

Mark Scheme

Question		Answer/Indicative content	Marks	Guidance															
1	a	<p>1 mark per row</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 70%;">Statement</th> <th style="width: 15%;">Low-level</th> <th style="width: 15%;">High-level</th> </tr> </thead> <tbody> <tr> <td>The same language can be used on computers that use different hardware</td> <td></td> <td style="text-align: center;">✓</td> </tr> <tr> <td>It allows the user to directly manipulate memory</td> <td style="text-align: center;">✓</td> <td></td> </tr> <tr> <td>It allows the user to write English-like words</td> <td></td> <td style="text-align: center;">✓</td> </tr> <tr> <td>It always needs to be translated into object code or machine code</td> <td></td> <td style="text-align: center;">✓</td> </tr> </tbody> </table>	Statement	Low-level	High-level	The same language can be used on computers that use different hardware		✓	It allows the user to directly manipulate memory	✓		It allows the user to write English-like words		✓	It always needs to be translated into object code or machine code		✓	<p>4 (AO1 1b)</p>	<p>No mark if more than 1 tick for that row.</p> <p>Allow other indications of choice (e.g. cross) as long as clear.</p> <p>Examiner's Comments</p> <p>This question was answered well by many candidates. The strongest responses showed a good understanding of the difference between low-level and high-level languages. Incorrect responses tended to be on the first row related to portability. A high-level language such as Python is portable, with translators available for many different types of processors. A low-level language is specific to one type of processor.</p>
Statement	Low-level	High-level																	
The same language can be used on computers that use different hardware		✓																	
It allows the user to directly manipulate memory	✓																		
It allows the user to write English-like words		✓																	
It always needs to be translated into object code or machine code		✓																	
	b	<p><code>total = num1 + num2</code></p>	<p>1 (AO3 2b)</p>	<p>Allow other logically valid responses that result in <code>total</code> storing the correct value. Accept other suitable assignment operators (e.g. <code>←</code>)</p> <p>e.g.</p> <pre>total = sum(num1, num2) total = num2 + num1 x = num1 + num2 total = x</pre> <p>Ignore any values given to the variable. Ignore capitalisation and minor misspelling. Ignore superfluous code that does not affect outcome.</p> <p>Examiner's Comments</p> <p>Candidates appear to be getting more confident at answering simple programming/pseudocode questions such as this. The majority of responses included code written to produce the required outcome. The use of multiple steps was allowed.</p>															
	c	i	<p><code>print(12 ^ 2)</code></p>	<p>1 (AO2 1a)</p> <p>Accept <code>**</code> or other sensible operator that indicates raising to a power.</p> <p>If pseudocode operator given, must be a single word/symbol (e.g. <code>pow</code>), not containing spaces.</p>															

Mark Scheme

Question		Answer/Indicative content	Marks	Guidance
	ii	if number MOD 2 == 0 then	1 (AO2 1a)	Accept % or other sensible operator that indicates modulus If pseudocode operator given, must be a single word/symbol (e.g. modulo), not containing spaces.
	iii	difference = measurement1 - measurement2	1 (AO2 1a)	Accept other sensible operator that indicates subtraction. If a pseudocode operator given, must be a single word/symbol (e.g. minus), not containing spaces.

Mark Scheme

Question	Answer/Indicative content	Marks	Guidance																																	
d	<p>1 mark each:</p> <ul style="list-style-type: none"> • Start is set to 3 on line 01 and 3 is output on line 03. • 2, 1 and 0 are output on next 3 iterations with start updated to 2, 1, 0, -1 on correct line numbers. • Finished is output on line 06 <table border="1" style="margin: 10px auto; border-collapse: collapse; text-align: center;"> <thead> <tr> <th style="padding: 2px;">Line</th> <th style="padding: 2px;">start</th> <th style="padding: 2px;">Output</th> </tr> </thead> <tbody> <tr style="background-color: #d9e1f2;"> <td style="padding: 2px;">01</td> <td style="padding: 2px;">3</td> <td style="padding: 2px;"></td> </tr> <tr style="background-color: #d9e1f2;"> <td style="padding: 2px;">03</td> <td style="padding: 2px;"></td> <td style="padding: 2px;">3</td> </tr> <tr style="background-color: #f4cccc;"> <td style="padding: 2px;">04</td> <td style="padding: 2px;">2</td> <td style="padding: 2px;"></td> </tr> <tr style="background-color: #f4cccc;"> <td style="padding: 2px;">03</td> <td style="padding: 2px;"></td> <td style="padding: 2px;">2</td> </tr> <tr style="background-color: #f4cccc;"> <td style="padding: 2px;">04</td> <td style="padding: 2px;">1</td> <td style="padding: 2px;"></td> </tr> <tr style="background-color: #f4cccc;"> <td style="padding: 2px;">03</td> <td style="padding: 2px;"></td> <td style="padding: 2px;">1</td> </tr> <tr style="background-color: #f4cccc;"> <td style="padding: 2px;">04</td> <td style="padding: 2px;">0</td> <td style="padding: 2px;"></td> </tr> <tr style="background-color: #f4cccc;"> <td style="padding: 2px;">03</td> <td style="padding: 2px;"></td> <td style="padding: 2px;">0</td> </tr> <tr style="background-color: #f4cccc;"> <td style="padding: 2px;">04</td> <td style="padding: 2px;">-1</td> <td style="padding: 2px;"></td> </tr> <tr style="background-color: #d9ead3;"> <td style="padding: 2px;">06</td> <td style="padding: 2px;"></td> <td style="padding: 2px;">Finished</td> </tr> </tbody> </table>	Line	start	Output	01	3		03		3	04	2		03		2	04	1		03		1	04	0		03		0	04	-1		06		Finished	<p>3 (AO3 2c)</p>	<p>Ignore lines 02 and 05 in answer unless these change or output any values.</p> <p>Candidate may repeat start value when unchanged, this is acceptable.</p> <p>Penalise incorrect or missing line numbers or <u>additional</u> output once only then FT. This includes where variable change and output appear on the same line.</p> <p>-1 must not be output for BP2</p> <p>Penalise missing or incorrect output once only for BP1 and FT for missing or incorrect output for BP2.</p> <p>Finished may be with or without quotes. Ignore case or minor spelling error.</p> <p>Max 2 marks if any incorrect output or changes to <i>start</i>.</p> <p>Do not accept calculated values of start (e.g. 3-1)</p> <p><u>Examiner's Comments</u></p> <p>This question assessed candidates' ability to trace through and understand the steps taken by an algorithm. This also tested their understanding of condition-controlled loops. Many responses were very successful with this and achieved full marks.</p> <p>Mistakes tended to be with identifying the line number where each change occurred or outputting values that were not actually output (e.g. -1).</p> <p>Examiners were instructed to only penalise a misunderstanding once. Where (for example) line numbers were incorrect, this would still have allowed 2 out of 3 marks to be achieved.</p>
Line	start	Output																																		
01	3																																			
03		3																																		
04	2																																			
03		2																																		
04	1																																			
03		1																																		
04	0																																			
03		0																																		
04	-1																																			
06		Finished																																		
	Total	11																																		

