Cambridge Assessment International Education

Cambridge International AS & A Level

CANDIDATE NAME									
CENTRE NUMBER						CANDIDATE NUMBER				

COMPUTER SCIENCE **9608/23**

Paper 2 Fundamental Problem-solving and Programming Skills **October/November 2020**

2 hours

You must answer on the question paper.

No additional materials are needed.

INSTRUCTIONS

- Answer **all** questions.
- Use a black or dark blue pen.
- Write your name, centre number and candidate number in the boxes at the top of the page.
- Write your answer to each question in the space provided.
- Do **not** use an erasable pen or correction fluid.
- Do **not** write on any bar codes.
- You may use an HB pencil for any diagrams, graphs or rough working.
- Calculators must **not** be used in this paper.

INFORMATION

- The total mark for this paper is 75.
- The number of marks for each question or part question is shown in brackets [ ].
- No marks will be awarded for using brand names of software packages or hardware.

This document has **16** pages. Blank pages are indicated.

1 **(a)** A programmer uses the process of stepwise refinement to break down a problem.

Explain the purpose of stepwise refinement.

.....

.....

.....

..... [2]

(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]

(c) An experienced programmer is working on a program that is written in a language she is not familiar with.

(i) State **one** feature of the program that she should be able to recognise.

.....

..... [1]

(ii) State the type of skill that would allow her to recognise this feature.

.....

..... [1]

(d) Give **three** methods that may be used to identify and locate errors in a program **after it has been written**.

You may include **one** feature found in a typical Integrated Development Environment (IDE).

1 .....

2 .....

3 .....

[3]

2 **(a)** An algorithm is needed to input a list of numbers representing test marks for a class of 30 students.
The algorithm will output the number of students who have a mark greater than 75. It will also output the average mark for the class.

Document the algorithm using **structured English**.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [8]

(b) Each pseudocode statement in the following table contains an error.

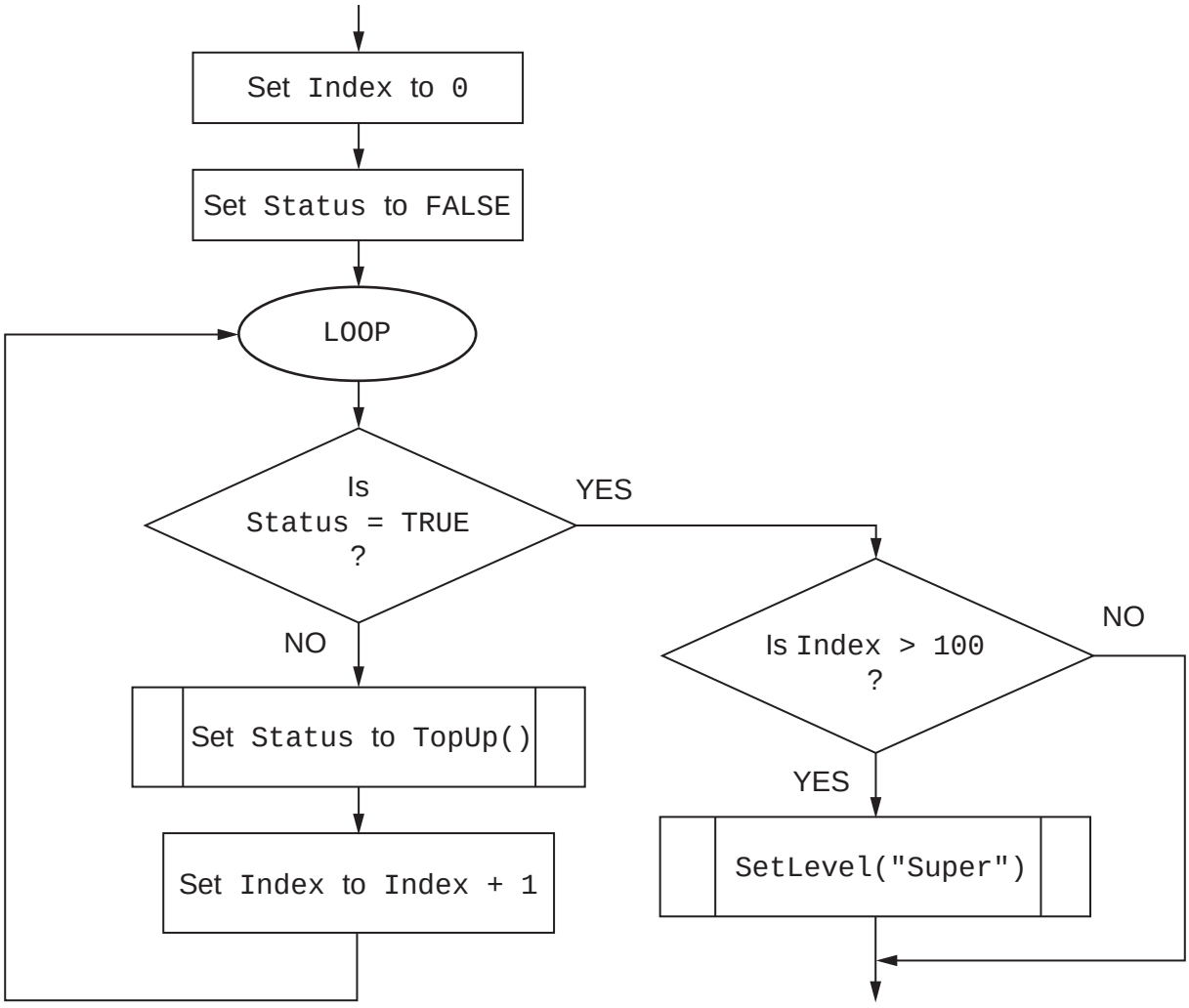
State the error in each case.

