Cambridge International AS & A Level

Cambridge International Examinations
Cambridge International Advanced Subsidiary and Advanced Level

CANDIDATE NAME

CENTRE NUMBER

CANDIDATE NUMBER

COMPUTER SCIENCE **9608/31**

Paper 3 Advanced Theory **October/November 2018**

1 hour 30 minutes

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.

Question 1 begins on the next page.

1 Consider the following user-defined data type.

```
TYPE Book
     DECLARE ISBN     : INTEGER
     DECLARE Author   : STRING
     DECLARE Title    : STRING
     DECLARE Supplier : (Amazone, Stones, Smiths, Blackwalls, Greens,
                         Coals, Boarders)
ENDTYPE
```

(a) Name the data type of `Book`.

.....[1]

(b) Name the non-composite data type used in the `Supplier` declaration.

.....[1]

(c) **(i)** Write a pseudocode statement to declare a variable, `BestSeller`, of type `Book`.

.....[1]

(ii) Write a pseudocode statement to assign "`John Williams`" to the author of `BestSeller`.

.....[1]

[Turn over

2 **(a)** A computer system stores real numbers using floating-point representation. The floating-point numbers have:

- eight bits for the mantissa
- four bits for the exponent.

The mantissa and exponent are both in two's complement form.

(i) Calculate the denary value of the following floating-point number.

Mantissa								Exponent			
0	0	1	1	1	0	0	0	0	1	1	1

Show your working.

Working .....

.....

.....

.....

.....

Answer .....

[3]

(ii) State how you know the floating-point number in **part (a)(i)** is not normalised.

.....

.....[1]

(iii) Normalise the floating-point number in **part (a)(i)**.

Mantissa								Exponent			

[2]

(b) **(i)** Write the largest positive number that this system can represent as a normalised floating-point number in this format.

Mantissa								Exponent			

[2]

(ii) Write the smallest positive number that can be stored as a normalised floating-point number in this format.

Mantissa								Exponent			

[2]

(c) The number of bits available to represent a real number is increased to 16.

State the effect this has on the numbers that can be represented, if the additional four bits are used in the:

(i) mantissa .....

.....[1]

(ii) exponent .....

.....[1]

(d) A student enters the following code into an interpreter.

```
X = 0.1
Y = 0.2
Z = 0.3
OUTPUT (X + Y + Z)
```

The student is surprised to see the output:

```
0.6000000000000001
```

Explain why this is output.

.....

.....

.....

.....

.....

.....[3]

3 A local college has CSMA/CD in operation on its Local Area Network (LAN).

(a) One function of CSMA/CD is to monitor traffic on the network.

State **two** other tasks performed by CSMA/CD.

1 .....

2 .....
[2]

(b) The network uses the TCP/IP protocol to transfer files across the network.

(i) State **three** functions of the **TCP** part of this protocol.

1 .....

.....

2 .....

.....

3 .....

.....
[3]

(ii) State **two** functions of the **IP** part of this protocol.

1 .....

2 .....
[2]

(iii) Identify **one** other common protocol that could be used to transfer files across the college network.

.....[1]

(c) Protocols are essential for successful transmission of data over a network. The TCP/IP protocol suite operates on many layers.

Give an appropriate protocol for each layer in the table.

Layer	Protocol
Application	
Transport	
Internet	

[3]

(d) The TCP/IP protocol is used to send an email message from one node on a LAN to a node on a different LAN.

State the steps that take place when the email message is sent and received.

.....

.....

.....

.....

.....

.....

.....

.....[4]

4 **(a)** A Boolean expression corresponds to the following truth table.

INPUT			OUTPUT
A	**B**	**C**	**X**
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

(i) Write the Boolean expression for the truth table by applying the sum-of-products.

X = ...[2]

(ii) Complete the Karnaugh Map (K-map) for the truth table.

		AB			
		00	**01**	**11**	**10**
C	**0**				
	1				

[1]

(iii) The K-map can be used to simplify the expression in **part (a)(i)**.

