<table>
<tr><th colspan="3">Question</th><th>Answer/Indicative content</th><th>Marks</th><th>Guidance</th></tr>
<tr><td>1</td><td></td><td></td><td>1 mark per row

<table><tr><th>Operator</th><th>Comparison</th><th>Arithmetic</th></tr><tr><td>==</td><td>✓</td><td></td></tr><tr><td>+</td><td></td><td>✓</td></tr><tr><td>DIV</td><td></td><td>✓</td></tr><tr><td>></td><td>✓</td><td></td></tr></table></td><td>4
(AO1 1a)</td><td></td></tr>
<tr><td></td><td></td><td></td><td>Total</td><td>4</td><td></td></tr>
<tr><td>2</td><td>a</td><td></td><td>7</td><td>1
(AO3 1)</td><td>Correct Answer Only</td></tr>
<tr><td></td><td>b</td><td></td><td>4</td><td>1
(AO3 1)</td><td>Correct Answer Only</td></tr>
<tr><td></td><td>c</td><td></td><td>6</td><td>1
(AO3 1)</td><td>Correct Answer Only</td></tr>
<tr><td></td><td>d</td><td></td><td>6</td><td>1
(AO3 1)</td><td>Correct Answer Only</td></tr>
<tr><td></td><td></td><td></td><td>Total</td><td>4</td><td></td></tr>
<tr><td>3</td><td>a</td><td></td><td>Linear search</td><td>1
(AO1 1b)</td><td>Accept Sequential Search</td></tr>
<tr><td></td><td>b</td><td></td><td>• Declares variable to store the count (before the for loop)
• Initialises variable to zero
• Increments variable by 1 each time the word is found
• Outputs count value</td><td>4
(AO3 2c)</td><td><u>Example answer :</u>
<pre>search = input("Enter a word")
count = 0
for i = 0 to 7
    if data[i] == search then
    count = count + 1
next i
print(count)</pre>Allow answers that completely rewrite the algorithm as long as points on left met.</td></tr>
<tr><td></td><td>c</td><td>i</td><td>• A is sorted
• B is not (yet) sorted</td><td>2
(AO1 1b)
(AO2 1a)</td><td></td></tr>
<tr><td></td><td></td><td>ii</td><td>Max 3 marks, 1 mark per point

• Compare "or" with next item on left
• …"or" is smaller than "when" so…
• … it repeats comparison with next item
• inserts "or" / item in correct place, based on comparisons
• Repeat to insert "it"</td><td>3
(AO2 1b)</td><td>Max two marks if candidate simply states to insert "or" then "it" into the correct place without discussing how this is determined. Max one if generic answer with no reference to given array.

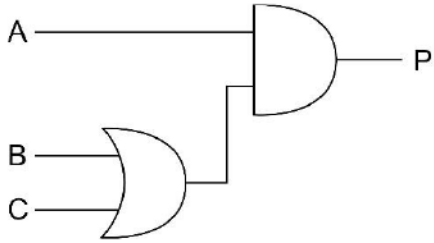
Allow answers referring to swapping items down the array to get to the correct position.</td></tr>
</table>

Question			Answer/Indicative content	Marks	Guidance
	d		• Merge sort	1 (AO1 1b)	Allow other sorting algorithms that use divide and conquer
			Total	11	
4			• decomposition • abstraction • thinking • sequence	4 (AO2 1a)	Ignore minor misspellings
			Total	4	

<table>
<tr><th colspan="3">Question</th><th>Answer/Indicative content</th><th>Marks</th><th>Guidance</th></tr>
<tr><td>5</td><td>a</td><td></td><td>Mark in pairs, 1 mark for advantage, 1 mark for expansion.
e.g.