Mark Scheme

Question		Answer/Indicative content	Marks	Guidance
2	a	<p>1 mark each:</p> <p>Syntax error</p> <ul style="list-style-type: none"> • Error in the rules/grammar (of the program language). • Program does not (fully) run / translate / execute / start (BOD) <p>Logic error</p> <ul style="list-style-type: none"> • Produces incorrect / unexpected result/output • Program runs/does not crash 	2 (AO1 1a)	<p>Question asks for a definition. Examples may strengthen the response but are not acceptable by themselves.</p> <p>Do not allow “error/problem in the code, does not work / does not do what designed/intended to do” for either, this applies to both.</p> <p>“Error in the syntax” or “error in the logic” are NE even with examples</p> <p><u>Examiner’s Comments</u></p> <p>This question was answered extremely well by most candidates. These candidates correctly defined the terms given.</p> <p>Candidates must understand that an example is not the same as a definition. For example, a misspelling of a command such as <code>print</code> would of course be a syntax error but this is not a definition; many other issues would cause a syntax error.</p> <p>Other incorrect responses included generic responses that could apply to either term, for example: “a mistake in the program” or “where the computer doesn’t understand the code”.</p>
	b	<p>Line number</p> <ul style="list-style-type: none"> • 02 <p>Correction</p> <ul style="list-style-type: none"> • <code>for scoreCount = 0 to scores.length - 1</code> <p>Line number</p> <ul style="list-style-type: none"> • 03 <p>Correction</p> <ul style="list-style-type: none"> • <code>total = scores[scoreCount] + total</code> • <code>total = total + scores[scoreCount]</code> • <code>total += scores[scoreCount]</code> 	4 (AO3 2c)	<p>1 mark for each line number correctly identified.</p> <p>1 mark for each correction. Correction must match line number.</p> <p>If wrong line number, do not mark correction. If no line number, mark correction only.</p> <p>Do not penalise if response removes <code>-1</code> from <code>scores.length</code> as long as it starts at 0.</p> <p>Do not penalise potential off by 1 errors for looping (Python).</p> <p>Do not penalise case or minor spelling errors as long as intention is clear.</p> <p>Allow description of change that would be made (e.g. “change 1 to 0”)</p>

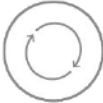
Mark Scheme

Question	Answer/Indicative content	Marks	Guidance
			<p>First correction is fixing indexing error so element 0 is included. This could be done on line 03 e.g. <code>scores[scoreCount-1]</code>. Second correction is fixing addition of total.</p> <p>If both errors fixed on line 03, full marks should be given. e.g. <code>total = total + scores[scoreCount-1]</code></p> <p><u>Examiner's Comments</u></p> <p>This proved to be a relatively challenging question for many candidates. This question relied on an understanding of the term "logic error".</p> <p>Two errors were present:</p> <ul style="list-style-type: none"> • line 02 (where the count-controlled loop ignored element 0 in the array) • line 03 (where the total was incorrectly calculated and stored). <p>Many responses identified at least one of the line numbers containing the error. Far fewer were able to successfully fix the errors satisfactorily.</p> <p>Examiners were instructed to be generous in interpreting potential fixes. The errors in either case could have been fixed using OCR Exam Reference Language (as presented) or using any other sensible form. Responses using programming syntax were credited, as were those who simply used English, e.g. "change the 1 to a 0 on line 02".</p> <p>Instructions were also included in the mark scheme to allow full marks for candidates who fixed the errors in one step on line 03.</p>
	Total	6	

Mark Scheme

Question		Answer/Indicative content	Marks	Guidance
3	a	<p>1 mark each</p> <ul style="list-style-type: none"> • stores/holds data/value/name/names [pos] • ...so (value) can be changed / swapped / moved / overwritten / inserted • ...without being lost. • will be assigned to <code>names [pos-1]</code> 	<p>2 (AO2 1b)</p>	<p>Do not allow answers that clearly refer storing the <u>position / index</u> (or any other out of context data) for BP1; it is the name itself that is being stored, not the position. If unclear, allow BOD.</p> <p>e.g. do not allow “holds the values of the index / holds value for position of the name”.</p> <p>Allow FT for subsequent points.</p> <p><u>Examiner’s Comments</u></p> <p>This question was an excellent discriminator in terms of candidate achievement. Many responses correctly described the <code>temp</code> variable as storing the name so that it could be overwritten during the process of swapping values. Partial credit was given for responses simply commenting on it storing data, as this is essentially the purpose of any variable.</p> <p>However, credit was not given where candidates discussed storing a position/index, as this was not the case; the <code>pos</code> variable is used as the index of the array. The content of this array index (a name) is stored in the <code>temp</code> variable.</p> <p>Where responses are open to interpretation, the mark scheme attempts to credit as many of these as possible. However, incorrect responses (such as reference to the position here instead of the name) are not credited.</p>
	b	<p>1 mark</p> <ul style="list-style-type: none"> • do not know how many iterations / swaps needed • do not know (at run time) how many times the value will change positions • do not know how many times a condition-controlled loop will need to run / execute <p>1 mark</p> <ul style="list-style-type: none"> • condition controlled loops run while/until a condition is true / is false / until a condition is met • repeats while value in <code>[pos-1]</code> is larger than value in <code>[pos]</code> // while (further) 	<p>2 (AO2 1b)</p>	<p>Max 1 from each section, 2 marks total.</p> <p>Do not allow “while names are in the wrong order”.</p> <p>BP4 must have reference to <u>checking</u> a condition / condition being met, not just having a condition.</p> <p><u>Examiner’s Comments</u></p> <p>Previous questions such as Question 2(a) assessed knowledge (AO1). This question focused on the application (AO2) of a technique (condition-controlled loops) to the algorithm given (an insertion sort). This assessed candidates’ understanding of the</p>

