Cambridge Assessment International Education

# Cambridge International AS & A Level

CANDIDATE NAME

CENTRE NUMBER

CANDIDATE NUMBER



**COMPUTER SCIENCE** **9618/32**

Paper 3 Advanced Theory **October/November 2025**

**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** The composite data type, `Car`, is defined in pseudocode as:

```
TYPE Car
   DECLARE RegNumber : STRING
   DECLARE Make : STRING
   DECLARE Model : STRING
   DECLARE BodyStyle : STRING
   DECLARE Colour : STRING
   DECLARE IntoStock : DATE
   DECLARE Price : REAL
ENDTYPE
```

**(a) (i)** Write the **pseudocode** statement to set up a variable for one record of the composite data type, `Car`.

.....

..... [1]

**(ii)** Write the **pseudocode** statements to assign the following values to the variable set up in part **(a)(i)**:

- `"Blue"` to `Colour`
- `21/10/2025` to `IntoStock`

.....

.....

.....

..... [2]

**(b)** The data type for `BodyStyle` is changed to an enumerated type, `Body`.

**(i)** Write the **pseudocode** statement for the type declaration of `Body` to hold the names of the available choices:

Convertible, Hatchback, Saloon, SUV

.....

.....

.....

..... [2]

**(ii)** Write the new **pseudocode** statement required to update the declaration of `BodyStyle` in the definition of `Car`.

.....

..... [1]

**2** Numbers are stored in a computer system using binary floating-point representation with:

- 10 bits for the mantissa
- 6 bits for the exponent
- two's complement form for both the mantissa and the exponent.

**(a)** Calculate the denary value of the given normalised binary floating-point number. Show your working.

| Mantissa | | | | | | | | | | Exponent | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 |

Working .....

.....

.....

.....

.....

Denary value .....

[3]

**(b)** Calculate the normalised binary floating-point representation of +26.6875 in this system. Show your working.

| Mantissa | | | | | | | | | | Exponent | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | | | | |

Working .....

.....

.....

.....

.....

.....

[3]

**3** **(a)** HTTP and IMAP are examples of protocols used in the Application Layer of the TCP/IP protocol suite.

State the purpose of the HTTP and IMAP protocols.

HTTP .....

.....

IMAP .....

.....

[2]

**(b)** Describe how files are shared using the BitTorrent protocol.

.....

.....

.....

.....

.....

.....

.....

..... [4]

**4** **(a)** Identify **one** benefit of circuit switching **and one** benefit of packet switching.

Circuit switching .....

.....

Packet switching .....

.....

[2]

**(b)** Identify **two** differences between circuit switching and packet switching.

1 .....

.....

.....

2 .....

.....

.....

[2]

**5** **(a)** Describe how interrupt handling is used in low-level scheduling.

.....

.....

.....

.....

.....

..... [2]

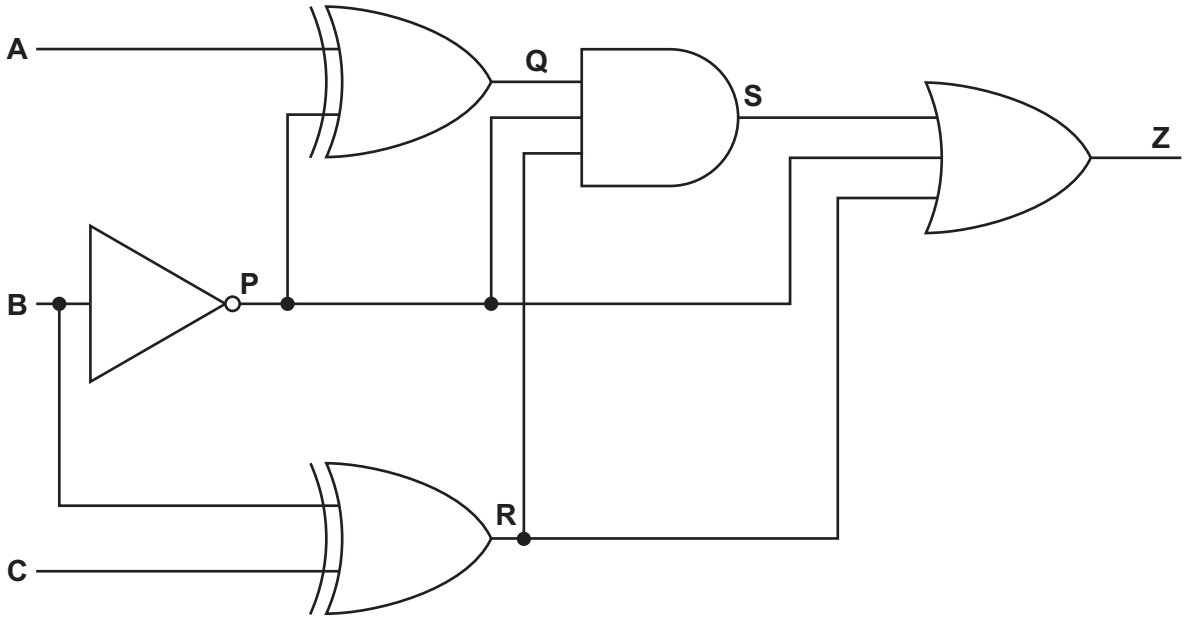
**(b)** In process management, a process can be in one of three process states: running, ready or blocked.

