Cambridge Assessment International Education

Cambridge International AS & A Level

CANDIDATE NAME

CENTRE NUMBER

CANDIDATE NUMBER

COMPUTER SCIENCE **9618/31**

Paper 3 Advanced Theory **October/November 2021**

1 hour 30 minutes

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. Any blank pages are indicated.

[Turn over

1 **(a)** Numbers are stored in a computer using floating-point representation with:

- 12 bits for the mantissa
- 4 bits for the exponent
- two's complement form for both the mantissa and exponent.

(i) Write the normalised floating-point representation of the following unsigned binary number using this system.

1011100.011001

Working .....

.....

.....

.....

.....

Mantissa												Exponent			

[2]

(ii) State the consequence of storing the binary number in **part (a)(i)** as a floating-point number in this system. Justify your answer.

Consequence .....

.....

Justification .....

.....

.....

[2]

(b) Explain the reason why binary numbers are stored in normalised form.

.....

.....

.....

.....

.....

.....

..... [3]

2 Draw **one** line from each programming paradigm to its **most appropriate** description.

Programming paradigm	Description
Declarative	Programs using the instruction set of a processor
Imperative	Programs based on events such as user actions or sensor outputs
Low-level	Programs using the concepts of class, inheritance, encapsulation and polymorphism
Object-oriented	Programs with an explicit sequence of commands that update the program state, with or without procedure calls
	Programs that specify the desired result rather than how to get to it

[4]

3 Enumerated and pointer are two non-composite data types.

(a) Write **pseudocode** to create an enumerated type called `Parts` to include these parts sold in a computer shop:

```
Monitor, CPU, SSD, HDD, LaserPrinter, Keyboard, Mouse
```

.....

.....

..... [2]

(b) Write **pseudocode** to create a pointer type called `SelectParts` that will reference the memory location in which the current part name is stored.

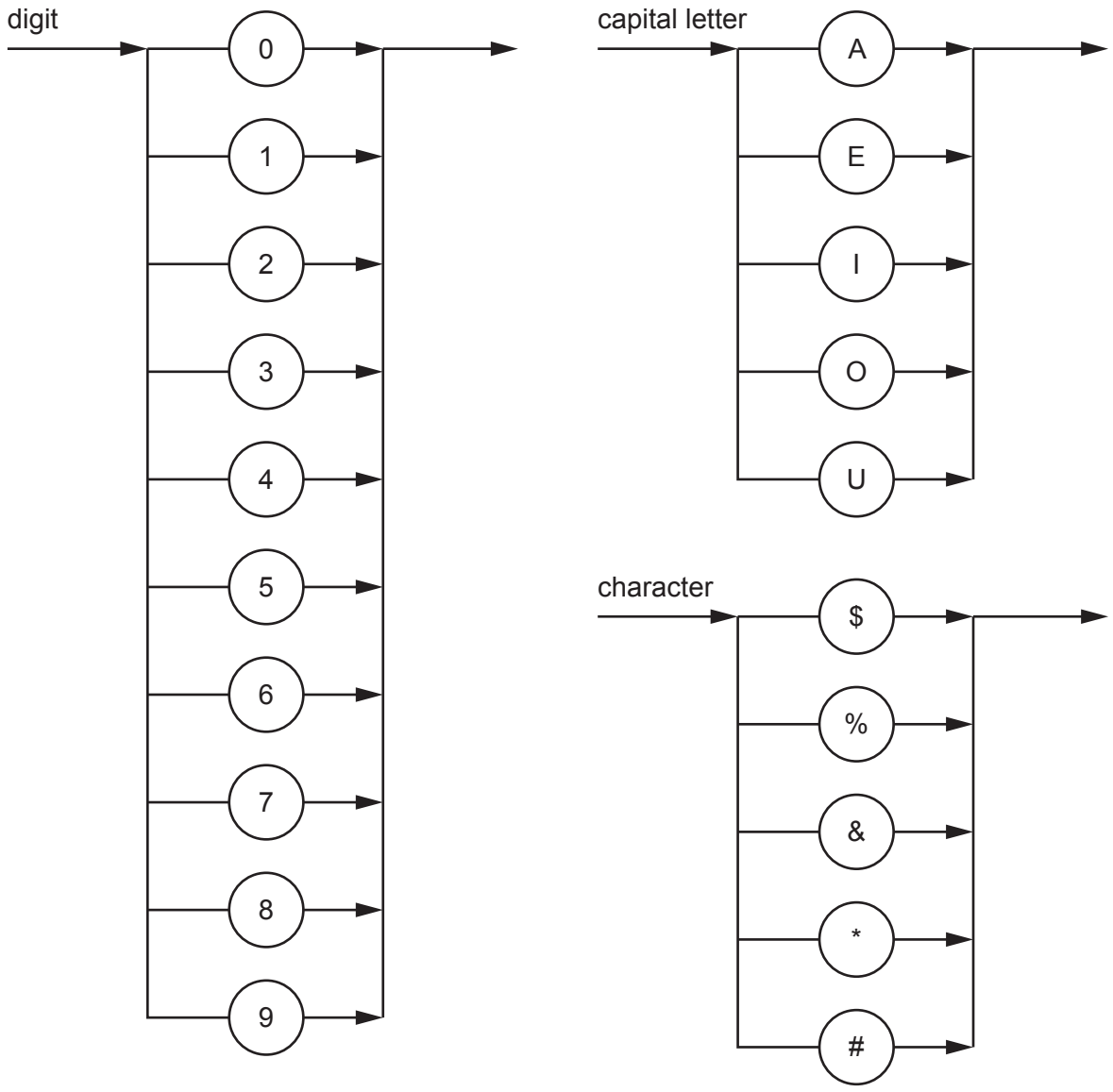
.....