Mark Scheme

Question	Answer/Indicative content	Marks	Guidance
	<p>swap needed</p> <ul style="list-style-type: none"> • will swap value until in correct position // will swap whilst in incorrect position • More efficient than / does not need to iterate as many times as count controlled / for loop 		<p>process an insertion sort follows to sort values.</p> <p>The J277 specification is clear that candidates do not have to remember the code for this algorithm. But the specification does state that candidates must be able to understand the main steps of the algorithm and identify the algorithm if given the code for it. In this case, candidates were given all of the code for the algorithm.</p> <p>Candidates generally found this question challenging. Many responses simply repeated the question and discussed sorting values. More successful responses understood that the inner loop is the part of the algorithm that moves the <code>name</code> repeatedly in the list and that we do not know how many times this move needs to be made. Further to this, it is the condition that the <code>name</code> is in the correct position (or even better, an explanation of how this is decided on) which ends the loop.</p> <div style="text-align: center;">  <p>Assessment for learning</p> </div> <p>When asked to explain why a particular technique is used, it is often useful for candidates to think about why alternative options have not been used. In this question, thinking about what would happen if a count-controlled loop had been used as the inner loop may have given candidates the insight to discuss why a condition-controlled loop has been used.</p>

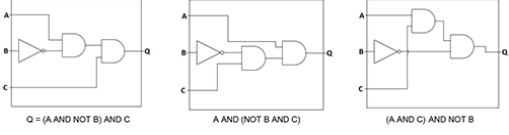
Mark Scheme

Question		Answer/Indicative content	Marks	Guidance
	c i	<p>1 mark each for insertion and bubble sort, max 2</p> <p>Insertion sort:</p> <ul style="list-style-type: none"> • inserts/moves values into correct position • inserts value once (then in correct position) • stops when end of array reached // completes in one pass through the array • moves items down the array / left • start of array becomes sorted first • creates a sorted array within an array // has a sorted/unsorted partition / section / list • starts on 2nd value • more efficient/faster than bubble sort • ... because fewer iterations / comparisons (on average) • ... when data more scrambled <p>Bubble sort :</p> <ul style="list-style-type: none"> • compares/swaps pairs of values • value is repeatedly moved/swapped (until in correct position) • repeats if a swap has been made // needs multiple passes • will complete a final iteration once sorted (to check for no swaps needed) • moves items up the array • end of array becomes sorted first • moves/bubbles the highest value to the top • less efficient/slower than insertion sort (on large sets of values) • ... more iterations / comparisons (on average) • ... when data more scrambled 	<p>2 (AO1 1b)</p>	<p>Answer must reference both bubble sort and insertion sort for 2 marks except if efficiency mark plus expansion given.</p> <p>Allow reference to big O for efficiency discussion.</p> <p>Only award efficiency once. Only award fewer iterations once</p> <p>Do not accept “completes in one iteration” for insertion sort.</p> <p>Accept list / data / values / etc for array.</p> <p>“when data more scrambled” only makes sense when discussing efficiency/speed, do not give marks for saying that either can handle data that is more scrambled (they both can sort data however it is arranged).</p> <p>Do not accept “bubble/insertion sort does not” for 2nd mark.</p> <p><u>Examiner’s Comments</u></p> <p>Question 3(c)(i) and Question 3(c)(ii) asked candidates to describe one difference and two similarities between an insertion sort and a bubble sort.</p> <p>Examiners were instructed to be generous in their interpretation of the requirements of both algorithms. This included considering both their algorithmic implementation and the understanding of how these are typically described on a classroom whiteboard.</p> <p>As such, many combinations of answers could gain the marks available.</p>

Mark Scheme

Question		Answer/Indicative content	Marks	Guidance
	ii	<p>1 mark each to max 2 e.g.</p> <ul style="list-style-type: none"> • Both produce a sorted list / array • Both work in place / without duplicating data / without using divide and conquer • Both need a temporary variable • Both swap values • Both use loops / iteration / repeats • Both loops are nested / inside each other • Both (may) need multiple passes • Both use selection • Both work with an array / list data structure • Both work from left to right • Both build up sorted list one item at a time (after every pass) • Both compare (pairs of) values • Both (typically) less efficient / slower than merge sort (or other sorting algorithms) • Both inefficient / slow for larger / unsorted lists // efficient for small / (nearly) sorted lists • Both start by comparing first two values 	<p>2 (AO1 1b)</p>	<p>Allow reference to both sorting / putting items into order for BP1.</p> <p>“Allows sorting of numbers and strings” meets BP1</p> <p>Allow answers relating to not needing additional memory as BP2.</p> <p>Allow “breaking into smaller lists” as divide and conquer for BP2.</p> <p>If answer is a statement (e.g. “uses loops”), assume candidate is talking about both algorithms doing this.</p> <p><u>Examiner’s Comments</u></p> <p>Question 3(c)(i) and Question 3(c)(ii) asked candidates to describe one difference and two similarities between an insertion sort and a bubble sort.</p> <p>Examiners were instructed to be generous in their interpretation of the requirements of both algorithms. This included considering both their algorithmic implementation and the understanding of how these are typically described on a classroom whiteboard.</p> <p>As such, many combinations of answers could gain the marks available.</p>
		Total	8	

