Cambridge International AS & A Level

Cambridge Assessment International Education
Cambridge International Advanced Subsidiary and Advanced Level

CANDIDATE NAME

CENTRE NUMBER | | | | | |

CANDIDATE NUMBER | | | | |

2543357708

COMPUTER SCIENCE **9608/22**

Paper 2 Fundamental Problem-solving and Programming Skills **May/June 2019**

2 hours

Candidates answer on the Question Paper.

No Additional Materials are required.

No calculators allowed.

READ THESE INSTRUCTIONS FIRST

Write your centre number, candidate number and name in the spaces at the top of this page.
Write in dark blue or black pen.
You may use an HB pencil for any diagrams, graphs or rough working.
Do not use staples, paper clips, glue or correction fluid.
DO **NOT** WRITE IN ANY BARCODES.

Answer **all** questions.
No marks will be awarded for using brand names of software packages or hardware.

At the end of the examination, fasten all your work securely together.
The number of marks is given in brackets [ ] at the end of each question or part question.

The maximum number of marks is 75.

This document consists of **16** printed pages.

© UCLES 2019

Cambridge Assessment International Education

[Turn over

1 **(a)** Algorithms are used in computer programming.

(i) Explain the term **algorithm**.

.....

.....

.....

..... [2]

(ii) Algorithms usually consist of three different types of activity.

Complete the table below.

The third activity type is given.

Activity type	Pseudocode example
.....	....
.....	....
Output	....

[5]

(b) Program variables have values as follows:

Variable	Value
Married	03/04/1982
ID	"M1234"
MiddleInitial	'J'
Height	5.6
IsMarried	TRUE
Children	2

(i) Evaluate each expression in the following table.

If an expression is invalid, write ERROR.

For the built-in functions list, refer to the **Appendix** on page 16.

Expression	Evaluates to
STRING_TO_NUM(RIGHT(ID, 3))	
INT(Height * Children)	
IsMarried AND Married < 31/12/1999	
LENGTH(ID & NUM_TO_STRING(Height))	
MID(ID, INT(Height) – Children, 2)	

[5]

(ii) Programming languages support different data types.

Give an appropriate data type for the following variables from **part (b)**.

Variable	Data type
Married	
ID	
MiddleInitial	
Height	
IsMarried	

[5]

2 (a) (i) Procedures and functions are examples of subroutines.

State a reason for using subroutines in the construction of an algorithm.

.....

..... [1]

(ii) Give **three** advantages of using subroutines in a program.

1 .....

.....

2 .....

.....

3 .....

.....

[3]

(iii) The following pseudocode uses the subroutine `DoSomething()`.

```
Answer ← 23 + DoSomething("Yellow")
```

State whether the subroutine is a function or a procedure. Justify your answer.

Type of subroutine .....

Justification .....

.....

[2]

(b) The program development cycle involves writing, translating and testing a high-level language program.

Describe these activities with reference to **each** of the following:

- editor
- translator
- debugger

.....

.....

.....

.....

.....

..... [3]

© UCLES 2019

(c) The following lines of code are taken from a high-level language program.

```
WHEN Result < 20
  {
   Call ResetSensor(3)
   Result := GetSensor(3)
  }
```

Identify the type of control structure **and** describe the function of the code.

Control structure .....

Function of code .....

.....

.....

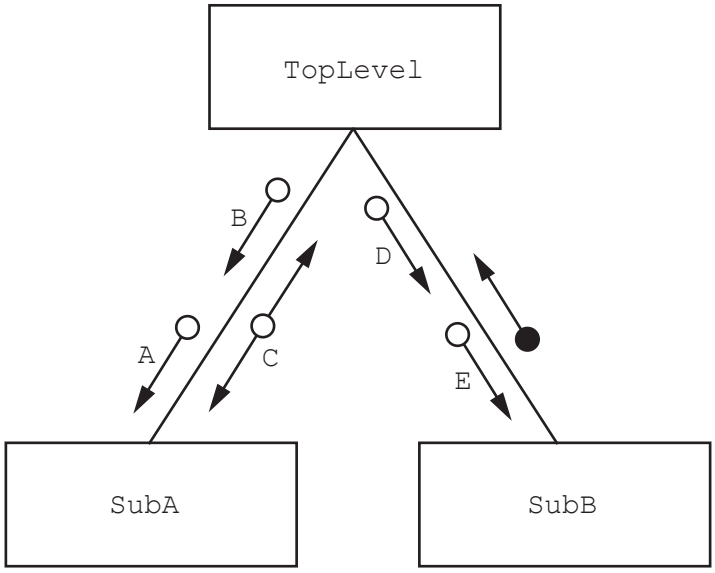
.....

.....

.....

[3]

3 The following structure chart shows the relationship between three modules.



Parameters `A` to `E` have the following data types:

```
A, D : STRING
C    : CHAR
B, E : INTEGER
```

(a) (i) Write the **pseudocode** header for module `SubA()`.

.....

.....

..... [3]

(ii) Write the **pseudocode** header for module `SubB()`.

.....

.....

..... [3]

(b) Module hierarchy and parameters are two features that may be represented on a structure chart.

State **two other** features than can be represented.

Feature 1 .....

Feature 2 .....

[2]

© UCLES 2019

Question 4 begins on the next page.