.....

..... [2]

4 The following syntax diagrams for a particular programming language show the syntax of:

- a digit
- a capital letter
- a character.



(a) Write the Backus-Naur Form (BNF) notation of the syntax diagram for character.

.....

.....

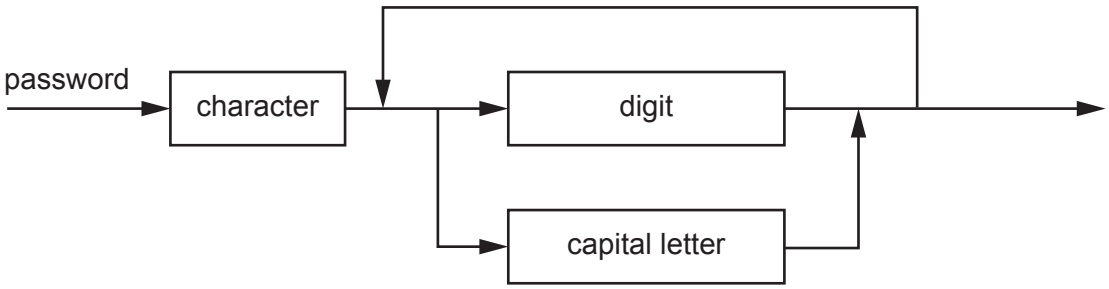
..... [2]

(b) A password must begin with a character and be followed by one or more digits or capital letters.

(i) State an example of a valid password.

..... [1]

(ii) A valid password is represented by the syntax diagram:



Write the BNF notation of the syntax diagram for password.

.....

.....

.....

.....

.....

.....

.....

..... [4]

5 **(a)** Compare sequential and serial methods of file organisation.

.....

.....

.....

.....

.....

.....

.....

.....

..... [4]

(b) State the most suitable method of file access when a record is referenced by a unique address on a disk-type storage medium.

..... [1]

(c) State the most suitable method of file access when a bank stores its data records in ascending order of account number.

..... [1]

6 **(a)** Explain how packet switching is used to transfer messages across the internet.

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [5]

(b) Outline the function of a router in packet switching.