• Easier / quicker to create…
• …. Language closer to English / uses keywords
• Provides higher level of abstraction from the underlying processor…
• … don't have to (usually) deal with allocating memory
• Portable // processor independent…
• … can write once for many processor types (e.g. mobile)</td><td>4
(AO1 1b, AO2 1b)</td><td>Accept other reasonable advantages and descriptions</td></tr>
<tr><td></td><td>b</td><td></td><td>1 mark per row

<table><tr><th>Statement</th><th>Compiler</th><th>Interpreter</th><th>Both</th></tr><tr><td>Translates high-level code to low-level instructions.</td><td></td><td></td><td>✓</td></tr><tr><td>Produces an executable file.</td><td>✓</td><td></td><td></td></tr><tr><td>Program needs to be translated every time it is run.</td><td></td><td>✓</td><td></td></tr></table></td><td>3
(AO1 1a)</td><td></td></tr>
<tr><td></td><td></td><td></td><td>Total</td><td>7</td><td></td></tr>
</table>

Question			Answer/Indicative content	Marks	Guidance
6	a		• B OR C • AND gate with A as one input… • …and output of BP1 as other input	3 (AO2 1b)	 Mark shape of gates, ignore candidate annotation.
	b		One mark per highlighted section **A** **B** **C** **P** 0 0 0 1 0 0 1 1 0 1 0 1 0 1 1 1 **1** 0 0 0 **1** 0 1 0 **1** 1 0 0 **1** 1 1 1	3 (AO1 1a, AO2 1a)	
	c		• **number** with **decimal places / fractional part**	1 (AO1 1a)	Do not accept examples on their own.
	d	i	• Boolean has two possible values / True or False // Result has **three** possible values // result has **more than two** options	1 (AO2 1a)	The results are words and not Boolean values.
		ii	1 mark for data type, 1 mark for matching justification. • String… • …can store "win", "loss" or "draw" // equivalent • Char… • …can store "W", "L" or "D" // equivalent • Integer / Real… • …can store 0, 1, 2 // equivalent	2 (AO2 1a)	Allow other sensible explanation that shows candidate has considered how each of the three states could be stored.

Question			Answer/Indicative content	Marks	Guidance

	e	i	1 mark per bullet point
			
			• Input code from user and store/use
			• Repeat for non-three character codes
			• Check for one code…
			• …and set level appropriately
			• Check for and set level for 2nd code
			• Set level to 1 for any other 3 character code
				6
				(AO3 2b)
					Example answer :
					`level = 0`
					`code = ""`
					`while code.length != 3 then`
					`  code = input("enter a 3 character code")`
					`endwhile`
					
					`switch code :`
					`  case "SVA":`
					`  level = 2`
					`  case "UTV":`
					`  level = 3`
					`  default:`
					`  level = 1`
					`endswitch`
					
					alternative example answer
					`level = 0`
					`code = ""`
					
					`while code.length != 3 then`
					`  code = input("enter a 3 character code")`
					`endwhile`
					`if code == "SVA" then`
					`  level = 2`
					`elseif code == "UTV" then`
					`  level = 3`
					`else`
					`  level = 1`
					`endif`

		ii	1 mark per bullet point
			
			• function `nextlevel()` defined with at least one parameter
			• returns parameter plus 1…
			• … except for if parameter is 3, then returns 1
				3
				(AO3 2b)
					Example answer :
					`function nextlevel(oldlevel)`
					`  if oldlevel == 3 then`
					`  return 1`
					`  else`
					`  return oldlevel + 1`
					`  endif`
					
					`end function`
					
					Accept correct use of MOD for BP2 and 3.
					
					Allow FT for BP3 if value output / calculated but not returned.

		iii	• print(nextlevel(3))	1 (AO3 2b)	Allow other logically correct answers
			Total	20	

<table>
<tr><th colspan="3">Question</th><th>Answer/Indicative content</th><th>Marks</th><th>Guidance</th></tr>
<tr><td>7</td><td>a</td><td>i</td><td>• DroneID, Mileage
• FROM
• AND
• > 50000</td><td>4
(AO3 2b)</td><td>Accept SELECT * or selection of additional fields for BP1.</td></tr>
<tr><td></td><td></td><td>ii</td><td>• <code>if Mileage-LastCheck > 10000</code>
• Output "Check" and "No Check" or equivalent correctly based on logical check for BP1</td><td>2
(AO3 2a)</td><td>BP1 could be >, < or either in words.

Ignore case and minor misspellings.

BP2 (output) could be either way around depending on comparison for BP1.</td></tr>
<tr><td></td><td>b</td><td></td><td>1 mark per row

• c = 90 on line 05
• c = 900 on line 05
• pilotCode = HK900 on line 07
• HK900 output on line 08</td><td>4
(AO3 1)</td><td>Ignore additional lines that do not affect outcome.
FT for missing or incorrect line numbers.
FT for output based on incorrect tracing of loop.

