AS/A Level past Exams Question 1:

(a) Program variables have values as follows:

Variable	Value
`Today`	`"Tuesday"`
`WeekNumber`	`37`
`Revision`	`'C'`
`MaxWeight`	`60.5`
`LastBatch`	`TRUE`

(i) Give an appropriate data type for each variable.

Variable	Data type
`Today`	
`WeekNumber`	
`Revision`	
`MaxWeight`	
`LastBatch`	

[5]

AS/A Level past Exams Question 1:

(ii) Evaluate each expression in the following table.
If an expression is invalid then write ERROR.

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

Expression	Evaluates to
`MID(Today, 3, 2) & Revision & "ape"`	
`INT(Maxweight + 4.2)`	
`LENGTH(MaxWeight)`	
`MOD(WeekNumber, 12)`	
`(Revision <= 'D') AND (NOT LastBatch)`	

[5]

AS/A Level past Exams Question 1:

(b) Simple algorithms usually consist of input, process and output.

Complete the table to show if each statement is an example of input, process or output. Place one or more ticks (✓) for each statement.

Item	Statement	Input	Process	Output
1	`SomeChars ← "Hello World"`			
2	`OUTPUT RIGHT(SomeChars,5)`			
3	`READFILE MyFile, MyChars`			
4	`WRITEFILE MyFile, "Data is " & MyChars`			

[4]

AS/A Level past Exams Question 1:

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

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [4]

AS/A Level past Exams Answer 1:

<table>
<tr><th>Question</th><th>Answer</th><th>Marks</th></tr>
<tr><td>1(a)(i)</td><td>
<table>
<tr><th>Variable</th><th>Data type</th></tr>
<tr><td>Today</td><td>STRING</td></tr>
<tr><td>WeekNumber</td><td>INTEGER</td></tr>
<tr><td>Revision</td><td>CHAR</td></tr>
<tr><td>MaxWeight</td><td>REAL</td></tr>
<tr><td>LastBatch</td><td>BOOLEAN</td></tr>
</table>
One mark per row
Accept suitable alternatives for REAL
</td><td>5</td></tr>
<tr><td>1(a)(ii)</td><td>
<table>
<tr><th>Expression</th><th>Evaluates to</th></tr>
<tr><td>MID(Today, 3, 2) & Revision & "ape"</td><td>“esCape”</td></tr>
<tr><td>INT(Maxweight + 4.2)</td><td>64</td></tr>
<tr><td>LENGTH(MaxWeight)</td><td>ERROR</td></tr>
<tr><td>MOD(WeekNumber, 12)</td><td>1</td></tr>
<tr><td>(Revision <= 'D') AND (NOT LastBatch)</td><td>FALSE</td></tr>
</table>
One mark per row
Row 1 must have capital ‘C’ and quotes
Rows 2 to 6 must not have quotes
</td><td>5</td></tr>
<tr><td>1(b)</td><td>
<table>
<tr><th>Item</th><th>Statement</th><th>Input</th><th>Process</th><th>Output</th></tr>
<tr><td>1</td><td>SomeChars ← "Hello World"</td><td></td><td>Y</td><td></td></tr>
<tr><td>2</td><td>OUTPUT RIGHT(SomeChars,5)</td><td></td><td>Y</td><td>Y</td></tr>
<tr><td>3</td><td>READFILE MyFile, MyChars</td><td>Y</td><td>(Y)</td><td></td></tr>
<tr><td>4</td><td>WRITEFILE MyFile, "Data is " & MyChars</td><td></td><td>Y</td><td>Y</td></tr>
</table>
One mark per row
</td><td>4</td></tr>
</table>

AS/A Level past Exams Answer 1:

Question	Answer	Marks

1(c)	`MyCount ← 101`
	
	`REPEAT`
	
	`    OUTPUT MyCount`
	
	`    MyCount ← MyCount + 2`
	
	`UNTIL MyCount > 199`
	
	One mark for each of the following:
	Counter initialisation before loop
	`Repeat ... Until` loop
	Method for choosing (correct range of) odd numbers
	Output all odd numbers in the range
		4

AS/A Level past Exams Question 2:

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]

AS/A Level past Exams Question 2:

(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]

AS/A Level past Exams Question 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]

AS/A Level past Exams Answer 2:

Question	Answer	Marks

5(a)	Pseudocode:
	
	`TYPE StockItem`
	`  DECLARE ProductCode : STRING`
	`  DECLARE Price : REAL`
	`  DECLARE NumberInStock : INTEGER`
	`ENDTYPE`
	(allow `END`)
	