.....

.....

.....

.....

.....

..... [3]

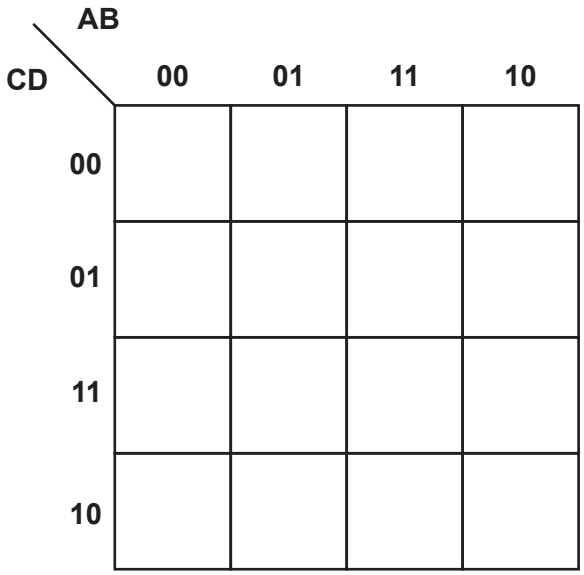
7 **(a)** Write the Boolean expression that corresponds to the given truth table as a sum-of-products.

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

Z = ...

..... [3]

(b) (i) Complete the Karnaugh map (K-map) for the given truth table.



[2]

(ii) Draw loop(s) around appropriate group(s) of 1s in the K-map to produce an optimal sum-of-products. [2]

(iii) Write the Boolean expression from your answer to **part b(ii)** as a simplified sum-of-products.

Z = ...

..... [2]

(iv) Write the simplified Boolean expression for your answer to **part b(iii)**.

Z = ...

..... [1]

8 **(a)** Describe the purpose of the Secure Sockets Layer (SSL) and Transport Layer Security (TLS) protocols.

.....

.....

.....

.....

..... [2]

(b) Explain how SSL/TLS protocols are used when a client-server communication is initiated.

.....

.....

.....

.....

.....

.....

.....

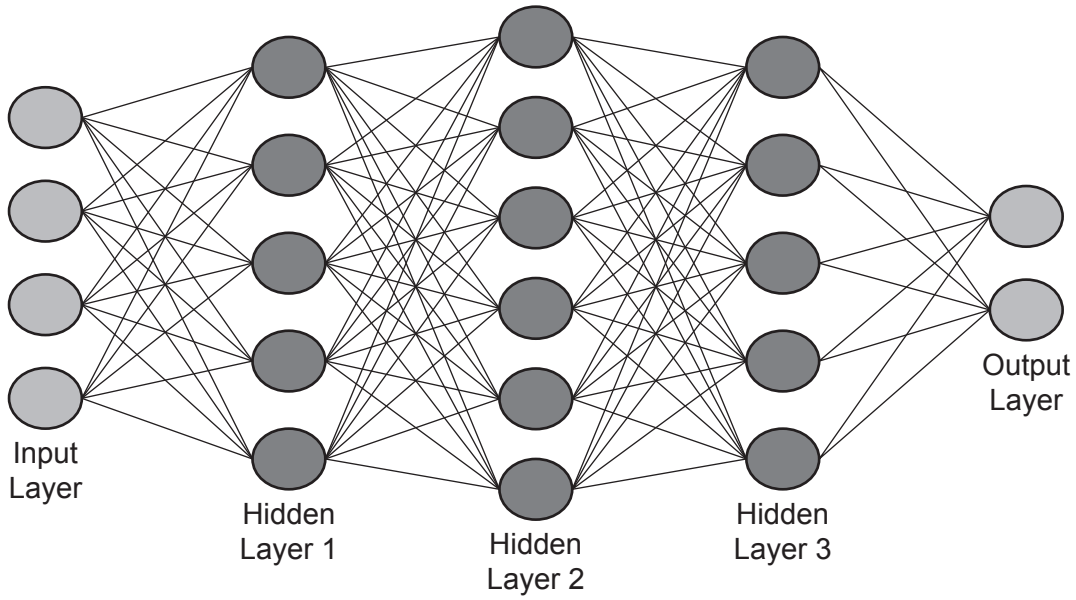
.....

