GCSE (9–1)

Computer Science

J276/02: Computational thinking, algorithms and programming

General Certificate of Secondary Education

Mark Scheme for Autumn 2021

Oxford Cambridge and RSA Examinations

OCR (Oxford Cambridge and RSA) is a leading UK awarding body, providing a wide range of qualifications to meet the needs of candidates of all ages and abilities. OCR qualifications include AS/A Levels, Diplomas, GCSEs, Cambridge Nationals, Cambridge Technicals, Functional Skills, Key Skills, Entry Level qualifications, NVQs and vocational qualifications in areas such as IT, business, languages, teaching/training, administration and secretarial skills.

It is also responsible for developing new specifications to meet national requirements and the needs of students and teachers. OCR is a not-for-profit organisation; any surplus made is invested back into the establishment to help towards the development of qualifications and support, which keep pace with the changing needs of today's society.

This mark scheme is published as an aid to teachers and students, to indicate the requirements of the examination. It shows the basis on which marks were awarded by examiners. It does not indicate the details of the discussions which took place at an examiners' meeting before marking commenced.

All examiners are instructed that alternative correct answers and unexpected approaches in candidates' scripts must be given marks that fairly reflect the relevant knowledge and skills demonstrated.

Mark schemes should be read in conjunction with the published question papers and the report on the examination.

1. Annotations

Annotation	Meaning
✓	Tick
✗	Cross
BP	Blank Page – this annotation **must** be used on all blank pages within an answer booklet (structured or unstructured) and on each page of an additional object where there is no candidate response.
^	Omission mark
BOD	Benefit of doubt given
NBOD	Benefit of doubt **not** given
FT	Follow through
NAQ	Not answered question
REP	Repeat
/	Slash
SEEN	Seen
NE	Not enough
TV	Too vague

<table>
<tr><th colspan="3">Question</th><th>Answer</th><th>Mark</th><th>Guidance</th></tr>
<tr><td>1</td><td>(a)</td><td></td><td>
<table>
<tr><th></th><th>ASCII</th><th>Extended ASCII</th><th>Unicode</th></tr>
<tr><td>Can represent thousands of characters, including Russian and Chinese symbols.</td><td></td><td></td><td>✓</td></tr>
<tr><td>Can represent European characters such as ç or â.</td><td></td><td>✓</td><td>✓</td></tr>
<tr><td>Uses different character codes for upper case and lower-case letters.</td><td>✓</td><td>✓</td><td>✓</td></tr>
</table>
</td><td>3</td><td>1 mark per row</td></tr>
<tr><td></td><td>(b)</td><td></td><td>• 1000101 (E)
• 1001000 (H)</td><td>2</td><td>Ignore leading zeros</td></tr>
<tr><td></td><td>(c)</td><td>(i)</td><td>• The <b>height</b> / <b>amplitude</b>…
• …as a numerical value
• …of the wave(form)
•</td><td>2</td><td><b>DO NOT</b> accept frequency
Do not accept "in binary" (given in question)</td></tr>
<tr><td></td><td></td><td>(ii)</td><td>• 48,000 <b>samples</b> taken…
• …per second</td><td>2</td><td>BOD How often samples are taken // frequency of samples</td></tr>
</table>

		(iii)	e.g. • Reduce the sample rate (from 48KHz to a lower rate) • …so fewer samples are taken per second • Reduce the bit depth (from 24 bits to a lower bit depth) • …so less data is used for each sample • Use lossy compression… • … to remove data (that won't be noticed) • Use lossless compression… • …to identify patterns in the data • …store this more efficiently • Reduce the length of the sound file … • … by example (from 30 seconds to a lower length) // less data to store	4	Any 4 points for 1 mark each Allow "compression" by itself for 1 mark if no other compression mark awarded. Allow suitable expansion of this for 1 mark. Do not accept "data is not lost" as expansion for lossless or "data is lost" as expansion for lossy.