4 The following is pseudocode for a string handling function.

For the built-in functions list, refer to the **Appendix** on page 16.

```
FUNCTION Search(InString : STRING) RETURNS INTEGER

   DECLARE NewString : STRING
   DECLARE Index : INTEGER
   DECLARE NextChar : CHAR
   DECLARE Selected : INTEGER
   DECLARE NewValue : INTEGER

   NewString ← '0'
   Selected ← 0

   FOR Index ← 1 TO LENGTH(InString)

      NextChar ← MID(InString, Index, 1)
      IF NextChar < '0' OR NextChar > '9'
         THEN
            NewValue ← STRING_TO_NUM(NewString)
            IF NewValue > Selected
               THEN
                  Selected ← NewValue
            ENDIF
            NewString ← '0'
         ELSE
            NewString ← NewString & NextChar
      ENDIF

   ENDFOR

   RETURN Selected

ENDFUNCTION
```

© UCLES 2019

(a) (i) The following assignment calls the `Search()` function:

```
Result ← Search("12∇34∇5∇∇39")
```

Complete the following trace table by performing a dry run of this function call.

The symbol `'∇'` represents a space character. Use this symbol to represent a space character in the trace table.

Index	NextChar	Selected	NewValue	NewString

[5]

(ii) State the value returned by the function when it is called as shown in **part (a)(i)**.

..... [1]

(b) There is an error in the algorithm. When called as shown in **part (a)(i)**, the function did not return the largest value as expected.

(i) Explain why this error occurred when the program called the function.

.....

.....

.....

..... [2]

(ii) Describe how the algorithm could be amended to correct the error.

.....

.....

.....

..... [2]

5 A student is learning about text files. She wants to write a program to count the number of lines in a file.

(a) Use **structured English** to describe an algorithm she could use.

.....

.....

.....

.....

.....

..... [3]

© UCLES 2019

(b) A procedure, `CountLines()`, is being written to count the number of lines in a text file. The procedure will:

- take a filename as a string parameter
- count the number of lines in the file
- output a single suitable message that includes the total number of lines.

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

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [6]

6 Nadine is developing a program to store the ID and preferred name for each student in a school. For example, student Pradeep uses the preferred name “Prad”.

The program will:

1. prompt and input a valid user ID and a preferred name
2. write the user ID and preferred name to one of two files
3. allow the user to end the program or repeat from step 1.

The program will consist of three separate modules. Each module will be implemented using either a procedure or a function.

Nadine has defined the modules as follows:

Module	Description
`TopLevel()`	• Call `GetInfo()` to obtain a string containing a valid user ID and a preferred name • Call `WriteInfo()` to write the string to either `File1.txt` or `File2.txt` depending on the first character of the user ID as follows: ○ ‘A’ to ‘M’: writes to `File1.txt` ○ ‘N’ to ‘Z’: writes to `File2.txt` For example, a string with a user ID of `"G1234"` writes to `File1.txt` • End the program if the file write was unsuccessful • Input (Y/N) to either repeat for the next user ID or to end the program
`GetInfo()`	• Input a user ID and repeat until the user ID is valid • Input a preferred name. This will be an empty string if no preferred name is input. • Concatenate the user ID and preferred name using a `'*'` character as a separator and return this string
`WriteInfo()`	• Open the file • Append the concatenated string to the file • Close the file • Return a Boolean value: ○ `TRUE` if the file write was successful ○ `FALSE` if the file write failed, for example, if the disk was full

A valid user ID:

- is five characters in length
- has a single upper case alphabetic character followed by four numeric characters, for example “G1234”.

Nadine has decided that global variables and nested modules must not be used.

Nadine wants all inputs to have suitable prompts.

© UCLES 2019

(a) Write **program code** for the module `GetInfo()`.

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

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

.... [8]

© UCLES 2019

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

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

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [8]

(c) Write **pseudocode** for the module declaration of `WriteInfo()`.

.....

.....

.....

..... [3]

Permission to reproduce items where third-party owned material protected by copyright is included has been sought and cleared where possible. Every reasonable effort has been made by the publisher (UCLES) to trace copyright holders, but if any items requiring clearance have unwittingly been included, the publisher will be pleased to make amends at the earliest possible opportunity.

To avoid the issue of disclosure of answer-related information to candidates, all copyright acknowledgements are reproduced online in the Cambridge Assessment International Education Copyright Acknowledgements Booklet. This is produced for each series of examinations and is freely available to download at www.cambridgeinternational.org after the live examination series.

Cambridge Assessment International Education is part of the Cambridge Assessment Group. Cambridge Assessment is the brand name of the University of Cambridge Local Examinations Syndicate (UCLES), which itself is a department of the University of Cambridge.

© UCLES 2019

Appendix

Built-in functions (pseudocode)

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

`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 `"BCD"`

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

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

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

`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.

Example: `NUM_TO_STRING(87.5)` returns `"87.5"`
Note: This function will also work if `x` is of type `INTEGER`

`STRING_TO_NUM(x : STRING) RETURNS REAL`
returns a numeric representation of a string.

Example: `STRING_TO_NUM("23.45")` returns `23.45`
Note: This function will also work if `x` is of type `CHAR`

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`

© UCLES 2019