Mark Scheme

Question	Answer/Indicative content	Marks	Guidance
4 a	<p>1 mark each, max 2 if not fully correct circuit.</p> <ul style="list-style-type: none"> • NOT B • AND gate with A / C as one direct input... • ...Second AND gate with other (unused) A / C as direct input and output of previous stage as other input <p>Fully correct circuit is any of :</p> <ul style="list-style-type: none"> • $Q = (A \text{ AND NOT } B) \text{ AND } C$ • $Q = A \text{ AND } (\text{NOT } B \text{ AND } C)$ • $Q = (A \text{ AND } C) \text{ AND NOT } B$ <p>See examples below :</p>  <p style="font-size: small;"> $Q = (A \text{ AND NOT } B) \text{ AND } C$ $A \text{ AND } (\text{NOT } B \text{ AND } C)$ $(A \text{ AND } C) \text{ AND NOT } B$ </p>	3 (AO3 2a)	<p>Shapes of logic gates must be correct. NOT gate must include circle for inversion. No other gates should include circle.</p> <p>AND gates must have two different inputs, NOT gate must have one input. All gates must have one output.</p> <p>Correct system will always have NOT B and two other AND gates correctly joined.</p> <p>Accept alternative systems that produce the correct output.</p> <p>Accept (BOD) three input AND gate for BP2 and BP3 if used correctly.</p> <p>OK if inputs/outputs not joined up to A/B/C/Q as long as intention clear.</p> <p>If lines cross on diagram, give BOD.</p> <p>If $(A \text{ AND } C) \text{ AND NOT } B$ drawn, allow NOT B as first input for BP3.</p> <p><u>Examiner's Comments</u></p> <p>This logic question is based around a typical outdoor light system. The majority of candidates were able to show understanding that both the switch (input C) and the motion sensor (input A) need to be triggered for the light (output Q) to be on. However, some candidates missed the description that this only occurs at nighttime. Input B is the light sensor and so a NOT gate is required on this input to fulfil this requirement.</p> <p>Given that AND gates generally have two inputs and this system uses three inputs, two AND gates were required. The mark scheme credits any and all arrangements of these to reach the correct output.</p> <p>Examiners were also instructed to credit use of a 3 input AND gate.</p>

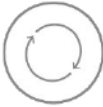
Mark Scheme

Question		Answer/Indicative content	Marks	Guidance
	b	1 mark each <ul style="list-style-type: none"> • Logic gate 1: OR • Logic gate 2: AND 	2 (AO2 1a)	Allow A OR B // B OR A for logic gate 1 Allow A AND B // B AND A for logic gate 2 If logic statement provided with multiple gates (e.g. A OR B AND C) this is incorrect. Allow use of symbols (e.g. \vee , + for OR, \wedge , \cdot for AND) Allow correct drawing of logic gates.
		Total	5	

Mark Scheme

Question			Answer/Indicative content	Marks	Guidance
5	a	i	<p>1 mark each to max 2</p> <ul style="list-style-type: none"> • Check the program works (as intended) • meets user requirements. • gives the correct output / result • Find / detect / check for errors / bugs • Check the program does not crash // is robust // executes / runs • To try and break the program // destructive testing • Test for / improve usability / user experience / performance // check user feedback is suitable • Allow any errors to be fixed // make changes / improvements as a result of testing • Ensure no problems / issues fixed when released. • Defensive design considerations / anticipating misuse / so cannot be misused 	<p>2 (AO1 1b)</p>	<p>Allow answers that explain what would happen if not tested (e.g. “there might be bugs”)</p> <p>Examiner’s Comments</p> <p>Testing as a process could be done for many reasons. The mark scheme attempts to credit as many sensible explanations for this as possible. This includes testing to find errors, checking for robustness and checking against user requirements. Furthermore, fixing errors that are found is also credited as this could form part of the testing process.</p> <p>The majority of responses demonstrated an understanding of this and achieved highly.</p>
		ii	<p>1 mark for name, 1 mark for matching description</p> <p>e.g.</p> <ul style="list-style-type: none"> • Final / terminal testing... • ... Completed at the end of development / before release. • ... to test the product as a whole. • Iterative / incremental testing... • ...completed during development. • ...after each module is completed. • ... test module in isolation • Normal testing... • ...test using data that should be accepted // • ...test that is expected to work / pass • Boundary / Extreme testing... • ...test using data that is on the edge of being acceptable / unacceptable. • ...test highest / lowest value • Invalid / Erroneous testing... • ...test using data that should be rejected / is not acceptable / causes an error 	<p>2 (1 AO1 1a) (1 AO2 1b)</p>	<p>Allow other sensible descriptive names for testing.</p> <p>Description must match test type.</p> <p>Must be a description and not just an example, but example may support description.</p> <p>Do not accept descriptions that simply repeat type of test without further clarification (e.g. “boundary, testing the boundary”).</p> <p>Allow :</p> <ul style="list-style-type: none"> • Black box testing... • ...testing without access / knowledge of a system’s workings. • White box testing... • ...testing with access / knowledge of system’s workings. <p>Allow other sensible / valid types of testing.</p> <p>Do not accept examples of validation (e.g. type test, range check)</p> <p>“data that is not expected” is NE for invalid/erroneous unless clarified.</p>

Mark Scheme

Question	Answer/Indicative content	Marks	Guidance
			<p><u>Examiner's Comments</u></p> <p>The J277 specification lists iterative and final/terminal testing as test types. However, many candidates interpreted this question as asking about test data (such as normal or erroneous data). Where candidates described the use of test data and link it to expected outcomes, this was credited by examiners.</p> <p>Other types of suitable testing that do not appear on the J277 specification (such as white box/black box testing, alpha/beta testing) were also accepted.</p> <div style="display: flex; align-items: center; margin: 10px 0;">  <p>Assessment for learning</p> </div> <p>The J277 specification states the minimum content that candidates are expected to know and understand at GCSE level. However, it is possible for teachers to go beyond this.</p> <p>For example, iterative and final/terminal testing are stated in the specification. However, there are other types of testing. Other technically correct responses will be accepted by examiners even if they do not appear in this specification.</p> <p>Another example is sorting algorithms. Merge sort, bubble sort and insertion sort appear in the specification. However, when candidates have been asked to name sorting algorithms, previous mark schemes credit other valid responses (such as quick sort, selection sort or bogo sort).</p>
iii	<p>1 mark for feature 1 mark for matching description e.g.</p> <ul style="list-style-type: none"> • Translator / compiler / interpreter ... • ... convert to low-level/machine code • ...allow program to be executed / run • ...produce executable file (only for compiler) • ...stops execution when error found (interpreter only) 	<p>4 (AO2 1b)</p>	<p>Allow other sensible names for features.</p> <p>Description must add more than is given in the identification of the feature to be awarded. For example, "keyword highlighting, highlights keywords" is 1 mark for the feature only.</p> <p>If compiler and interpreter given as two distinct features, allow both (with suitable descriptions). Do not allow translator and compiler/interpreter.</p>