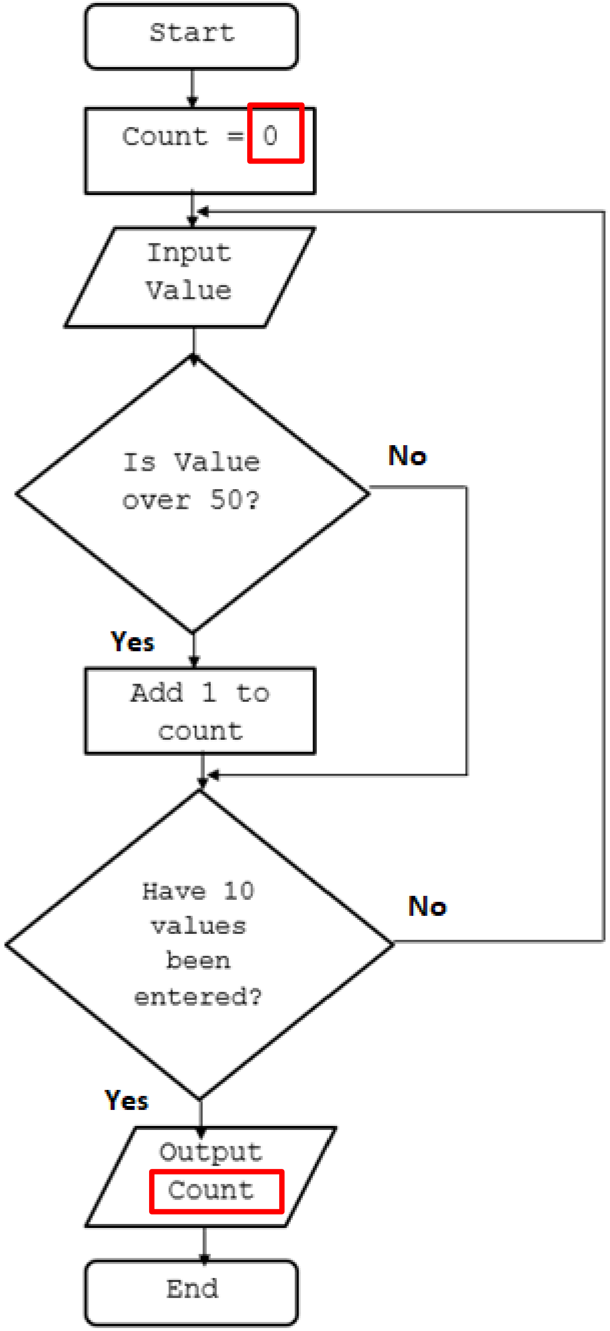
<table>
<tr><th colspan="3">Question</th><th>Answer</th><th>Mark</th><th>Guidance</th></tr>
<tr><td><b>2</b></td><td><b>(a)</b></td><td></td><td>
<table>
<tr><th>Statement</th><th>True (✓)</th><th>False (✓)</th></tr>
<tr><td>The list of words is initially split into a sorted set and an unsorted set</td><td>✓</td><td></td></tr>
<tr><td>The insertion sort uses a divide stage and then a conquer stage.</td><td></td><td>✓</td></tr>
<tr><td>The list of words must be in order before the insertion sort can start</td><td></td><td>✓</td></tr>
<tr><td>Each word is inserted into the correct place in the array, one by one</td><td>✓</td><td></td></tr>
<tr><td>The insertion sort will not work because the word “wall” appears twice.</td><td></td><td>✓</td></tr>
</table>
</td><td>1</td><td>1 mark per row</td></tr>
</table>

	(b)		• Pick middle value / pumpkin // find midpoint • Compare this to house, **no match** • pumpkin>house… • …so discard top half of list // focus on bottom half • Pick **middle value** again, either <u>house</u> or <u>flour</u>… • …finds value // repeat to find value	4	Do not award generic responses except for BP1 Must clearly show the steps taken for <u>this</u> list to achieve more than 1 mark. Do not award "splits the list in half" for BP1 or 4 – incorrect Allow diagrams to demonstrate the process Allow reasonable attempt at BP3 to allow access to BP4

