OCR
Oxford Cambridge and RSA

GCSE (9-1)

Computer science

Unit **J276/02**: Algorithms and programming

General Certificate of Secondary Education

Mark Scheme for June 2018

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.

© OCR 2018

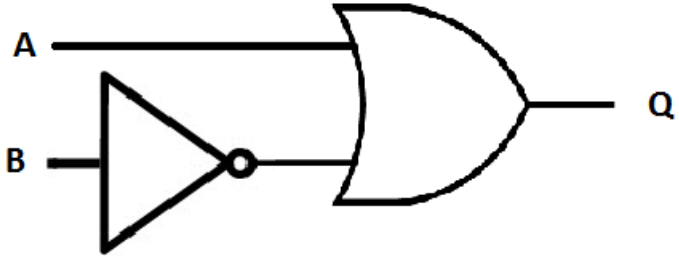
Annotations

Annotation	Meaning
BP	Blank page
	Highlight
	Off page comment
^	Omission mark
BOD	Benefit of doubt
	Cross
FT	Follow through
NAQ	Not answered question
NBOD	Benefit of doubt not given
REP	Repeat
	Slash
	Tick
TV	Too vague
0	Zero (big)
SEEN	Noted but no credit given

Question			Answer	Mark	Guidance
1	(a)		1 mark per bullet, max 3 • String • Integer / Int • Boolean	3	Accept text / varchar for string. Do not accept character. Do not accept number/numeric for integer Accept yes/no, true/false for Boolean.
1	(b)	(i)	1 mark per bullet, max 2 if not in correct order or additional statements given. • SELECT StudentName • FROM conduct • WHERE Points < 0	3	Capitalisation does not affect the mark. Spellings of fields, tables must be correct. Ignore brackets. Ignore quotes around StudentName, Conduct or Points. Mark quotes around 0 in WHERE clause as incorrect. StudentName must not include space Accept <= -1 or equivalent for 3rd bullet point.
1	(b)	(ii)	• * / star / asterisk	1	Wildcard (*) must be clearly identified as the answer. Do not allow any other SQL statements alongside this unless this is given as an example.

1	(c)		1 mark per bullet, max 4
			
			• Selection(IF) used
			• Comparing studentdata[3]…
			• …with “TRUE” or “FALSE” // TRUE or FALSE
			• Correct outputs (“sent” **and** “not sent”)
				4	Example algorithm
					
					`if studentdata[3] == "TRUE" then`
					`  print "sent"`
					`else`
					`  print "not sent"`
					`end if`
					
					Bullet point 3 can only be awarded If an attempt is made at identifying `studentdata` (e.g. with the wrong index or no index). Do not allow simply comparing anything with True / False.
					
					Bullet point 3 can be implicit.

Question			Answer	Mark	Guidance
					Capitalisation not important. "Sent" and "not sent" do not have to be exactly this – can be alternative message conveying same idea.
2	**(a)**	**(i)**	• 2, 3, 4	1	All three numbers needed in the correct order (with no other numbers) for mark.
2	**(a)**	**(ii)**	• 15	1	Accept 3 x 5
2	**(b)**		1 mark per bullet, max 2 • Sequence • Iteration / loops / repetition	2	Ignore spelling. Do not allow examples (eg FOR loop / WHILE loop)
2	**(c)**	**(i)**	1 mark per bullet, max 2 • A (name/identifier for a) **memory location** • used to (temporarily) **holds/contains/stores** data / value // is assigned a value • that **can be** changed / **possible** to change (while the program is running)	2	Do not accept "will change" for bullet point 4. Do not allow "holds/stores something" or "holds/stores information" for bullet point 2. Do not accept name / identifier without reference to a memory location. Do not accept "a value given a name" or equivalent.
2	**(c)**	**(ii)**	1 mark per bullet, max 2 • k • p • m	2	Ignore capitalisation. Correct answer only. Do not allow other code in answer.
3	**(a)**	**(i)**	1 mark per bullet, max 2 • AND / conjunction • NOT / negation	2	Allow Boolean notation.