Mark Scheme

Question	Answer/Indicative content	Marks	Guidance
	<ul style="list-style-type: none"> • Run-time environment / output window... • ...allows program / code to be run / executed • ...shows output of the program / code • Error reporting / diagnostics • ... identify location/detail of errors • ...suggests fixes • Debugger ... • ...find errors • Stepping ... • ... execute/run the program line by line • Variable watch... • ... see the contents/data held in variables • Break points ... • ... will allow the program to stop at a chosen / set position • Text/code editor... • ...allows program code to be written / entered / changed • ...allows errors to be fixed • Pretty printing // keyword highlighting... • ... allows keywords / variables to be coloured / identified • Keyword completion // syntax suggestion... • ...suggests code/syntax when first part entered. 		<p>Description must match feature.</p> <p>“finding errors” is NE for description of error reporting.</p> <p>Allow sensible references to AI where appropriate. Sensible description of use needed.</p> <p>Allow other sensible features of an IDE (e.g. line numbering, auto indent, collapsed blocks, etc) with suitable description.</p> <p>For text editor / error diagnostics / debugger, allow other sensible features listed as features in the mark scheme as description (e.g. “text editor, provides pretty printing”, “debugger, provides stepping”)</p> <p><u>Examiner’s Comments</u></p> <p>This question was generally well answered with a variety of features given. The question specifically asks about features used when testing a program. Therefore, features such as debugging tools, stepping and variable watch windows were very common responses.</p> <p>However, more general responses were also accepted, such as text editors, translators, and keyword completion. These could all potentially be used when editing programs after errors had been identified.</p> <p>Less successful responses tended to be descriptions that simply repeated the name of the feature given. For example “debugging tools, to allow debugging” would gain 1 mark for the feature identification but not the description.</p>

Mark Scheme

Question	Answer/Indicative content	Marks	Guidance
b	<p>1 mark for method, 1 mark to max 2 for each use</p> <p>e.g.</p> <ul style="list-style-type: none"> • Range check • ... checks upper/max / lower/min / boundaries • ... make sure the players answer / input is between sensible limits (e.g. 20 or less, between 2 and 20 inclusive) // not negative • ...by example of program code • Type check • ... making sure the data inputted is of the correct data type • ... make sure answer / input is an integer (or equivalent e.g. whole number) • Presence check • ... making sure a value is inputted / not blank • ... reference to answer / input • ...by example of program code • Length check • ... limit number of characters // check maximum / minimum string length • ... answer / input must be 1 or 2 characters • ...by example of program code • Format check • ... making sure the data inputted follows a set pattern • ... checking answer / input consists of only 1 or 2 numeric digits • ...by example of program code • Look up / table check • ... making sure the data inputted is one from an allowed set of values • ... checking that answer / input is one of [2, 3...20] inclusive • ...by example of program code 	<p style="text-align: center;">6</p> <p>(4 AO2 1b)</p> <p>(2 AO1 1a)</p>	<p>Validation must be applied to the rules of the game as given; do not accept uses related to input not asked for (e.g. names, passwords, etc).</p> <p>Do not accept uses that simply repeat the name of the check (e.g. “range check, checks a range of numbers”)</p> <p>For range check, values must be sensible (e.g. 1 to 50), and allow input of 2 to 20. Do not allow 1 / 10 (answer could be over this).</p> <p>For length check, must be clear that the string version of the data input is being checked to award use marks (e.g. characters not digits).</p> <p>Accept alternative names or descriptions (e.g. existence check, boundary check) but name of check must be sensible.</p> <p>Mark each answer as a whole, ignore method/use headings.</p> <p>Do not accept defensive design elements (e.g. input sanitisation, authentication)</p> <p>Examples of program code can be actual code (e.g. if inp>=2 and inp<=20) or identification of technique (e.g. “use IF statement / Case statement to limit values to between 1 and 20”). Do not accept code just showing casting.</p> <p><u>Examiner’s Comments</u></p> <p>Responses which focused on the explicit link to the game described in this question tended to do well.</p> <p>The stronger responses stated a validation method and linked the use of the validation method to the game. For example, a requirement of the game is that two random numbers between 1 and 10 are picked. It is sensible to suggest that validation ensuring the total is between 2 and 20 could be implemented. Further discussion relating to how this could be done, even as far as suggesting sensible high-level code that could be used would have developed the response.</p>

Mark Scheme

Question	Answer/Indicative content	Marks	Guidance
			<p>Some candidates gave examples of validation which did not clearly link to the game. Generic examples were partially credited. Explicit links to the game were required in order to gain all marks available.</p> <p style="text-align: center;">  Misconception </p> <p>Input validation applies to values input by the user. In this case, the requested input is the sum of two numbers, each of which are between 1 and 10. It is not necessary to validate the random number generation (as this has not been input) and it would be inappropriate to limit user inputs to between 1 and 10; the total could easily be (for example) $8 + 6 = 14$.</p> <p>Where candidates suggested validating inputs to allow between 1 and 10, not all marks available were given due to this misconception.</p>

