OCR
Oxford Cambridge and RSA

GCSE (9–1)

Computer Science

J276/02: Computational thinking, algorithms and programming

General Certificate of Secondary Education

Mark Scheme for June 2019

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.

Annotations

Annotation	Meaning
BP	Blank page
^	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

<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>(i)</td><td>1 mark per bullet to max 2.

• Height/amplitude of waveform is sampled/measured
• Converted to / stored as binary/digital
• Sample / measurements taken at a regular interval / set interval / by sensible example (eg 44,000 times per second)</td><td>2
AO1 1b (2)</td><td>Do not accept frequency

Do not accept unrealistic sample rates (e.g. once per second).</td></tr>
<tr><td>1</td><td>(a)</td><td>(ii)</td><td>1 mark per bullet to max 1.

• number of samples taken per second / per time period
• How <u>often/regularly</u> a sample is taken</td><td>1
AO1 1a (1)</td><td>Accept reference to Hertz (Hz) as time period.</td></tr>
<tr><td>1</td><td>(a)</td><td>(iii)</td><td>1 mark per tick to max 2.
<table><tr><th></th><th>Tick (✓) two boxes</th></tr><tr><td>The file size of the digital recording will be smaller</td><td></td></tr><tr><td>The file size of the digital recording will be larger</td><td>✓</td></tr><tr><td>The quality of playback of the digital recording will be better.</td><td>✓</td></tr><tr><td>The quality of playback of the digital recording will be worse.</td><td></td></tr></table></td><td>2
AO1 1b (2)</td><td>If 3 ticks given, max 1 mark
If 4 ticks given, 0 marks.</td></tr>
<tr><td>1</td><td>(b)</td><td>(i)</td><td>1 mark per bullet to max 3.

• Image made of / split up into pixels
• Each pixel given a binary code…
• …which represents the colour of that pixel
• Each colour is given a different/unique binary code.
• Metadata stored alongside the image</td><td>3
AO1 1b (3)</td><td>BP1 needs idea of picture made up of pixels, not just mention of the word “pixel”

Not enough to say “each colour is given a binary code”, must have the idea of this being unique or different for each different colour.

Accept examples of metadata such as height/width, geolocation, etc. Do not accept file size/file name.</td></tr>
</table>

1	**(b)**	**(ii)**	1 mark per bullet to max 2. • Computers consist of transistors / switches / logic circuits / gates… • …which only have two values / on or off / 1 or 0 / open or closed	2 AO1 1b (2)	Only give BP2 if BP1 given. BP1 must refer to **hardware** that switches between two states. Do not accept processor for this.
1	**(b)**	**(iii)**	1 mark per bullet 2 marks max for advantages 2 marks max for disadvantages Advantages • **File** size is reduced / gets smaller // image contains less data • ..so quicker to **upload / download / load / transfer** • ...takes up less storage space // space on the web server • …less (mobile) data usage • …permanent reduction Disadvantages • Quality of image is reduced (compared to original) • …because data is lost / removed // by example (eg fewer colours / lower resolution) • …reduction is permanent / not reversible	4 AO2 1b (4)	Do not accept “size” to mean “file size” for advantage. “Data is permanently removed” gets 2 marks for a disadvantage

2	**(a)**		1 mark per bullet to max 4, 1 mark per row • 10 • 6 • 6 • 2	4 AO2 1b (4)	Correct Answer Only Do not accept "X", "Y", etc.

2	**(b)**		1 mark per bullet to max 6.
			
			• Inputs two value (as X and Y)
			• Compares if X is larger than Y…
			• …Outputs Y*X only when <u>False</u>
			• Compares if X is less than 12…
			• …Outputs X only when True <u>and X > Y</u>
			• …Outputs Y only when False <u>and X > Y</u>
				6
				AO3 2b (6)
					Question specifically asks for pseudocode.
					
					Outputs should only be given if they occur with the right condition(s).
					
					<u>Example algorithm</u>
					
					`input x`
					`input y`
					`if x > y then`
					`  if  x < 12 then`
					`  print x`
					`  else`
					`  print y`
					`  end if`
					`else`
					`  print y*x`
					`end if`
					
					Variables do not have to be called x and y.
					
					Accept equivalent comparisons (e.g. `if X <= Y`)
					
					Allow FT for outputs from incorrect comparisons where a sensible attempt has been made.

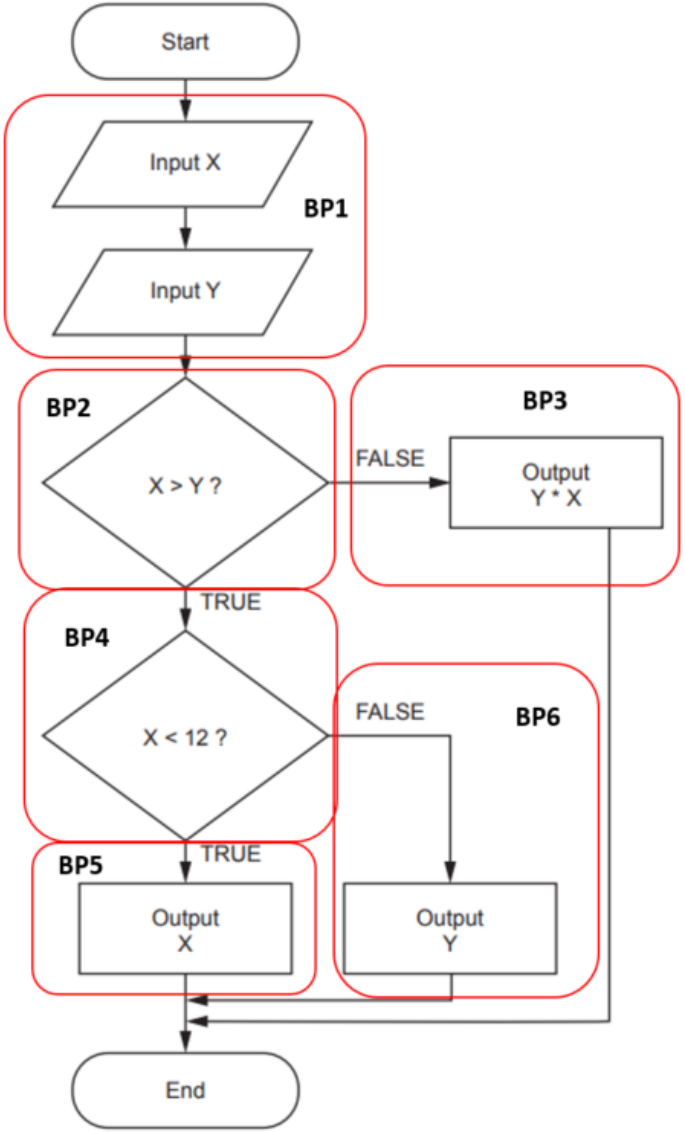
Please note how mark bullet points match up to the flowchart given in the question.