<table>
<tr><th colspan="3">Question</th><th>Answer</th><th>Mark</th><th>Guidance</th></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>3</td><td>(a)</td><td>(ii)</td><td>
<table>
<tr><th>A</th><th>B</th><th>Q</th></tr>
<tr><td>0</td><td>0</td><td>1</td></tr>
<tr><td>0</td><td>1</td><td>1</td></tr>
<tr><td>1</td><td>0</td><td>1</td></tr>
<tr><td>1</td><td>1</td><td>0</td></tr>
</table>
</td><td>4</td><td>1 mark per row</td></tr>
<tr><td>3</td><td>(b)</td><td></td><td>1 mark per bullet, max 2
 

OR gate with two inputs // NOT gate on B input
Logic system as above with no other gates, with labelled inputs of A and B.
</td><td>2</td><td>First mark can be awarded if candidate has either a NOT gate from B, or an OR gate with two inputs anywhere in their answer.

Second mark is only awarded of the logic system as shown is given with no other additional gates.

Correct logic diagrams needed for OR and NOT, including circle on NOT. Use professional judgement. Ignore labelling.

No need to label Q output.</td></tr>
</table>

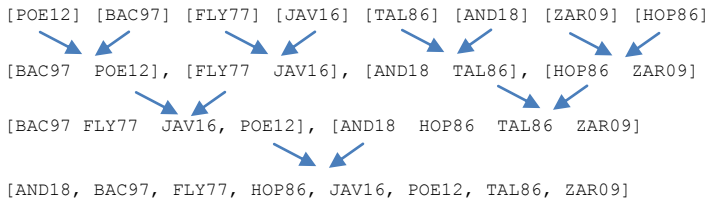
Question			Answer	Mark	Guidance
4	(a)	(i)	1 mark per filled gap, max 3 01 function librarycode(title, **year**) 02 parta = title.substring(0, **3**) 03 partb = year.substring(2, 2) 04 **return** parta.upper + partb 05 endfunction	3	Ignore capitalisation. Allow **librarycode =** for 3rd mark – this is an equivalent in some languages for returning a value (eg. Visual Basic).

Question			Answer	Mark	Guidance

4	(a)	(ii)	1 mark per bullet, max 6
			
			• Input title and year from user
			• Open bookcodes.txt
			• Call the librarycode() function…
			• … with the two parameters that match input values
			• … write out **code obtained** to the text file
			• Close text file
				6	Example algorithm
					
					`title = input("enter title")`
					`year = input ("enter year")`
					`code = librarycode(title, year)`
					`myFile = openWrite("bookcodes.txt")`
					`myFile.writeLine(code)`
					`MyFile.close()`
					
					Note, pseudocode shown above is an example – candidates may answer very differently, but award marks if intention can be seen.
					
					Bullet points 3,4 and 5 could be done in one line:
					`myFile.writeLine(librarycode(title, year))`
					
					Do not award bullet point 3 if candidate is defining the function rather than calling it.
					
					Allow bullet point 2 (opening text file) if correctly referred to during write operation.
					
					Bullet point 3 must include brackets () to signify it is the function being called or indication that is being called.

4	(b)	(i)	1 mark per bullet, max 2. • Function **returns** a value • Procedure **does not return** a value	2	Allow "does not" for second mark if intention is clear (ie if it is obvious that the "not" refers to not returning a value). Allow discussion of how returned value in a function can be used (e.g. to assign to a variable or to use this returned value in some way).

Question			Answer	Mark	Guidance
4	(b)	(ii)	1 mark per bullet, max 4. Mark in pairs. e.g. • Breaks down / decomposes / modularises the problem / program // structures the program • …making it easier to design/create/test • …each subroutine can be tested separately • Reuse code (in different programs) • …quicker to develop (new) programs • …build on existing work / use of a library of subroutines • Avoid repetition of code (in the same program) • …makes program shorter / smaller • … subprogram called instead of copying/pasting. • … quicker to develop (new) programs • Easier to maintain …as code is easier to understand/read • …as code is shorter • Easier to debug • …as code is shorter • …same bugs will not have been copied to other areas of the program. • Work can be split up in a team • ...to suit developers' skill set • …to work on different subprogram at the same time / develop separately • Allows for abstraction / removes complexity • …subprograms can be used by programmers who do not need to understand how they work.	4	Maximum of two benefits with expanstions to be marked as per question. Allow other sensible expansions. Allow expansions which cross over to other benefits (e.g. breaks down the problem / to make it easier to maintain). Allow "can be called multiple times" Allow "file size is smaller". Do not allow "more efficient" without further explanation.

