0 1 . 1 Define the term algorithm.

[2 marks]

0 1 . 2 The following are computer science terms (labelled **A** – **E**).

A abstraction
B data type
C decomposition
D efficiency
E input

For each of the definitions in the table, write the label of the most suitable computer science term. Use a label only once.

[3 marks]

	Label
Breaking a problem down into a number of sub-problems.	
The process of removing unnecessary detail from a problem.	
Defines the range of values a variable may take.	

Turn over for the next question

0 2 The algorithm in **Figure 2** is a sorting algorithm.

- Array indexing starts at 0.
- Line numbers are included but are not part of the algorithm.

Figure 2

```
1   arr ← [4, 1, 6]
2   sorted ← false
3   WHILE sorted = false
4      sorted ← true
5      i ← 0
6      WHILE i < 2
7         IF arr[i+1] < arr[i] THEN
8            t ← arr[i]
9            arr[i] ← arr[i+1]
10           arr[i+1] ← t
11           sorted ← false
12        ENDIF
13        i ← i + 1
14     ENDWHILE
15  ENDWHILE
```

0 2 . 1 State the data type of the variable `sorted` in the algorithm shown in **Figure 2**.

[1 mark]

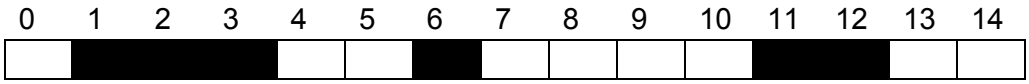
_____

0 3 A developer wants to simulate a simple version of the game of Battleships™. The ships are located on a one-dimensional array called `board`. There are always three ships placed on the board:

- one 'carrier' that has size three
- one 'cruiser' that has size two
- one 'destroyer' that has size one.

The size of the board is always 15 squares. A possible starting configuration is shown in **Figure 9** where the indices are also written above the board.

Figure 9



The carrier, for example, is found at locations `board[1]`, `board[2]` and `board[3]`.

A player makes a guess to see if a ship (or part of a ship) is located at a particular location. If a ship is found at the location then the player has 'hit' the ship at this location.

Every value in the `board` array is 0, 1 or 2.

- The value 0 is used to indicate an empty location.
- The value 1 is used to indicate if a ship is at this location and this location has **not** been hit.
- The value 2 is used to indicate if a ship is at this location and this location has been hit.

The developer identifies one of the sub-problems and creates the subroutine shown in **Figure 10**.

Figure 10

```
SUBROUTINE F(board, location)
   h ← board[location]
   IF h = 1 THEN
      RETURN true
   ELSE
      RETURN false
   ENDIF
ENDSUBROUTINE
```

0 3 . 1 The subroutine in **Figure 10** uses the values `true` and `false`. Each element of the array `board` has the value `0`, `1` or `2`.

State the most appropriate data type for these values.

[2 marks]

Values	Data type
`true`, `false`	
`0`, `1`, `2`	

0 4 The algorithm shown in **Figure 3** is used to check if the start of an instruction for a particular assembly language is valid.

The string representation of the assembly language instruction is stored in the variable `instr`

Characters in the string are indexed starting at zero. For example `instr[2]` would access the third character of the string stored in the variable `instr`

Figure 3

```
code ← ''
i ← 0
WHILE instr[i] ≠ ':' AND i < 4
   code ← code + instr[i]
   i ← i + 1
ENDWHILE
valid ← False
IF code = 'ADD' OR code = 'SUB' OR code = 'HALT' THEN
   valid ← True
ENDIF
```

0 4 . 1 Shade **one** lozenge to show the most appropriate data type of the variable `i` in the algorithm in **Figure 3**.

[1 mark]

A Character ⬭

B Integer ⬭

C Real ⬭

D String ⬭

0 4 . 2 State the data type of the variable `valid` in the algorithm in **Figure 3**.

[1 mark]

_____

0 4 . 3 State the final value of the variable `valid` in the algorithm in **Figure 3** for the following different starting values of `instr`

[3 marks]

Value of `instr`	**Final value of** `valid`
`ADD R0, R1`	
`ADD: R0, R1`	
`HALT`	

0 4 . 4 State what an assembly language program must be translated into before it can be executed by a computer.

[1 mark]

0 4 . 5 State **two** reasons why a programmer, who can program in both high-level and low-level languages, would usually choose to develop in a high-level language rather than a low-level language.

[2 marks]

Reason 1

Reason 2

0 4 . 6 Develop an algorithm, using either pseudo-code **or** a flowchart, that:

- initialises a variable called `regValid` to `False`
- sets a variable called `regValid` to `True` if the string contained in the variable `reg` is an uppercase `R` followed by the character representation of a single numeric digit.

Examples:
- if the value of `reg` is `R0` or `R9` then `regValid` should be `True`
- if the value of `reg` is `r6` or `Rh` then `regValid` should be `False`

You may wish to use the subroutine `isDigit(ch)` in your answer. The subroutine `isDigit` returns `True` if the character parameter `ch` is a string representation of a digit and `False` otherwise.

[3 marks]

0 5 A developer needs to store data about thousands of songs in a program. She needs to be able to hold information on every song's title, singer and year of release.

Explain how the developer could use a combination of an array and records to store this information.

In your answer you should refer to the data types that would be used by the developer.