	Mark as follows:
	• One mark for `TYPE` and `ENDTYPE`
	• One mark for `Productcode`
	• One mark for `Price` **and** `NumberInStock`
		3

5(b)	`DECLARE Stock : ARRAY [1:1000] OF StockItem` One mark per underlined phrase	3

5(c)	`Stock[20].Price ← 105.99`
	`Stock[20].NumberInStock ← Stock[20].NumberInStock + 12`
	
	One mark per statement
		2

AS/A Level past Exams Answer 2:

Question	Answer	Marks

5(d)	Pseudocode:
	
	`DECLARE n : INTEGER`
	`FOR n ← 1 to 1000`
	`  IF Stock[n].Price >= 100`
	`  THEN`
	`  OUTPUT "ProductCode: " & Stock[n].ProductCode _`
	`  " Number in Stock: " & Stock[n].NumberInStock`
	`  ENDIF`
	`NEXT`
	
	One mark for each of:
	• Loop through all elements of the array
	• Check Price > 99.99
	• `OUTPUT` of 2 fields ...
	• ... with suitable supporting text text
	
	(Or could ask for tabular form with column headers)
		4

AS/A Level past Exams Question 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]

AS/A Level past Exams Question 3:

(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]

AS/A Level past Exams Answer 3:

Question	Answer	Marks
3(a)	POP(): • The value 'E' is removed from the stack (and assigned to variable MyVar) • Top of Stack pointer is incremented to 102 PUSH(): • Top of Stack pointer is decremented to 101 • 'Z' is loaded into address 101 Allow follow through for PUSH()	4
3(b)	• The received string will be reversed • because the stack operates as a FILO structure	2

AS/A Level past Exams Question 4:

`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

.....

.....

.....

.....

.....

.....

.....

.....

.....

AS/A Level past Exams Answer 4:

Question	Answer	Marks

7	Pseudocode :
	
	`PROCEDURE LogEvents()`
	
	`DECLARE FileData : STRING`
	`DECLARE ArrayIndex : INTEGER`
	`OPENFILE "LoginFile.txt" FOR APPEND`
	`FOR ArrayIndex ← 1 TO 500 // 0 TO 499`
	`IF LogArray[ArrayIndex]<> "Empty"`
	`THEN`
	`FileData ← LogArray[ArrayIndex]`
	`WRITEFILE "LoginFile.txt", FileData`
	`ENDIF`
	`NEXT`
	
	`CLOSEFILE "LoginFile.txt"`
	
	`ENDPROCEDURE`
	
	1 mark for each of the following:
	1 Procedure heading and ending (ignore any input parameters but don't allow a return value)
	2 Declare `ArrayIndex` (any name) as integer
	3 Open file `LoginFile` for append
	4 Correct loop
	5 Extract data from array **in a loop**
	6 Check for unused element **in a loop**
	7 Write data to file **in a loop**
	8 Close the file **outside the loop**
	
	Allow single write to file **outside** loop if complete string built **within** loop
		8

AS/A Level past Exams Question 5:

(b) Programming languages support different data types. These usually include `STRING` and `REAL`.

Complete the table by giving **four other** data types **and** an example data value for each.

Data type	Example data value

[4]

AS/A Level past Exams Answer 5:

1(b)	Many acceptable answers, must be four different data types together with appropriate values One mark per row For example: Note: STRING and REAL are excluded as these are given in the question.	4

Data type	Example data value
BOOLEAN	FALSE
CHAR	'!'
DATE	01/01/01
INTEGER	27

AS/A Level past Exams Question 6:

(a) The following pseudocode includes a procedure that searches for a value in a 1D array and outputs each position in the array where the value is found.

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

```
DECLARE NameList : ARRAY [1:100] OF STRING
DECLARE SearchString : STRING

PROCEDURE Search()
DECLARE Index : INTEGER

FOR Index ← 1 TO 100
    IF NameList[Index] = SearchString
        THEN
            OUTPUT "Found at " & NUM_TO_STRING(Index)
    ENDIF
ENDFOR
ENDPROCEDURE
```

The specification of module `Search()` changes. The pseudocode needs to be amended to meet a new requirement.

The procedure needs to be implemented as a function, `Search()`, which will:

- take the search value as a parameter
- return an integer which is:
    - either the index value where the search value is **first** found
    - or `-1` if the search value is **not** found.

Write the **pseudocode** for the function `Search()`.

.....

.....