Mark Scheme

Question	Answer/Indicative content	Marks	Guidance
c	<p>1 mark each to max 6</p> <ul style="list-style-type: none"> • Initialise / declare <code>score</code> (to zero) before use, outside of any loop • Generates 2 random numbers <u>between 1 and 10</u> • Inputs answer from user displaying suitable numbers • Checks if input is <u>correct answer...</u> • ... if correct adds 1 to <code>score</code> • Repeats BP2 to 5 three times (for bullet points attempted) • Outputs <code>score</code> <u>after reasonable attempt at counting</u> 	<p style="text-align: center;">6 (AO3 2b)</p>	<p>No need to cast data to string/integer.</p> <p>If random numbers chosen, BP3 must use these. If no random numbers chosen, allow manually setting values</p> <p>BP6 can be awarded for either a loop repeating 3 times or the same code written out 3 times</p> <p>BP5 can be given FT if sensible attempt at BP4</p> <p>Do not award BP6 if same numbers used for every question. Must pick new values each time.</p> <p>Do not penalise potential off by 1 errors for looping (Python) or random number generation</p> <p><u>Example answer</u></p> <pre>score = 0 for count = 1 to 3 num1 = random(1, 10) num2 = random(1, 10) ans = input("What is" + num1 + " + " + num2 + "?") if ans = num1 + num2 then score = score + 1 end if next count print("You scored " + score)</pre> <p><u>Examiner's Comments</u></p> <p>As this question appears in Section A, candidates are free to respond in any suitable way, including using flowcharts, structured English, pseudocode or a high-level language.</p> <p>The majority of high scoring responses used a high-level language consistently.</p> <p>Where flowcharts or structured English were used, responses needed to clearly show the steps to be taken and not simply repeat the question to achieve marks.</p> <p>The given question is already decomposed for candidates and many were able to use these bullet points to build a solution that achieved the majority</p>

Mark Scheme

Question	Answer/Indicative content	Marks	Guidance
			<p>of marks available.</p> <p>Many responses used random number generation and iteration to create an elegant response that met all mark points. This was pleasing to see and it is extremely encouraging that candidates can use techniques such as these where appropriate without being prompted.</p> <p>Other responses manually repeated asking the required questions; on this occasion, these were also credited and could have achieved full marks.</p> <p>Where a mistake was made in one section (such as with iteration), examiners were instructed to use FT (follow through) where possible. This allowed candidates to score marks in later sections if their responses were logically constructed. This is to be fair to candidates so that mistakes are only penalised once in any given question.</p> <p>A significant number of responses did not access many marks in this question. This would suggest that more practical programming time in lessons would be beneficial.</p>
	Total	20	


Mark Scheme

Question		Answer/Indicative content	Marks	Guidance																														
6	a	<p>1 mark for each row</p> <table border="1" style="margin-left: auto; margin-right: auto; border-collapse: collapse; text-align: center;"> <thead> <tr> <th style="padding: 5px;">Variable</th> <th style="padding: 5px;">Boolea n</th> <th style="padding: 5px;">Ch ar</th> <th style="padding: 5px;">Str ing</th> <th style="padding: 5px;">Integ er</th> <th style="padding: 5px;">Re al</th> </tr> </thead> <tbody> <tr> <td style="padding: 5px;">UserName</td> <td></td> <td></td> <td style="text-align: center;">✓</td> <td></td> <td></td> </tr> <tr> <td style="padding: 5px;">EmergencyP honeNumber</td> <td></td> <td></td> <td style="text-align: center;">✓</td> <td></td> <td></td> </tr> <tr> <td style="padding: 5px;">DoorSensor Active</td> <td style="text-align: center;">✓</td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td style="padding: 5px;">DoorActive Time</td> <td></td> <td></td> <td></td> <td style="text-align: center;">✓</td> <td></td> </tr> </tbody> </table>	Variable	Boolea n	Ch ar	Str ing	Integ er	Re al	UserName			✓			EmergencyP honeNumber			✓			DoorSensor Active	✓					DoorActive Time				✓		4 (AO3 2a)	<p>No mark if more than 1 tick on a row.</p> <p>Allow other indications of choice (e.g. cross) as long as clear.</p>
Variable	Boolea n	Ch ar	Str ing	Integ er	Re al																													
UserName			✓																															
EmergencyP honeNumber			✓																															
DoorSensor Active	✓																																	
DoorActive Time				✓																														
	b	<p>1 mark each:</p> <ul style="list-style-type: none"> Attempt at using selection / condition controlled loop Checking if system armed // while system armed If Door Sensor active OR Window Sensor active (both checks required) calling SoundAlarm correctly 	4 (AO3 2b)	<p>Selection could be done using IF statement, case statement or any other sensible valid method.</p> <p>Allow reference to AlarmActivated or equivalent instead of SystemArmed</p> <p>Ignore any inputs or modification of variables.</p> <p>Allow True / False as strings. Allow checking against strings (e.g. if SystemArmed == "active")</p> <p>Allow checking armed/disarmed for BP2 and BP3</p> <p>Only award BP4 if SoundAlarm correctly called / not called in every situation. If issues on previous lines (e.g. lack of brackets where needed) means this is not the case, do not award BP4.</p> <p>Checking could be done by evaluating variable directly (if SystemArmed) or by comparison (if SystemArmed == True)</p> <p><u>Example answer 1</u></p> <pre> if SystemArmed then if DoorSensorActive then SoundAlarm() else if WindowSensorActive then SoundAlarm() endif endif </pre>																														

Mark Scheme

Question	Answer/Indicative content	Marks	Guidance
			<p><u>Example answer 2</u></p> <pre>while SystemArmed then if DoorSensorActive then SoundAlarm() else if WindowSensorActive then SoundAlarm() endif endif</pre> <p><u>Example answer 3</u></p> <pre>if SystemArmed and (DoorSensorArmed or WindowSensor) then SoundAlarm() endif</pre> <p>Note – above example needs brackets,</p> <pre>if SystemArmed and DoorSensorArmed or WindowSensor then</pre> <p>is not logically valid for this scenario (will sound alarm when not armed if window sensor is active)</p> <p><u>Example answer 4</u></p> <pre>if SystemArmed and DoorSensorArmed SoundAlarm() else if SystemArmed and WindowSensorArmed SoundAlarm() endif</pre> <p><u>Examiner's Comments</u></p> <p>Many responses achieved highly on this question. The question asks for a simple program to be written that checks the given variables and calls the given procedure when necessary.</p> <p>Examiners were instructed to be generous with the first mark, crediting any use of selection or condition-controlled iteration. Responses may therefore have been rewarded for an attempt at this question even if their solution was not fully functional.</p> <p>Centres should encourage candidates to</p>