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

Statement	Error
`Code ← LEFT(3, "Europe")`	
`Hour ← MID("ALARM:12:02", 7, 6)`	
`Size ← LENGTH(27.5)`	
`Num ← INT(27 / (Count + 3)`	
`Result ← "Conditional" AND "Loop"`	

[5]

(c) Part of a program flowchart is shown.



Write **program code** to implement the flowchart shown. Variable declarations are not required.

Programming language .....

Program code

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [6]

3 A global 1D array, `ProdNum`, of type `INTEGER` contains `5000` elements and is used to store product numbers.

A procedure is needed to sort `ProdNum` into ascending order using a bubble sort algorithm.

Write **program code** for the procedure `BubbleSort()`.

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

Programming language .....

Program code

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [7]

4 **(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()`.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [6]

(b) A change to the specification in **part (a)** required a modification of the algorithm.

Give the term used for this type of modification.

..... [1]

(c) A change to the specification is only one reason to modify an algorithm.

Give **another** reason for the modification of an algorithm.

..... [1]

(d) Consider the following pseudocode:

```
10 DECLARE VarA : INTEGER
11 VarA ← 20
12
13 CALL ProcA(VarA)
14 OUTPUT VarA       // first value output
15
16 CALL ProcB(VarA)
17 OUTPUT VarA       // second value output
18
19
20 PROCEDURE ProcA(BYVALUE ThisValue : INTEGER)
21    ThisValue ← ThisValue + 5
22 ENDPROCEDURE
23
24 PROCEDURE ProcB(BYREF ThisValue : INTEGER)
25    ThisValue ← ThisValue + 5
26 ENDPROCEDURE
```

Procedures `ProcA()` and `ProcB()` use two methods of passing parameters.

Complete the following table.

	Output	**Explanation**
First value (line 14)	.....	....
Second value (line 17)	.....	....

[4]

(e) The procedures `ProcA` and `ProcB` in **part (d)** are examples of program modules.

Give **two** advantages of using program modules in program design.

1 .....

.....

2 .....

.....

[2]

5 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.

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

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

.... [6]

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

The module description is repeated here for reference.

Module	Description
`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

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

Programming language .....

Program code

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [6]

(c) Write **program code** for the module `IncrementHashtag()`.

The module description is repeated here for reference.

Module	Description
`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

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

Programming language .....

Program code

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [4]

(d) A procedure, `OutputMostPop()`, is needed to output the most popular hashtag. The most popular hashtag is the one with the highest count value stored in the `TagCount` array.

As a reminder, the program includes 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.

If the maximum count value occurs once, the procedure will output the corresponding hashtag and the count value.

It is possible for more than one hashtag to have the same highest count value. In this case, the procedure will output the maximum count value together with the number of hashtags with this maximum count value.

In both cases, the procedure must also output a suitable message.

You can assume that the arrays contain data for at least one hashtag.

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

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

[8]

Appendix

Built-in functions (pseudocode)

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

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

Example: `LENGTH("Happy Days")` returns `10`

`LEFT(ThisString : STRING, x : INTEGER) RETURNS STRING`
returns leftmost `x` characters from `ThisString`

Example: `LEFT("ABCDEFGH", 3)` returns string `"ABC"`

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

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

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

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

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

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

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

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

Operators (pseudocode)

Operator	Description
`&`	Concatenates (joins) two strings Example: `"Summer" & " " & "Pudding"` produces `"Summer Pudding"`
`AND`	Performs a logical `AND` on two Boolean values Example: `TRUE AND FALSE` produces `FALSE`
`OR`	Performs a logical `OR` on two Boolean values Example: `TRUE OR FALSE` produces `TRUE`