.....

.....

..... [4]

9 **(a)** The diagram shown represents an artificial neural network.



(i) State the reason for having multiple hidden layers in an artificial neural network.

.....

..... [1]

(ii) Explain how artificial neural networks enable machine learning.

.....

.....

.....

.....

.....

.....

.....

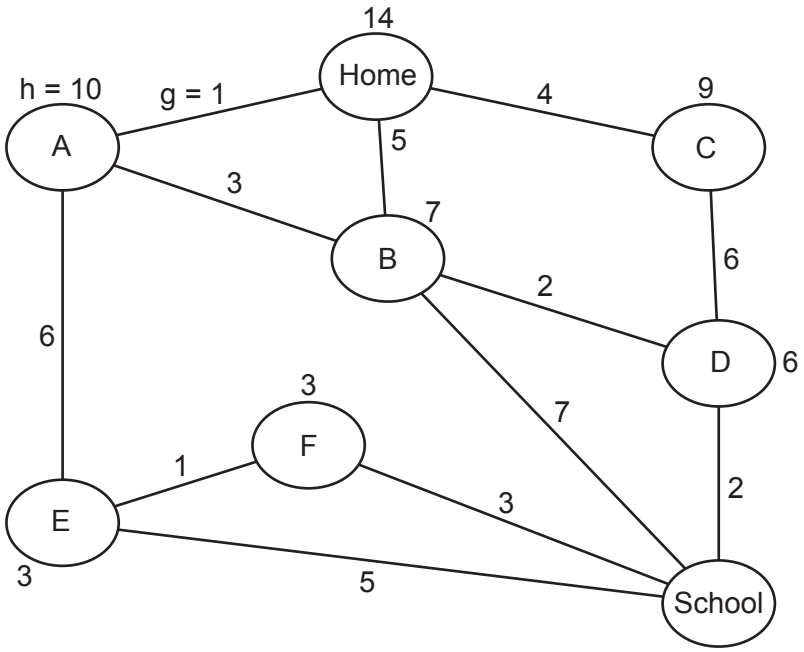
.....

.....

..... [4]

(b) Find the shortest path between the Home and School nodes using the A* algorithm. Show your working in the table provided.

The first two rows in the table have been completed.



Node	Cost from Home node (g)	Heuristic (h)	Total (f = g + h)
Home	0	14	14
A	1	10	11

Final path	

[5]

10 (a) State **three** essential features of **recursion**.

1 .....

.....

2 .....

.....

3 .....

.....

[3]

(b) Explain the reasons why a stack is a suitable Abstract Data Type (ADT) to implement recursion.

.....

.....

.....

.....

.....

..... [3]

(c) Identify **two** ADTs other than a stack.

1 .....

2 .....

[2]

(d) The function `StackFull()` checks whether a stack is full.

The function uses the variable `TopOfStack` to represent the pointer to the most recent position used on the stack, and the variable `Max` to represent the maximum size of the stack. Assume `TopOfStack` and `Max` are global variables.

```
FUNCTION StackFull() RETURNS BOOLEAN
    IF TopOfStack = Max THEN
       RETURN TRUE
    ELSE
       RETURN FALSE
    ENDIF
ENDFUNCTION
```

An algorithm `AddInteger` is required to add a new integer data element to a stack.

The stack is implemented as an array `ArrayStack`.

The function `AddInteger()` calls `StackFull()` and returns an appropriate message.

Complete the pseudocode for the function `AddInteger()`.

```
FUNCTION AddInteger(NewInteger : INTEGER) RETURNS STRING
```

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

```
ENDFUNCTION
```

[5]

BLANK PAGE

BLANK PAGE