<table><tr><th>Line number</th><th>a</th><th>b</th><th>c</th><th>pilotCode</th><th>Output</th></tr><tr><td>01</td><td>H</td><td></td><td></td><td></td><td></td></tr><tr><td>02</td><td></td><td>K</td><td></td><td></td><td></td></tr><tr><td>03</td><td></td><td></td><td>9</td><td></td><td></td></tr><tr><td>05</td><td></td><td></td><td>90</td><td></td><td></td></tr><tr><td>05</td><td></td><td></td><td>900</td><td></td><td></td></tr><tr><td>07</td><td></td><td></td><td></td><td>HK900</td><td></td></tr><tr><td>08</td><td></td><td></td><td></td><td></td><td>HK900</td></tr></table></td></tr>
<tr><td></td><td>c</td><td></td><td>One mark per bullet point

• Any value between 0 and 20 (e.g. 4)
• True
• Invalid // erroneous // sensible alternative
• False</td><td>4
(AO3 2c)</td><td><table><tr><th>Experience in years</th><th>Type of test</th><th>Expected output</th></tr><tr><td>4</td><td>Normal</td><td>True</td></tr><tr><td>20</td><td>Boundary</td><td>True</td></tr><tr><td>32</td><td>Erroneous/Invalid</td><td>False</td></tr></table></td></tr>
</table>

Question			Answer/Indicative content	Marks	Guidance

	d	i	• Attempt at using selection
			• Calculates pay correctly for pilots with **less than** 2 years experience
			• Calculates pay correctly for pilots with 2 to 5 years experience. **Must include both 2 and 5**.
			• Calculates pay correctly for pilots with more than 5 years experience
				4
				(AO3 2b)
					Example answer :
					`experience = input(“Enter years of experience”)`
					`miles = input(“Enter miles flown”)`
					
					`totalPay = 0`
					`if exp <2 then`
					`  totalPay = 120+(0.45*miles)`
					`elseif exp <=5 then`
					`  totalPay = 150+(0.65*miles)`
					`else`
					`  totalPay = 180+(0.85*miles)`
					`end if`
					`print(totalPay)`
					
					FT for BP4 only where a reasonable attempt at calculating pay has been made.

		ii	• totalPay • (experience, miles) // (miles, experience)	2 (AO3 2c)	`totalPay = calculatePay(experience, miles)`

	e		One mark per bullet point
			
			• Ask the user for **two** inputs and store/use these
			• Open pilots.txt (for write/append)
			• Write **both** inputs **to opened text file**
			• Close text file
				4
				(AO3 2b)
					Award BP4 for implicit closing of file (e.g. using `with…` in Python)
					
					Example answer :
					`dronepilotData = open("pilots.txt")`
					`pilotCode = input("enter code")`
					`dob = input("enter date of birth")`
					`dronepilotData.writeLine(pilotCode, dob)`
					`dronepilotData.close()`
					
					Allow data to be written either by simply writing both values or by concatenating into one string with a separating comma.

Question			Answer/Indicative content	Marks	Guidance

	f		One mark per bullet point to max 6
			
			• Function header `pilotValid()` with (at least one) parameter
			• Checks array element [i,0] for **each** item…
			• …calculates total
			• …casting to **float / real**
			• Checks if 9 hours or fewer
			• **Returns** a value
			• …returns "warning" and "valid" correctly
				6
				(AO3 2b)
					Example answer :
					`function pilotValid(pilotCode)`
					`  total = 0`
					`  status = ""`
					`  for i = 0 to 5`
					`  if journeys[i,0] == pilotCode then`
					`  temp = float(journeys[i,1])`
					`  total = total + temp`
					`  endif`
					`  next i`
					`  if total > 9 then`
					`  status = "warning"`
					`  else`
					`  status = "valid"`
					`  endif`
					`  return status`
					`endfunction`
					
					Do not allow casting to integer for BP4, data shows some journeys have decimal places.
					
					Allow FT for BP5 if attempt made at calculation.
					Allow FT for BP7 if value is output instead of returned.

			Total	**30**	