Complete the table to identify **one** reason why a process could be in each of the three states.

| Process state | Reason |
|---|---|
| running | ..... ..... ..... ..... |
| ready | ..... ..... ..... ..... |
| blocked | ..... ..... ..... ..... |

[3]

**6** **(a)** The diagram shows a logic circuit.



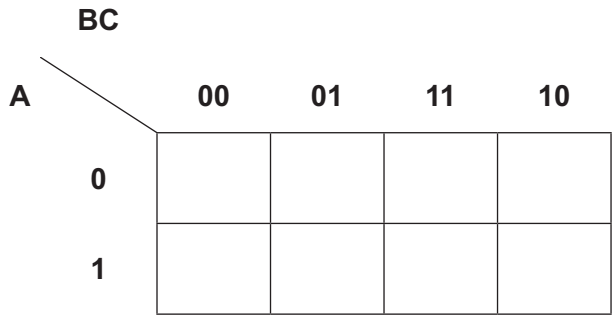

Complete the truth table for the given logic circuit.
Show your working.

| | | | Working space | | | | |
|---|---|---|---|---|---|---|---|
| **A** | **B** | **C** | **P** | **Q** | **R** | **S** | **Z** |
| 0 | 0 | 0 | | | | | |
| 0 | 0 | 1 | | | | | |
| 0 | 1 | 0 | | | | | |
| 0 | 1 | 1 | | | | | |
| 1 | 0 | 0 | | | | | |
| 1 | 0 | 1 | | | | | |
| 1 | 1 | 0 | | | | | |
| 1 | 1 | 1 | | | | | |

[3]

**(b)** **(i)** Complete the Karnaugh map (K-map) for the Boolean expression:

$\overline{A}.\overline{B}.C + \overline{A}.B.C + A.\overline{B}.C + A.B.\overline{C}$




[2]

**(ii)** Draw loop(s) around appropriate group(s) 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. Do **not** carry out any further simplification.

.....

..... [2]

**7** **(a)** Asymmetric encryption is a type of cryptography.

Identify **one** other type of cryptography.

.....

..... [1]

**(b)** An organisation holds two asymmetric encryption keys, which they intend to use to receive secure transmissions.

Explain how the organisation makes use of the two keys to receive a secure transmission.

.....

.....

.....

.....

.....

.....

.....

..... [4]

**8** **(a)** State the purpose of the optimisation stage in the compilation of a program.

.....

.....

.....

..... [1]

**(b)** Convert this Reverse Polish Notation (RPN) back to its original infix form:

`a b - c + c a - * d /`

.....

.....

.....

.....

.....

