Cambridge International AS & A Level

Cambridge Assessment International Education
Cambridge International Advanced Subsidiary and Advanced Level

CANDIDATE NAME

CENTRE NUMBER | | | | | |

CANDIDATE NUMBER | | | | |

3025588347

COMPUTER SCIENCE **9608/21**

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.

DC (ST) 163545/2

1 (a) (i) Algorithms may be expressed using four basic constructs. One construct is sequence.

Complete the following table for two other constructs.

Construct	Pseudocode example
.....	....
.....	....

[4]

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

Complete the table by placing ticks ('✓') in the relevant boxes.

Pseudocode statement	Input	Process	Output
`Temp ← SensorValue * Factor`			
`WRITEFILE "LogFile.txt", TextLine`			
`WRITEFILE "LogFile.txt", MyName & MyIDNumber`			
`READFILE "AddressBook.txt", NextLine`			

[4]

(b) Program variables have values as follows:

Variable	Value
Title	"101 tricks with spaghetti"
Version	'C'
Author	"Eric Peapod"
PackSize	4
WeightEach	6.2
Paperback	TRUE

(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
MID(Title, 5, 3) & RIGHT(Author, 3)	
INT(WeightEach * PackSize)	
PackSize >= 4 AND WeightEach < 6.2	
LEFT(Author, ASC(Version) - 65)	
RIGHT(Title, (LENGTH(Author) - 6))	

[5]

(ii) Programming languages support different data types.

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

Variable	Data type
Title	
Version	
PackSize	
WeightEach	
Paperback	

[5]

(c) White-box and black-box are two types of testing. In white-box testing, data are chosen to test every possible path through the program.

Explain how data are chosen in black-box testing.

.....

..... [2]

[Turn over

2 **(a)** One type of loop that may be found in an algorithm is a count-controlled loop.

State **one other** type **and** explain when it should be used.

Type .....

Explanation .....

.....

.....

[2]

(b) Chris is asked to work on a program that has been coded in a language he is not familiar with.

He has identified that the program contains the constructs: sequence, iteration and selection.

Identify **three other** features of the program that he should expect to recognise.

Feature 1 .....

Feature 2 .....

Feature 3 .....

[3]

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

```
ON x {
      15: Call ProcA
      20: y := 0
      25: y := 99
      NONE: Call ProcError
     }
```

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

Control structure .....

Description .....

.....

.....

.....

[3]

3 **(a)** A student is developing an algorithm to search through a 1D array of 100 elements. Each element of the array, `Result`, contains a `REAL` value.

The algorithm will output:

- the average value of all the elements
- the number of elements with a value of zero.

The structured English description of the algorithm is:

1. SET Total value to 0
2. SET Zero count to 0
3. SELECT the first element
4. ADD value of element to Total value
5. IF element value is 0 then INCREMENT Zero count
6. REPEAT from step 4 for next element, until element is last element
7. SET Average to Total / 100
8. OUTPUT a suitable message and Average
9. OUTPUT a suitable message and Zero count

Write **pseudocode** for this algorithm.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [7]

(b) The student decides to change the algorithm and implement it as a procedure, `ScanArray()`, which will be called with three parameters.

```
ScanArray(AverageValue, ZeroCount, ArrayName)
```

`ScanArray()` will modify the first two parameters so that the new values are available to the calling program or module.

Write the **pseudocode** procedure header for `ScanArray()`.

.....

.....

.....

..... [4]

Question 4 begins on the next page.

4 The following pseudocode is a string handling function.

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

```
FUNCTION Clean(InString : STRING) RETURNS STRING

   DECLARE NewString : STRING
   DECLARE Index : INTEGER
   DECLARE AfterSpace : BOOLEAN
   DECLARE NextChar : CHAR
   CONSTANT Space = ' '

   AfterSpace ← FALSE
   NewString ← ""

   FOR Index ← 1 TO LENGTH(InString)
      NextChar ← MID(InString, Index, 1)
      IF AfterSpace = TRUE
         THEN
            IF NextChar <> Space
               THEN
                  NewString ← NewString & NextChar
                  AfterSpace ← FALSE
            ENDIF
         ELSE
            NewString ← NewString & NextChar
            IF NextChar = Space
               THEN
                  AfterSpace ← TRUE
            ENDIF
      ENDIF
   ENDFOR

   RETURN NewString

ENDFUNCTION
```

(a) (i) Complete the trace table by performing a dry run of the function when it is called as follows:

```
Result ← Clean("X∇∇∇Y∇and∇∇Z")
```

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

Index	AfterSpace	NextChar	NewString

[6]

(ii) State the effect of the function `Clean()`.

.....

..... [1]

(iii) The pseudocode is changed so that the variable `AfterSpace` is initialised to `TRUE`.

Explain what will happen if the function is called as follows:

```
Result ← Clean("∇∇X∇∇∇Y∇and∇∇Z")
```

.....

.....

.....

..... [2]

(b) The following pseudocode declares and initialises an array.

```
DECLARE Code : ARRAY[1:100] OF STRING
DECLARE Index : INTEGER

FOR Index ← 1 TO 100
   Code[Index] ← ""
ENDFOR
```

The design of the program is changed as follows:

- the array needs to be two dimensional, with 500 rows and 4 columns
- the elements of the array need to be initialised to the string `"Empty"`

Re-write the **pseudocode** to implement the new design.

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [4]

(c) State the term used for changes that are made to a program in response to a specification change.

..... [1]

Question 5 begins on the next page.

5 **(a)** Programming languages usually contain a range of built-in functions, such as a random number generator.

State **three** advantages of using built-in functions.

1 .....

2 .....

3 .....

[3]

(b) A student is learning about random number generation.

She is investigating how many times the random function needs to be called before every number in a given series is generated.

She is using **pseudocode** to develop a procedure, `TestRand()`, which will:

- use the random number function to generate an integer value in the range 1 to 50 inclusive
- count how many times the random function needs to be called before all 50 values have been generated
- output a message giving the number of times the random function was called.

13

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

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

..... [8]

6 A text file, `MyCDs.txt`, stores information relating to a Compact Disc (CD) collection. Information about each CD is stored on three separate lines in the file as follows:

```
Line 1:   <Artist Name>
Line 2:   <CD Title>
Line 3:   <Storage Location>
```

Information is stored as data strings.

A section of the file is shown:

File line	Data
100	`"Green Floyd"`
101	`"Bowlful of Cereal"`
102	`"Shelf 4"`
103	`"Strolling Bones"`
104	`"Exile on Station Road"`
105	`"Box 12"`

(a) A program, `CDOrganiser`, will be written to manage the stored information. The program will consist of three modules: `AddCD`, `FindCD` and `RemoveCD`.

Give **three** reasons why it is good practice to construct the program using modules.

1 .....

2 .....

3 .....

[3]

(b) The module, `FindCD()`, will check whether a given CD exists in the collection. The module will be implemented as a function.

The function will:

- be called with two strings as parameters, representing the artist name and CD title
- return a string that gives the storage location, or an empty string if the given CD has not been found.

Write **program code** for the function `FindCD()`.

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]

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`

`ASC(ThisChar : CHAR) RETURNS INTEGER`
returns the ASCII value of character `ThisChar`

Example: `ASC('A')` returns `65`

`RAND(x : INTEGER) RETURNS REAL`
returns a real number in the range `0` to `x` (`x` not inclusive).

Example: `RAND(87)` could return `35.43`

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`