Draw loop(s) around appropriate groups of 1s in the table in **part (a)(ii)** to produce an optimal sum-of-products. [3]

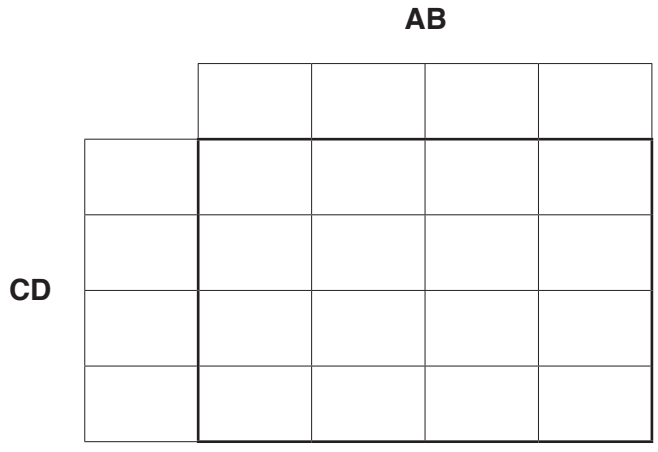
(iv) Write the simplified sum-of-products expression for your answer to **part (a)(iii)**.

X = ...[3]

(b) A logic circuit with four inputs produces the following truth table.

INPUT				OUTPUT
A	**B**	**C**	**D**	**X**
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

(i) Complete the K-map that corresponds to the truth table.



[4]

(ii) Draw loop(s) around appropriate groups of 1s in the table in **part (b)(i)** to produce an optimal sum-of-products. [2]

(iii) Write the simplified sum-of-products expression for your answer to **part (b)(ii)**.

X = ...[2]

5 A computer process can be in one of three states: running, ready or blocked.

(a) Explain how the processes are affected when the following events take place.

(i) The running process needs to read a file from a disk.

.....

.....

.....

.....[2]

(ii) The running process uses up its time slice.

.....

.....

.....

.....[2]

(b) (i) State the conditions that are necessary for a process to move from the ready to the running state.

.....

.....

.....

.....[2]

(ii) State the conditions that are necessary for a process to move from the blocked to the ready state.

.....

.....

.....

.....[2]

(c) Give **three** reasons why process scheduling is needed.

1 .....

.....

2 .....

.....

3 .....

.....

[3]

6 The compilation process has a number of stages. The first stage is lexical analysis.

A compiler uses a keyword table and a symbol table. Part of the keyword table is shown.

- Tokens for keywords are shown in hexadecimal.
- All of the keyword tokens are in the range `00` – `5F`.

Keyword	Token
←	01
*	02
=	03
IF	4A
THEN	4B
ENDIF	4C
ELSE	4D
FOR	4E
STEP	4F
TO	50
INPUT	51
OUTPUT	52
ENDFOR	53

Entries in the symbol table are allocated tokens. These values start from 60 (hexadecimal).

Study the following code.

```
Start ← 1
INPUT Number
// Output values in a loop
FOR Counter ← Start TO 12
    OUTPUT Number * Counter
ENDFOR
```

(a) Complete the symbol table to show its contents after the lexical analysis stage.

Symbol	Token	
	Value	Type
`Start`	`60`	Variable
`1`	`61`	Constant

[3]

(b) The output from the lexical analysis stage is stored in the following table. Each cell stores one byte of the output.

Complete the output from the lexical analysis stage. Use the keyword table and your answer to **part (a)**.

`60`	`01`														

[2]

(c) The output of the lexical analysis stage is the input to the syntax analysis stage.

Identify **two** tasks in syntax analysis.

1 .....

.....

2 .....

.....

[2]

(d) The final stage of compilation is optimisation.

(i) Code optimisation produces code that minimises the amount of memory used.

Give **one** additional reason why code optimisation is performed.

.....

.....[1]

(ii) A student uses the compiler to compile some different code.

After the syntax analysis stage is complete, the compiler generates object code.

The following lines of code are compiled.

```
X ← A + B
Y ← A + B + C
Z ← A + B + C + D
```

The compilation produces the following assembly language code.

```
LDD 236     //     loads value A to accumulator
ADD 237     //     adds value B to accumulator
STO 512     //     stores accumulator in X
LDD 236     //     loads value A to accumulator
ADD 237     //     adds value B to accumulator
ADD 238     //     adds value C to accumulator
STO 513     //     stores accumulator in Y
LDD 236     //     loads value A to accumulator
ADD 237     //     adds value B to accumulator
ADD 238     //     adds value C to accumulator
ADD 239     //     adds value D to accumulator
STO 514     //     stores accumulator in Z
```

Rewrite the assembly language code after it has been optimised.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....[5]

BLANK PAGE

BLANK PAGE