AS/A Level past Exams Answer 6:

Question	Answer	Marks

4(a)	`FUNCTION Search(SearchString : STRING) RETURNS INTEGER`
	
	`DECLARE RetVal : INTEGER`
	`DECLARE Index : INTEGER`
	
	`RetVal ← -1`
	`Index ← 1`
	
	`WHILE Index <= 100 AND RetVal = -1`
	`IF NameList[Index] = SearchString`
	`THEN`
	`RetVal ← Index`
	`ENDIF`
	`Index ← Index + 1`
	`ENDWHILE`
	
	`RETURN RetVal`
	
	`ENDFUNCTION`
	
	Mark as follows:
	1 Function heading and ending including parameter
	2 Declaration of integer for `Index`
	3 Initialisation **and** increment of `Index` (implied in `FOR` loop)
	4 Conditional loop // `FOR` loop with immediate `RETURN` if `SearchString` found
	5 Comparison of array element with `SearchString` **AND** assigning **just the first** occurrence to `RetVal` OR setting the termination condition
	6 Return `RetVal` (correctly in both cases)
		6

AS/A Level past Exams Question 7:

A hashtag is used on a social media network to make it easier to find messages with a specific theme or content. A hashtag is a string consisting of a hash character '#' followed by a number of alphanumeric characters.

A message may contain several hashtag strings. A hashtag may be terminated by a space character, the start of the next hashtag, or by the end of the message.

For example, the following message contains three hashtags:

```
"#Alarm34 is the result of #BatteryFailure in the #PowerModule"
```

The hashtags in this message are `"#Alarm34"`, `"#BatteryFailure"` and `"#PowerModule"`.

A program is being developed to monitor their use.

The program will include two global arrays each containing 10 000 elements:

- A 1D array, `TagString`, of type `STRING` storing each hashtag in a single element of the array. All unused array elements contain an empty string (`""`).
- A 1D array, `TagCount`, of type `INTEGER` storing a count of the number of times each hashtag is used. The count value in a given element relates to the hashtag value stored in the element in the `TagString` array with the corresponding index value.

AS/A Level past Exams Question 7:

A developer has started to define the modules. Module `GetStart()` has already been written.

Module	Description

`GetStart()`	• Called with two parameters:
	  ◦ a message of type `STRING`
	  ◦ an integer giving the number of the required hashtag; for example, `GetStart(Message, 3)` would search for the third hashtag in the string `Message`
	• Returns an integer value representing the start position of the hashtag in the message, or value −1 if that hashtag does not exist

`AddHashtag()`	• Called with a hashtag of type `STRING` • Copies the hashtag to the next free element of the `TagString` array, and sets the corresponding element of the `TagCount` array to 1 • Returns `FALSE` if there are no unused elements in the `TagString` array, otherwise returns `TRUE`
`CountHashtag()`	• Called with a message of type `STRING` • Searches the message for hashtags using `GetStart()` • Returns a value representing the number of hashtags in the message
`IncrementHashtag()`	• Called with a hashtag of type `STRING` • Increments the value of the appropriate element in the `TagCount` array if the hashtag is found • Returns `TRUE` if the hashtag is found, or `FALSE` if the hashtag is not found

AS/A Level past Exams Question 7:

(a) Write **pseudocode** for the module `AddHashtag()`.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

AS/A Level past Exams Answer 7:

PUBLISHED

Question	Answer	Marks

5(a)	`FUNCTION AddHashtag (HashTag : STRING) RETURNS BOOLEAN`
	`  DECLARE Index : INTEGER`
	`  DECLARE Added : BOOLEAN`
	`  CONSTANT EMPTY = ""`
	
	`  Added ← FALSE`
	`  Index ← 1 // first element`
	
	`  REPEAT`
	`  IF TagString[Index] = EMPTY`
	`  THEN`
	`  TagString[Index} ← HashTag`
	`  TagCount[Index] ← 1`
	`  Added ← TRUE`
	`  ELSE`
	`  Index ← Index + 1`
	`  ENDIF`
	
	`  UNTIL Index > 10000 OR Added = TRUE`
	
	`  RETURN Added`
	`ENDFUNCTION`
	
	1 mark for each of the following:
	
	1 Declaration of two local variables: Integer for `index` & Boolean for return value (unless immediate Return used)
	2 Conditional loop through all elements until empty element found OR end of array
	3 Test if `TagString` element is empty in a loop
	4 If so then assign `HashTag` to `TagString[]` and 1 to `TagCount[]`
	5 Set loop termination
	6 Return Boolean (for both cases)
		6