Question			Answer	Mark	Guidance

3	**(a)**		• Initialises (total) as 0 (outside loop if present)
			• Inputs a number and stores the value
			• Adds the input **to the total** (initialised in BP1 if present)
			• Prints the total
			• Iterates over BP2-4 (if present)…
			• …until total is **over** 100
				6	Example answer
					`total = 0`
					`while total <=100`
					`    x = input("Enter a number")`
					`    total = total + x`
					`    print(total)`
					`endwhile`

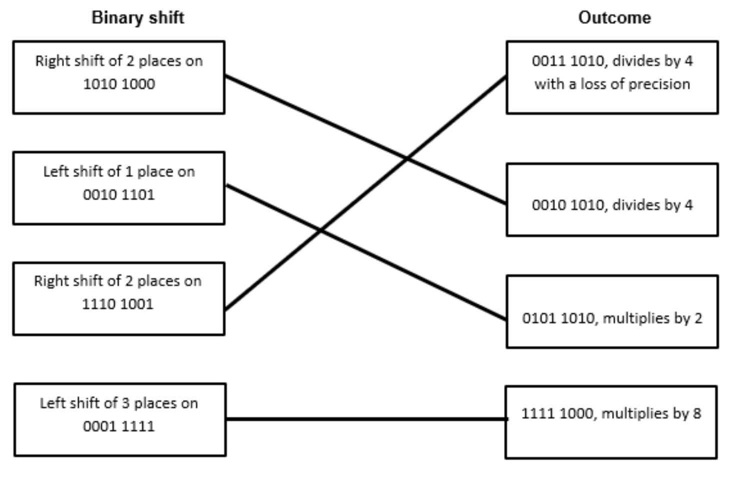
	(b)	**(i)**	• Number with a decimal / fractional part • Suitable example (e.g. 17.24)	2	One mark for definition, one mark for example Do not accept float as definition Allow fractions as example
		(ii)	• Whole number // number with no decimal / fractional part • Suitable example (e.g. 17)	2	One mark for definition, one mark for example

	(c)	**(i)**	• Count = **0** • Output **Count** • All non-decision boxes and YES from decision boxes linked in a sequential fashion from Start to End. • NO from first decision box linked to skip over increment of count • NO from second decision box linked back to INPUT Ignore superfluous instructions as long as they do not affect the outcome of the algorithm. BOD misspelling of `Count` as long as it is recognisable Ignore capitalisation.	5	 Start Count = 0 Input Value Is Value over 50? No Yes Add 1 to count Have 10 values been entered? No Yes Output Count End

		(ii)	1 mark per bullet point, max 5
			• Initialises a count variable to 0
			• asks user for an input
			• Check if input is over 50…
			• … increment count variable if True
			• Repeats BP 2 and 3 (if present) until 10 numbers have been entered
			• Outputs count once 10 numbers have been entered
				5	Example answer
					
					`count = 0`
					`for x = 1 to 10`
					`    value = input("enter a value")`
					`    if value > 50 then`
					`        count = count + 1`
					`    endif`
					`next x`
					`print(count)`
					
					Response must be in pseudocode as per question, flowcharts or structured English are NAQ.

	(d)		e.g. • Abstraction • … focussing on the important elements // ignoring elements that do not contribute to the solution // simplifying the problem • Decomposition • …breaking a problem down (into its constituent parts) • Algorithmic thinking • …set out the steps needed to solve the problem // represented in a flow chart / as pseudocode	4	Mark in pairs. 1 mark for name, 1 mark for description. Description must match technique (if given).

Question			Answer	Mark	Guidance
4	(a)		• Contents of variable **can** be changed; contents of constants **cannot** be changed (while the programming is running)	1	Both sides needed for mark.
	(b)	(i)	• 16	1	
		(ii)	• 2	1	
		(iii)	• 9	1	
	(c)	(i)	• second.substring(3,5)	1	Ignore print / lack of print. Allow other suitable methods of string manipulation as long as variables used. Allow any valid method that extracts rightmost 5 or 6 characters of `second` variable.
		(ii)	• first.substring(0,8)	1	Ignore print / lack of print. Allow other suitable methods of string manipulation as long as variables used. Allow any valid method that extracts leftmost 8 or 9 characters of `first` variable.
		(iii)	• first.substring(9,7) + " " + second • "Science " + second • first.substring(9,7) + " is great"	1	Ignore print / lack of print. Allow other suitable methods of string manipulation as long as variables(s) used. Allow alternative concatenation symbols (e.g. & or **.**). Allow concatenation functions Must have correct spacing in outcome.