Mark Scheme

Question	Answer/Indicative content	Marks	Guidance
			<p>attempt each question for precisely this reason; it is typical that a small number of marks are allocated to attempting a solution on many programming questions for J277/02.</p> <p>A significant number of responses were given 3 out of 4 marks as they misunderstood the role of operator precedence in their solution; this is detailed in the "misconception" box below.</p> <div style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <p style="text-align: center;"> Misconception</p> <p>Where multiple conditions are used in selection, these have an order of precedence very much like BIDMAS does in mathematics; an AND operator will always take precedence over an OR operator. A NOT operator (not used in this question) would have even higher precedence.</p> <p>This can cause problems in candidate responses. A common candidate response was:</p> <pre style="font-family: monospace;">if SystemArmed AND DoorSensorActive OR WindowSensorActive then SoundAlarm()</pre> <p>However, because the AND operator takes precedence, the first check done here is if the system is armed and the door sensor is active. The result of this is then evaluated with an OR operator to check if the window sensor is active.</p> <p>This results in the alarm sounding if the window sensor is active, even if the system is not armed. This was clearly not the candidate's intention.</p> <p>To fix this, candidates could have either:</p> <ul style="list-style-type: none"> • put brackets/parentheses around the <code>Door OR Window</code> section of their response • written the response as separate checks. This could have been done in multiple ways, including nested <code>if</code> </div>

Mark Scheme

Question		Answer/Indicative content	Marks	Guidance
				<p>statements or repeated checks.</p> <p>Exemplar 1</p> <pre> if System Armed: if Door Sensor Active or Window Sensor Active: Sound Alarm()</pre> <p>Exemplar 1 shows one way that full marks are achieved on this question. The candidate has used nested <code>if</code> statements to check if the system is armed, and if true, then checking if either sensor has been activated. The <code>SoundAlarm()</code> procedure is only called if both <code>if</code> statements evaluate to True.</p>
	c	i	1 (AO3 1)	<p>Examiner's Comments</p> <p>Concatenation is the process of joining strings together. In OCR Exam Reference Language, this is done using the <code>+</code> symbol. Line 04 joins together <code>sensorType</code> and <code>sensorNumber</code>, assigning the concatenated result to the variable <code>sensorID</code>.</p>
		ii	1 (AO3 1)	<p>Do not penalise case, spacing or minor misspellings.</p> <p>Examiner's Comments</p> <p>The vast majority of candidates are confident with identifying variable identifiers, such as <code>sensorNumber</code>. Small errors in spelling or spacing were not penalised. Candidates should be encouraged to be accurate with their namings, particularly if these are already given to them in the question.</p>

Mark Scheme

Question	Answer/Indicative content	Marks	Guidance
	<p>iii</p> <p>1 mark for</p> <ul style="list-style-type: none"> • Boolean 	<p>1 (AO3 1)</p>	<p>Ignore minor misspelling.</p> <p>Accept Bool.</p> <p><u>Examiner's Comments</u></p> <p>Many candidates found this question challenging. The question relies on understanding of how selection statements operate. The answer of Boolean can be inferred from how the function return value is used.</p> <p>This use of Boolean values in selection statements also caused confusion in the previous J276 specification. It would be beneficial for centres to cover this specifically. More detail is given in the misconception box below.</p> <div style="border: 1px solid black; border-radius: 50%; width: 40px; height: 40px; display: flex; align-items: center; justify-content: center; margin: 10px auto;"> ? </div> <p>Misconception</p> <p>Where Boolean values are used in selection, there is no need to compare this to a <code>True</code> or <code>False</code> value. Line 02 uses an IF statement based on the return value from a function <code>CheckSensorCode()</code>. Because the function returns a Boolean value, the IF statement is valid. Looking at another example, if the variable <code>x</code> is Boolean, then both of the following would be valid:</p> <ul style="list-style-type: none"> • <code>if x == True then...</code> • <code>if x then...</code> <p>In both cases, if the value of <code>x</code> is <code>True</code>, the code underneath would be executed. The second version (without comparing a Boolean value to <code>True</code> or <code>False</code>) is more elegant.</p> <p>However, both would be accepted as responses in an examination.</p>

Mark Scheme

Question		Answer/Indicative content	Marks	Guidance
	iv	1 mark from <ul style="list-style-type: none"> • Line 01 • Line 02 • Line 03 • Line 05 	1 (AO3 1)	<p><u>Examiner's Comments</u></p> <p>The program code given contains two explicit calls to functions. There is one function call on line 02 for <code>CheckSensorCode()</code> and one function call on line 05 for <code>ResetSensor()</code>. However, many programming languages (including Python) implement input as a function and so examiners were instructed to also give credit where candidates identified lines 01 or 03.</p>
	v	1 mark each <ul style="list-style-type: none"> • Selection • Sequence 	2 (AO3 1)	<p>Ignore minor spelling errors / differences</p> <p>Do not accept examples (e.g. IF)</p> <p><u>Examiner's Comments</u></p> <p>The three programming constructs given in the specification are selection, sequence and iteration.</p> <p>Sequence and selection are used within this program.</p> <p>There is no use of iteration in the given program.</p> <p>Surprisingly for an AO1 question, this proved relatively challenging for candidates with many identifying other features of the code, such as inputs, function calls or variables.</p>

Mark Scheme

Question	Answer/Indicative content	Marks	Guidance
d	<p>1 mark each</p> <ul style="list-style-type: none"> • <code>SELECT SensorID // SELECT *</code> • <code>FROM events</code> • <code>WHERE Length > 20 AND sensorType = "Door"</code> <code>//</code> <code>WHERE sensorType = "Door"</code> <code>AND Length > 20</code> 	<p>3 (AO3 2c)</p>	<p>Max 2 if out of order or anything extra that affects the output.</p> <p>BP1 can select multiple fields as long as SensorID is included.</p> <p>Ignore case. Only penalise spaces if obvious.</p> <p>Field names must be correct.</p> <p>“door” must be in quotation marks for BP3. Allow quotation marks for field names and table name</p> <p>BP3 can use == or = for equivalence.</p> <p>Allow alternative WHERE clauses that are logically correct (e.g. <code>WHERE length >=21</code>)</p> <p><u>Examiner's Comments</u></p> <p>Structured Query Language is obviously well understood by many candidates. Many high-quality responses were produced.</p> <p>Most responses were able to use SELECT and FROM appropriately to produce a logically correct response. However, the vast majority of responses missed off the requirement that only door sensors were required to be included, gaining 2 out of 3 marks in the process.</p> <p>Although a suggested response is shown in the mark scheme, any logically correct SQL that produces the required output would be accepted by examiners.</p> <p>Where a mistake was made consistently (such as using colons after the SQL keyword), this was penalised once and then FT (follow through) allowed for subsequent marks.</p>

