Cambridge International A Level

Cambridge International Examinations
Cambridge International Advanced Level

CANDIDATE NAME			
CENTRE NUMBER		CANDIDATE NUMBER	

8509480502

COMPUTER SCIENCE **9608/33**

Paper 3 Advanced Theory **October/November 2015**

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.

This document consists of **12** printed pages.

DC (NH) 117457

CAMBRIDGE
International Examinations

[Turn over

1 In a particular computer system, real numbers are stored using floating-point representation with:

- 8 bits for the mantissa, followed by
- 8 bits for the exponent

Two's complement form is used for both mantissa and exponent.

(a) **(i)** A real number is stored as the following two bytes:

Mantissa

0	0	1	0	1	0	0	0

Exponent

0	0	0	0	0	0	1	1

Calculate the denary value of this number. Show your working.

....

....

....

....

....

....[3]

(ii) Explain why the floating-point number in **part (a)(i)** is not normalised.

....

....[2]

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

Mantissa

Exponent

[2]

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

Mantissa

Exponent

[2]

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

Mantissa

Exponent

[2]

(iii) If a positive number is added to the number in **part (b)(i)** explain what will happen.

....

....

....

....[2]

(c) A student writes a program to output numbers using the following code:

```
X ← 0.0
FOR i ← 0 TO 1000
   X ← X + 0.1
   OUTPUT X
ENDFOR
```

The student is surprised to see that the program outputs the following sequence:

0.0 0.1 0.2 0.2999999 0.3999999 ……

Explain why this output has occurred.

....

....

....

....

....

....[3]

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

- Tokens for keywords are shown in hexadecimal.
- All 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 piece of code:

```
Counter ← 1.5
INPUT Num1
  // Check values
IF Counter = Num1
   THEN
      Num1 ← Num1 + 5.0
ENDIF
```

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

Symbol	Token	
	Value	Type
Counter	60	Variable
1.5	61	Constant

[3]

(b) Each cell below represents one byte of the output from the lexical analysis stage.

Using the keyword table and your answer to **part (a)** complete the output from the lexical analysis.

60	01														

[2]

(c) This line of code is to be compiled:

```
A ← B + C + D
```

After the syntax analysis stage, the compiler generates object code. The equivalent code, in assembly language, is shown below:

```
LDD 234     //loads value B
ADD 235     //adds value C
STO 567     //stores result in temporary location
LDD 567     //loads value from temporary location
ADD 236     //adds value D
STO 233     //stores result in A
```

(i) Name the final stage in the compilation process that follows this code generation stage.

....[1]

(ii) Rewrite the equivalent code given above to show the effect of it being processed through this final stage.

....

....

....

....

....

....[2]

(iii) State **two** benefits of the compilation process performing this final stage.

Benefit 1 ....

....

Benefit 2 ....

....[2]

3 An email is sent from one email server to another using packet switching.

(a) State **two items** that are contained in an email packet apart from the data.

1 ....

2 ....[2]

(b) Explain the role of routers in sending an email from one email server to another.

....

....

....

....

....

....[3]

(c) Sending an email message is an appropriate use of packet switching.

Explain why this is the case.

....

....

....

....

....

....[2]

(d) Packet switching is not always an appropriate solution.

Name an alternative communication method of transferring data in a digital network.

....[1]

(e) Name an application for which the method identified in **part (d)** is an appropriate solution. Justify your choice.

Application ....

Justification ....

....

....

....

....[3]

4 **(a)** Three descriptions and two types of processor are shown below.

Draw a line to connect each description to the appropriate type of processor.

Description	**Type of processor**
Makes extensive use of general purpose registers	RISC
Many addressing modes are available	CISC
Has a simplified set of instructions	

[3]

(b) In a RISC processor three instructions (A followed by B, followed by C) are processed using pipelining.

The following table shows the five stages that occur when instructions are fetched and executed.

(i) The 'A' in the table indicates that instruction A has been fetched in time interval 1.

Complete the table to show the time interval in which each stage of each instruction (A, B, C) is carried out.

Stage	**Time interval**								
	1	2	3	4	5	6	7	8	9
Fetch instruction	A								
Decode instruction									
Execute instruction									
Access operand in memory									
Write result to register									

[3]

(ii) The completed table shows how pipelining allows instructions to be carried out more rapidly. Each time interval represents one clock cycle.

Calculate how many clock cycles are saved by the use of pipelining in the above example.

Show your working.

....

....

....

....

....

....[3]

5 (a) (i) Complete the Boolean function that 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	0
1	1	0	1
1	1	1	1

X = $\overline{A}$. B . C + ...[3]

The part to the right of the equals sign is known as the sum-of-products.

(ii) For the truth table above complete the Karnaugh Map (K-map).

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

[1]

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

(iii) Draw loop(s) around appropriate groups of 1's to produce an optimal sum-of-products. [2]

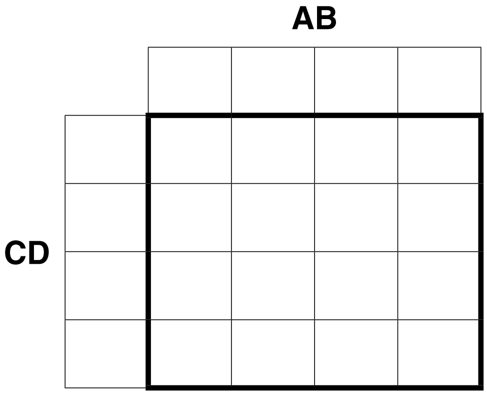
(iv) Using your answer to **part (a)(iii)**, write the simplified sum-of-products Boolean function.

X = ...[2]

(b) The truth table for a logic circuit with four inputs is given below:

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	1
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	0
1	1	1	0	1
1	1	1	1	1

(i) Complete the K-map corresponding to the truth table above.



[4]

(ii) Draw loop(s) around appropriate groups of 1's to produce an optimal sum-of-products. [2]

(iii) Using your answer to **part (b)(ii)**, write the simplified sum-of-products Boolean function.

X = ...[2]

6 A number of processes are being executed in a computer.

(a) Explain the difference between a program and a process.

....

....

....

....[2]

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

(b) For each of the following, the process is moved from the first state to the second state. Describe the conditions that cause each of the following changes of the state of a process:

From running to ready ....

....

....

....

From ready to running ....

....

....

....

From running to blocked ....

....

....

....[6]

(c) Explain why a process cannot be moved from the blocked state to the running state.

....

....

....

....

....

....[3]

(d) Explain the role of the high-level scheduler in a multiprogramming operating system.

....

....

....

....[2]