Question			Answer	Mark	Guidance
4	(c)	(i)	1 mark per bullet, max 4. • List split into individual elements (may be done over several steps or just as a starting point) • **Merge** individual elements into **sorted** lists of size 2 • **Merge** lists of size 2 into **sorted** lists of size 4 • **Merge** lists of size 4 into final **sorted** list.	4	Candidates **can** describe how the merge sort would work rather than showing output values at each stage. Ignore intermediate steps. **Do not** give final mark for simply showing the list sorted. Must have the (correct) idea of where it being merged from previous lists. Candidates' answers describing / showing other sorting algorithms (e.g. bubble sort, insertion sort) are worth 0 marks. 
4	(c)	(ii)	1 mark per bullet, max 2. • Faster/quicker (to sort)… • …for large lists // for lists that are **more unordered** • Has a consistent running time (for a lists of same length)… • …doesn't depend on how ordered original list is	2	Accept (correct) reference to big O notation for 2nd mark on either mark point although this is beyond scope of GCSE specification. Allow "more efficient" for BOD on first bullet point.
5	(a)	(i)	• 1000 0100	2	1 mark per nibble. Mark right to left.
5	(a)	(ii)	• B 5	2	1 mark per hex digit

<table>
<tr><th colspan="3">Question</th><th>Answer</th><th>Mark</th><th>Guidance</th></tr>
<tr><td>5</td><td>(a)</td><td>(iii)</td><td>1 mark per bullet, max 1.
• 00001101
• Divides by 4</td><td>1</td><td>Accept 001101 / 1101. Allow any number of leading zeros.</td></tr>
<tr><td>5</td><td>(a)</td><td>(iv)</td><td>1 mark per bullet, max 2.

• Left shift
• one place</td><td>2</td><td>Do not accept answers that simply show the number shifted.</td></tr>
<tr><td>5</td><td>(b)</td><td>(i)</td><td><table><tr><td>a</td><td>1100001</td></tr><tr><td>e</td><td>1100101</td></tr></table></td><td>2</td><td>1 mark per row. Correct answer only. Do not allow leading zeros.</td></tr>
<tr><td>5</td><td>(b)</td><td>(ii)</td><td>1 mark per bullet, max 2.

• Extended ASCII uses more bits // ASCII uses fewer bits
• Extended ASCII can represent more characters // ASCII can represent fewer characters
• … by example (e.g. extended ASCII can represent European symbols / other languages)</td><td>2</td><td>Allow numbers (e.g. ASCII has 7 bits, Ex. ASCII has 8 bits) for either bullet point but these must be realistic.

Bullet point 1 and 2 must be a comparison (e.g. “ASCII is 7 bits” is not enough on its own).

Do not accept answers that are technically wrong (e.g. “ASCII does not contain symbols such as ?, !, #”)</td></tr>
<tr><td>6</td><td>(a)</td><td></td><td><table><tr><th>Will loop infinitely</th><th>Will <u>not</u> loop infinitely</th></tr><tr><td>✓</td><td></td></tr><tr><td>✓</td><td></td></tr><tr><td></td><td>✓</td></tr><tr><td></td><td>✓</td></tr></table></td><td>4</td><td>1 mark per row.

More than one tick in a row = 0 marks for that row.</td></tr>
</table>

Question			Answer	Mark	Guidance

6	(b)		1 mark per bullet, max 3.
			
			• FOR loop used
			• That outputs the counter variable
			• loops 10 time
				3	Example algorithm
					
					`for i = 1 to 10`
					`  print i`
					`next`
					
					Do not accept WHILE loop for first mark, although other marks can be accessed.
					
					No need for `next`
					
					If candidate manually increments counter within FOR loop, do not award bullet point 3.
					