Mark Scheme

Question	Answer/Indicative content	Marks	Guidance
e	<p>1 mark each</p> <ul style="list-style-type: none"> • Define procedure SaveLogs... • ...with two valid parameters • Open file (for write/append) ... • ... using the file name passed in as parameter • Write data to file... • ...using the data passed in as parameter • Close file 	<p>6 (AO3 2b)</p>	<p>Must be clear that answer is a procedure definition, do not credit calling procedure for BP1. Allow function definition.</p> <p>If parameters are later overwritten, do not credit BP2 but FT for BP4 and 6.</p> <p>Closing text file does not need reference to file name/object – e.g. “close file” is enough. However, if given reference must be correct.</p> <p>If code given outside of procedure, do not give BP4 and BP6</p> <p>Allow FT for multiple occurrences of same mistake (e.g. not using filename correctly for open and close)</p> <p><u>Example answer</u></p> <pre> procedure SaveLogs(data, filename) logFile = open(filename) logFile.writeLine(data) logFile.close() endprocedure </pre> <p><u>Examiner’s Comments</u></p> <p>This question proved to be challenging for many candidates. The question combined defining a procedure with the use of text files.</p> <p>The tasks required were partially decomposed in the bullet points. A candidate attempting these in order would have achieved a significant number of marks.</p> <p>Candidates could also have achieved numerous marks for a partial solution (e.g. defining a procedure that didn't use text files or writing to a text file outside of a procedure) and the mark scheme was deliberately constructed to credit these responses.</p> <p>Full marks were often given where candidates appear to have had practical experience of these two techniques.</p> <p>Exemplar 2</p>

Mark Scheme

Question			Answer/Indicative content	Marks	Guidance
					<pre> procedure SaveLogs(data, fileName) file = open(fileName) file.writeLine(data) file.close() endprocedure </pre> <p>Exemplar 2 shows a response that scored full marks. The procedure has been defined with multiple parameters which are then used to open the file and to write the data. The candidate has also achieved the bullet point 7 on the mark scheme (closing the file) but this wasn't necessary in this case.</p>
	f	i	1 mark for: <ul style="list-style-type: none"> • Casting / cast 	1 (AO3 2a)	Accept type casting Do not accept conversion. Do not accept examples of casting. <p>Examiner's Comments</p> The use of the term "casting" to convert one data type to another is now well known and understood by candidates. This is given and referred to in the J277 specification and is essential knowledge.
		ii	1 mark each to max 6 <ul style="list-style-type: none"> • Input date and store in variable / use directly • Access all seven (indexes 0 to 6) events in array // loop for each event in array • Attempt at selection... • ...to compare date input against date <u>in array</u> (element 0) • ...adding <u>length</u> (element 3) <u>from array</u> to the total <u>if dates match</u>. • Outputting <u>calculated</u> total and date in appropriate message(s) <u>at the end</u> 	6 (AO3 2b)	BP2 can be achieved either by iteration accessing each event or manually repeating code to access each event. Must be 0 to 6, not 1 to 7. <p>Allow reference to <code>events</code> (table given) or <code>arrayEvents</code> (2D array) in answer as long as used consistently.</p> BP2 loop allow off by one errors (Python), looping to array length or array length – 1. Allow for each item in array or any other suitable loop. BP4 and BP5 allow array reference as either column major or row major. Output can either be once at the end or on every iteration, as long as it is output at the end. Only give output mark if attempt made to calculate total <u>within the algorithm</u> . Do not penalise capitalisation or minor misspellings of variable names. <p><u>Example answer 1</u></p>

Mark Scheme

Question	Answer/Indicative content	Marks	Guidance
			<pre> total = 0 date = input("Please enter date") for count = 0 to events.length-1 if events[0, count] == date then total = total + events[3,count] endif next count print("There were " + total + " events on " + date) </pre> <p><u>Example answer 2</u></p> <pre> total = 0 date = input("Please enter date") for item in events: if item[0] == date then total = total + item[3] endif next count print("There were " + total + " events on " + date) </pre> <p><u>Examiner's Comments</u></p> <p>The final question in Section B is expected to be a high demand question.</p> <p>The techniques required (iteration through a 2D array, selection, keeping a running total of times) are within the specification but it is acknowledged that the level of challenge was high.</p> <p>Examiners were instructed to give marks for an attempt at a solution (as with previous questions).</p> <p>For this question marks were given for:</p> <ul style="list-style-type: none"> • any attempt at selection • any solution that accessed each element in the given array, even if this was via a manual process. <p>Therefore, many candidates gained multiple marks for an attempt that only partially solved the problem.</p> <p>A significant number of candidates were able to create a solution that fully met the</p>

Mark Scheme

Question	Answer/Indicative content	Marks	Guidance
			<p>requirements of the question. This was often done in an elegant and efficient manner.</p> <p>This is extremely pleasing and shows excellent understanding and significant experience of practical programming.</p> <p>Exemplar 3</p> <pre> total = 0 date = input("Enter a date.") total count = 0 for count = 0 to arrayEvents.length if arrayEvents[0, count] == date then total = total + arrayEvents [3, count] endif endfor print("Sensors were activated for", total, "Seconds on", date) </pre> <p>Exemplar 3 shows a high scoring response. A date has been asked for as input which has then been used to compare to each element at position 0 in the array.</p> <p>Where any of these match, the total variable is updated to keep a running total of the corresponding element at position 3 in the array.</p> <p>After each element has been checked, the total and date are output in a suitable message.</p> <p>This is not the only method by which a response could be given full marks but is perhaps the most common.</p>
	Total	30	