..... [3]

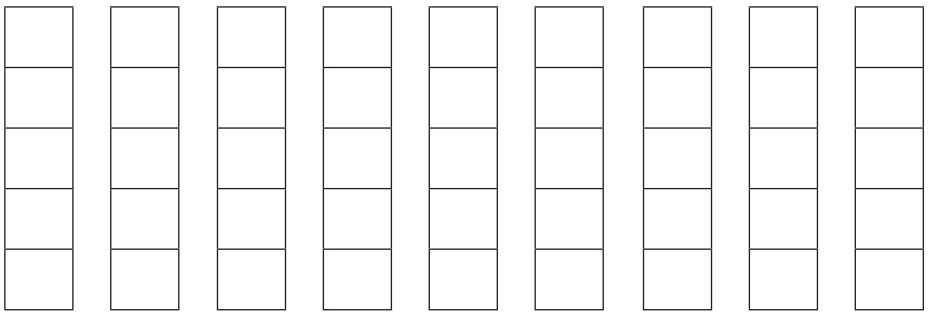
**(c)** The RPN expression:

`c a / b d - * b +`

is to be evaluated, where:

`a = 3`, `b = 16`, `c = 9` and `d = 6`.

Show the changing contents of the stack as the RPN expression is evaluated.



[4]

**9** **(a)** Deep Learning is a form of Machine Learning.

**(i)** State **one** example where Deep Learning is used.

.....

..... [1]

**(ii)** Identify how Deep Learning can be made more effective.

.....

..... [1]

**(b)** Describe the back propagation of errors method in Machine Learning.

.....

.....

.....

.....

.....

.....

.....

..... [4]

**10** An exception is an error that may cause a program to halt unexpectedly.

**(a)** Describe how program termination due to an exception can be avoided.

.....

.....

.....

.....

..... [2]

**(b)** Identify **two** possible causes of exceptions.

1 .....

.....

2 .....

..... [2]

**11** The table shows assembly language instructions for a processor that has one register, the Accumulator (ACC).

| Label | Instruction | | Explanation |
|---|---|---|---|
| | **Opcode** | **Operand** | |
| | LDM | #n | Load the number n to the ACC |
| | LDD | <address> | Load the contents of the location at the given address to ACC |
| | LDI | <address> | The address to be used is at the given address. Load the contents of this second address to the ACC. |
| | ADD | <address> | Add the contents of the given address to the ACC |
| | SUB | <address> | Subtract the contents of the given address from the ACC |
| | STO | <address> | Store the contents of the ACC at the given address |
| <label>: | | <data> | Gives a symbolic address <label> to the memory location with the contents <data> |
| # denotes a denary number, e.g. #123<br><label> can be used in place of <address> | | | |

The current contents of memory are:

| Address | Contents |
|---|---|
| 150 | 26 |
| 300 | 86 |
| 420 | 150 |

Write **assembly language** code, using **only** the given instruction set to:

- store the contents of location 300 as labelled variable `A`
- store the contents of location 420 as labelled variable `B`
- add the value stored in the address contained in variable `B` to the value contained in variable `A`
- store the result in variable `Answer`.

Show the initialisation and values of the variables `A`, `B` and `Answer` in the table provided.

.....

.....

.....

.....

.....

.....

.....

.....

| Label | Content |
|---|---|
| | |
| | |
| | |
| | |

[7]

**12 (a)** A stack has been implemented using pseudocode to store a maximum of 100 string items using the global variables in the following table:

| Identifier | Data type | Description | Initialisation value |
|---|---|---|---|
| `Base` | `INTEGER` | pointer for the bottom of the stack | `0` |
| `Top` | `INTEGER` | pointer for the top of the stack | `-1` |
| `StackArray` | `STRING` | 1D array to implement the stack | `[0:99]` |
| `Max` | `INTEGER` | maximum number of items in the stack | `100` |

The value of `Top` is incremented each time a data item is added to the stack and decremented each time a data item is removed. If the stack is full, an appropriate error message is output.

**(i)** Complete the **pseudocode** for the procedure to add a data item onto the stack.

```
PROCEDURE Push(.....)
   IF Top < Max - 1 THEN

      Top ← .....

      ..... ← NewData
   ELSE

      OUTPUT .....
   ENDIF
ENDPROCEDURE
```

[4]

**(ii)** Write **pseudocode** to input a new data item and add it to the stack using `Push()`.

.....

.....

.....

..... [2]

**(b)** Explain the reasons why a stack is used when a recursive algorithm is executed.

.....

.....

.....

.....

.....

..... [3]

**BLANK PAGE**

**BLANK PAGE**

**BLANK PAGE**

**BLANK PAGE**