					Accept pseudocode that suggests looping 10 times, even if this may not function correctly in a specific language.

7	(a)	(i)	1 mark per bullet, max 2. • `else` • `print ("unknown")`	2	Accept logically correct equivalents for else (e.g. `elseif a!="LAN" and/or a !="WAN")`. Do not allow `elseif` on its own Accept other keywords for print (e.g. "output") as long as the intention is clear. Accept other messages as equivalent to "uknown" (e.g. "not known " / "error")) Message to be printed must be in quotes. Allow "`else then`".

Question			Answer	Mark	Guidance
7	(a)	(ii)	1 mark per bullet, max 2. • aimed at humans//understandable by humans / programmers • English like structure / syntax • Must be translated/compiled/interpreted (before it can be run) • Allows programmer to deal with the problem instead of considering the underlying hardware // an abstraction from the hardware // hardware independent // portable	2	Allow examples of keywords (eg IF / ELSE / WHILE) as 2nd bullet point. Do not award marks for naming languages such as Java , Python, etc. Do not award marks for stating what a high level language isn't (i.e. describing what low level code is). Do not allow "easy to use" Do not allow 'has to be converted' without into what i.e machine code etc.
7	(b)		1 mark per bullet, max 4. e.g. • Editor • …to enable **program code** to be entered/edited • Error diagnostics / debugging • …to display information about errors (syntax / run-time) / location of errors • … suggest solutions • Run-time environment • …to enable to the program to be run • … check for run time errors / test the program • Translator / compiler / interpreter • …to convert the high level code into <u>machine code</u> / <u>low level code / binary</u> • ...to enable to code to be executed / run	4	One mark for identifying, one mark for describing. Accept description of a tool without (or with incorrect) naming of the tool. Allow sensible descriptions which go across pairs or name other tools sensibly (e.g. editor / highlighting syntax) Allow any sensible tool that an IDE provides (e.g. auto documentation, help tools, pretty printing etc.)

Question			Answer	Mark	Guidance
			• Breakpoints • …to stop/pause program execution at a specific point • Watch window • …to check contents of variables • Stepping • …to execute program line by line • Syntax completion… • …suggests/corrects code • Keyword highlighting / colour coding keywords / pretty printing… • …colours command words / variables		

8			1 mark per bullet, max 6.
			
			• Initialisation of A, B and C as zero.
			• Allows input (of anything) from the user
			• Incrementing A, B and C depending on input
			• Repeats bullet points 2 and 3
			• …stopping **only** when "END" is entered
			• Prints out all 3 individual counts <u>and</u> prints **calculated** total count
				6	Example algorithm
					
					`acount = 0`
					`bcount= 0`
					`ccount= 0`
					`vote = ""`
					`while vote != "END"`
					`  vote = input("enter A, B or C")`
					`  if vote == "A" then`
					`  acount = acount + 1`
					`  elseif vote == "B" then`
					`  bcount = bcount + 1`
					`  elseif vote == "C" then`
					`  ccount = ccount + 1`
					`  end if`
					`endwhile`
					`print acount`
					`print bcount`
					`print ccount`
					`print acount+bcount+ccount`

Question			Answer	Mark	Guidance
					Do not penalise for missing initialisation of variable used in the while loop or total (if used) Comparison with value inputted MUST be a string (e.g. `if vote == A`) is incorrect as A here is a variable, not a string. Answer can be any recognised algorithm – pseudocode, flowcharts, structured English, etc. Mark on whether the bullet points on the left hand side have been met. Does not have to match algorithm above. 4th bullet point (repeat) can be given for any sensible attempt at iteration. Use professional judgement on where loops end (WHILE / END WHILE or indentation).

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

Oxford Cambridge and RSA Examinations
is a Company Limited by Guarantee
Registered in England
Registered Office; The Triangle Building, Shaftesbury Road, Cambridge, CB2 8EA
Registered Company Number: 3484466
OCR is an exempt Charity

OCR (Oxford Cambridge and RSA Examinations)
Head office
Telephone: 01223 552552
Facsimile: 01223 552553

© OCR 2018