Question			Answer	Mark	Guidance
5	(a)		1011 0010	2	1 mark per nibble. Mark right to left. Must be 8 bits (as per question)
	(b)		• Transistor has two **states** • 1 represents on, 0 represents off • Each transistor stores one bit • Multiple transistors used to store a binary value	2	Allow values for BP1
	(c)		C7	2	1 mark per hex digit, mark from right to left. Max 1 mark if more than 2 characters given.
	(d)		• Incorrect ticked • Data cannot be stored in hexadecimal // all data is stored in binary // hexadecimal is a shortcut for computer scientists	2	1 mark for identifying issue, 1 mark for reason why. Allow FT for BP2 if candidate agrees but provides further clarification that shows they understand.
	(e)		 Binary shift Outcome Right shift of 2 places on 1010 1000 Left shift of 1 place on 0010 1101 Right shift of 2 places on 1110 1001 Left shift of 3 places on 0001 1111 0011 1010, divides by 4 with a loss of precision 0010 1010, divides by 4 0101 1010, multiplies by 2 1111 1000, multiplies by 8	3	3 marks for all connections correctly made 2 marks for 2 or 3 connections correctly made 1 mark for any connection correctly made
	(f)		1100 1100	2	1 mark per nibble. Each pair of nibbles in question can be added individually so no requirement for FT marks.

<table>
<tr><th colspan="3">Question</th><th>Answer</th><th>Mark</th><th>Guidance</th></tr>
<tr><td>6</td><td>(a)</td><td></td><td><table><tr><th>Function call</th><th>Returned value</th></tr><tr><td>checkblock(2,1)</td><td><b>B</b></td></tr><tr><td>checkblock(3,0)</td><td><b>A</b></td></tr><tr><td>checkblock(2,3)</td><td><b>FREE</b></td></tr></table></td><td>3</td><td>Do not accept “blank” or any other returned value for third call.

Ignore case and spelling as long as recognisable.</td></tr>
<tr><td></td><td>(b)</td><td></td><td>• <b>Returns</b> a value // passes back a value</td><td>1</td><td></td></tr>
<tr><td></td><td>(c)</td><td>(i)</td><td>• Parameter values outside index range / larger than 4 / smaller than 0 // -1, 16 is not a valid block</td><td>1</td><td>Answer must refer to either array or gameboard / grid / block</td></tr>
<tr><td></td><td></td><td>(ii)</td><td>• Use selection / IF / Switch-Case / range check
• …check that <b>parameters</b> are >=0 <b>and</b> <= 4…
• <b>…Return</b> error code if invalid // set outcome to error</td><td>3</td><td>Allow equivalent checks (e.g. <5, between 0 and 4) for BP2
Allow reference to r and c as parameters.
BOD handle error for BP3 (e.g. repeat until valid)
Answer must be a description, code by itself is NAQ</td></tr>
<tr><td></td><td>(d)</td><td></td><td>• Input two position values separately
• calls checkblock() function…
• …with input parameters
• … returned value used in selection
• If free, stores “A” to correct index of gamegrid array (FT for incorrect selection)
• Loops until free position chosen</td><td>6</td><td>If flowchart / structured English, do not allow simple repeat of question.
<u>Example answer</u>
loop = True
while loop
    row = input("enter row")
    col = input("enter column")
    if checkblock(row,col) == "FREE" then
        gamegrid[row,col] = "A"
        loop = False
    endif
endwhile</td></tr>
</table>

OCR (Oxford Cambridge and RSA Examinations)
The Triangle Building
Shaftesbury Road
Cambridge
CB2 8EA

OCR Customer Contact Centre

Education and Learning
Telephone: 01223 553998
Facsimile: 01223 552627
Email: general.qualifications@ocr.org.uk

www.ocr.org.uk

For staff training purposes and as part of our quality assurance programme your call may be recorded or monitored

Cambridge
Assessment