[4 marks]

0 6 **Figure 3** shows an algorithm, represented using pseudo-code, that calculates the delivery cost for an order from a takeaway company.

Figure 3

```
orderTotal ← USERINPUT
deliveryDistance ← USERINPUT
deliveryCost ← 0.0
messageOne ← "Minimum spend not met"
messageTwo ← "Delivery not possible"
IF deliveryDistance ≤ 5 AND orderTotal > 0.0 THEN
   IF orderTotal > 50.0 THEN
      deliveryCost ← 1.5
      OUTPUT deliveryCost
   ELSE IF orderTotal > 25.0 THEN
      deliveryCost ← (orderTotal / 10) * 2
      OUTPUT deliveryCost
   ELSE
      OUTPUT messageOne
   ENDIF
ELSE
   OUTPUT messageTwo
ENDIF
```

0 6 . 1 Using **Figure 3**, complete the table.

[2 marks]

Input value of `orderTotal`	**Input value of** `deliveryDistance`	**Output**
55.5	2	
35.0	5	

0 6 . 2 State how many possible values the result of the comparison `deliveryDistance ≤ 5` could have in the algorithm shown in **Figure 3**.

[1 mark]

_____

0 6 . 3 State the most suitable data type for the following variables used in **Figure 3**.

[2 marks]

Variable identifier	Data type
`deliveryCost`	
`messageOne`	

0 6 . 4 State **one** other common data type that you have **not** given in your answer to Question **02.3**.

[1 mark]

Turn over for the next question

0 7 **Figure 8** shows an algorithm, written using pseudo-code, that uses a RECORD data structure for storing information about a film.

Each record stores four pieces of information about a film:

- film title
- certificate (eg 12A, PG)
- year the film was made
- if the film is currently being shown at a cinema.

There are records for three films and these films are stored alphabetically in an array called `filmCollection`.

The pseudo-code outputs the title of the newest of the three films.

- Part of the algorithm has been replaced by the label L1.

Figure 8

```
RECORD Film
   title : String
   certificate : String
   year : Integer
   beingShown : Boolean
ENDRECORD

hulk ← Film('Hulk', '12A', 2005, False)
ironMan ← Film('Iron Man', '12A', 2008, False)
antMan ← Film('Ant-Man', '12A', 2015, False)
filmCollection ← [antMan, hulk, ironMan]
year ← 0
position ← 0

FOR i ← 0 TO L1
   IF filmCollection[i].year > year THEN
      year ← filmCollection[i].year
      position ← i
   ENDIF
ENDFOR

OUTPUT filmCollection[position].title, ' is the
newest film'
```

0 7 . 1 How many different values can the field `beingShown` have?

Shade **one** lozenge.

[1 mark]

A 2 ⬭

B 3 ⬭

C 128 ⬭

D 256 ⬭

0 8 The algorithm in **Figure 4** is a sorting algorithm.

- Array indexing starts at 0.
- Line numbers are included but are not part of the algorithm.

Figure 4

```
1   arr ← [4, 1, 6]
2   swapsMade ← false
3   WHILE swapsMade = false
4      swapsMade ← true
5      i ← 0
6      WHILE i < 2
7         IF arr[i+1] < arr[i] THEN
8            t ← arr[i]
9            arr[i] ← arr[i+1]
10           arr[i+1] ← t
11           swapsMade ← false
12        ENDIF
13        i ← i + 1
14     ENDWHILE
15  ENDWHILE
```

0 8 . 1 State the data type of the variable `swapsMade` in the algorithm shown in **Figure 4**.

[1 mark]

0 9 **Figure 1** shows an algorithm, represented using pseudo-code.

The algorithm assigns different values to two variables, then asks the user to input a letter.

Figure 1

```
film ← "Godzilla vs. Kong"
year ← 2021
OUTPUT "Please guess a letter"
letter ← USERINPUT
```

0 9 . 1 Which pseudo-code statement assigns the length of the string `film` to a variable called `value`?

Shade **one** lozenge.

[1 mark]

A `film ← LEN(value)` ⬭

B `film ← film + value` ⬭

C `value ← film` ⬭

D `value ← LEN(film)` ⬭

0 9 . 2 The `POSITION` subroutine returns the position of the first occurrence of a character in a string.

For example:

- `POSITION("Godzilla vs. Kong", "o")` would return `1`
- `POSITION("Godzilla vs. Kong", "z")` would return `3`

`letter` and `film` are variables used in the algorithm in **Figure 1**.

Complete the pseudo-code statement to find the position of the first occurrence of the contents of `letter` in `film` and store this position in the variable `location`

You **must** use the `POSITION` subroutine in your answer.

[1 mark]

`location ←` _____

_____

0 9 . 3 Which of the following would be the most suitable data type for the variable `year`?

Shade **one** lozenge.

[1 mark]

A Boolean ⬭

B character ⬭

C integer ⬭

D real ⬭

0 9 . 4 Describe what is meant by an assignment statement in a program.

[1 mark]

_____

_____

0 9 . 5 Write a C# program that:

- gets the user to enter the name of a film
- displays `You entered` followed by the name of the film entered by the user.

The output from the program **must** be on one line.

You **should** use meaningful variable name(s) and C# syntax in your answer.

The answer grid below contains vertical lines to help you indent your code accurately.

[2 marks]