array was already full or if the word was already in the array

Write a detailed description of the algorithm for `AddWord()`. Do **not** include pseudocode statements in your answer.

.....

.....

.....

.....

.....

.....

.....

.....

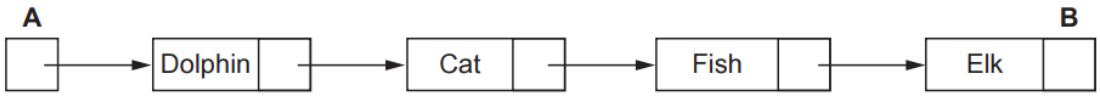
.....

.....

..... [4]

s21 qp 22:

3 The following diagram represents an Abstract Data Type (ADT).



(a) Identify this type of ADT.

..... [1]

(b) Give the technical term for the item labelled **A** in the diagram.

..... [1]

(c) Give the technical term for the item labelled **B** in the diagram.

Explain the meaning of the value given to this item.

Term .....

Meaning .....

.....

..... [2]

(d) Complete the diagram to show the ADT after the data has been sorted in alphabetical order.

[2]

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

She wants to write a program to:

- 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.

(i) Write pseudocode to represent the algorithm.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [6]

(ii) The student decides to modify the algorithm so that each element of the array will contain a unique value.

Describe the changes that the student needs to make to the algorithm.

.....

.....

.....

.....

.....

..... [3]

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

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

[5]

specimen paper:

3 A stack is created using a high-level language. The following diagram represents the current state of the stack. The Top of Stack pointer points to the last item added to the stack.

Address	Value	Pointer
99		
100		
101	E	⇐ TopOfStack
102	D	
103	C	
104	B	
105	A	

(a) Two operations associated with this stack are `PUSH()` and `POP()`.

Describe these operations with reference to the diagram.

```
MyVar = POP()
```

.....

.....

.....

```
PUSH('Z')
```

.....

.....

.....

[4]

(b) Two programs use a stack to exchange data. Program `AddString` pushes a string of characters onto the stack one character at a time. Program `RemoveString` pops the same number of characters off the stack, one character at a time. The string taken off the stack is different from the string put on the stack.

Explain why the strings are different.

.....

.....

.....

..... [2]

5 A company keeps details of its product items in a 1D array, `Stock`. The array consists of 1000 elements of type `StockItem`.

The record fields of `StockItem` are:

Field	Typical value
`ProductCode`	`"BGR24-C"`
`Price`	`102.76`
`NumberInStock`	`15`

(a) Write pseudocode to declare the record structure `StockItem`.

.....

.....

.....

.....

..... [3]

(b) Write pseudocode to declare the `Stock` array.

.....

..... [3]

(c) Write pseudocode to modify the values to element 20 as follows:

- set the price to 105.99
- increase the number in stock by 12

.....

.....

.....

..... [2]

(d) A stock report program is developed.

Write pseudocode to output the information for each stock item that has a price of at least 100.

Output the information as follows:

```
Product Code: BGR24-C Number in Stock: 15
```

.....

.....

.....

.....

.....

.....

.....

..... [4]

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]