BP1 is for taking both inputs

BP2 and 4 are for correct comparisons of variables. This may be done in alternative ways (e.g. X<=Y, X>=12, etc)

BP3, 5 and 6 are for the correct outputs in the right place.

If the answer logically works to produce the correct output, it should be marked as correct.



2	**(c)**		1 mark per bullet to max 4, 2 mark max per method • Compiler • …translates code **in one go** / all at once • …produces an executable file // does not need to be compiled again • Interpreter • …translates code line by line. • …will be interpreted / translated every time it is run.	4 AO1 1b (4)	Mark first method only in each section
3	**(a)**	**(i)**	1 mark per bullet to max 1 • An error that results in **incorrect output / unexpected result** • Contains an error but **still runs / doesn't crash**	1 AO1 1b (1)	Do not accept examples of logic errors.

3	**(a)**	**(ii)**	`if num MOD 2 == 0 then`
			
			`if num MOD 2 = 0 then`
				1
				AO3 2b (1)
					Important point is that >= is changed to == or =.
					Accept alternatives that produce the same result (e.g. <=0, <1, !=1, etc.)
					
					Ignore any casting (e.g. using int() to convert to a number)
					
					Accept other minor changes to the line as long it logically works.
					
					Accept versions of MOD from high level languages (e.g. Python : `if num % 2 == 0`)

3	**(b)**	**(i)**	1 mark per bullet to max 1 • An error in the grammar of the program // error that breaks the rules of the programming language • Contains an error but will not run / translate / execute	1 AO1 1b (2)	**Do not** accept examples of syntax error (e.g. misspelling).
3	**(b)**	**(ii)**	`print("odd")`	1 AO2 1b (1)	Must include quotes (single or double). Do not penalise spelling mistakes in message. Accept sensible alternatives to "odd" Accept alternatives for print / output as long as spelling is accurate
4	**(a)**	**(i)**	1 mark per bullet to max 2 • Removing / hiding / obscuring **unnecessary** detail • Focusing on the important detail • Simplifies **the problem** // reduces complexity // Easy to solve / understand	2 AO1 1a (1) AO1 1b (1)	Accept answers relating to using fewer computational resources Must be the programmer making the decision.
4	**(a)**	**(ii)**	1 mark per bullet to max 1 • Suitable example of what can be focused on (e.g. player name, match results, goals scored) • Suitable example of what to remove/hide (anything relevant that is not results/goals scored) • Suitable example of a simplification made	1 AO2 1a (1)	Mark first answer only Allow any suitable example of abstraction as long it is relevant to the system. Allow either first name or surname to be removed as an example, but do not allow both to be removed.

4	**(b)**		1 mark per bullet, mark in pairs. Max 2 per point. e.g. • Input sanitisation • …cleaning up input data / removing unwanted data • …by example (e.g. removing special characters / preventing SQL injection) • Validation • …checking whether input data should be allowed / is sensible / follows criteria • …by example (e.g. goals cannot be less than 0) • Verification • … checking whether data has been entered correctly • …by example (e.g. double entry / visual check) • Authentication • …ensuring only allowed / authorised users can gain access • …by example (e.g. usernames /passwords) • Maintainable code • …to allow other programmers to understand the code • …by example(e.g. comments, indentation, meaningful variable names)	4 AO2 1a (2) AO2 1b (2)	Mark first answer only in each section For validation, allow one example of a type of validation (e.g. type check, range check) e.g. question so allow other sensible examples such as audit logging, encryption of data Do not allow “data is correct” as expansion for validation – validation checks that data is sensible or follows rules, NOT that it is correct. Planning for contingencies and anticipating misuse are not examples by themselves, but discussion of these may fit under other points – e.g. input sanitisation, validation.

4	**(c)**		1 mark per bullet to max 3 • `count` • `= nogoalscount + 1` • `nogoalscount`	3 AO3 2b (3)	Correct answer only. Accept alternatives to adding 1 to variable (e.g. `+= 1` / `++`) Penalise spelling once only, FT for further mistakes. Do not penalise case. Accept sensible messages printed out alongside `nogoalscount`
5	**(a)**		1 mark per bullet to max 2 • 163 • **Correct** working shown.	2 AO1 1b (2)	Award working mark independently of final answer but working must be correct (e.g. (16 x 10) + 3)
5	**(b)**		1 mark per bullet to max 2 • 91 • **Correct** working shown.	2 AO1 1b (2)	Award working mark independently of final answer but must be correct (e.g. 1+2+8+16+64 // correct binary headings with correct binary underneath)
5	**(c)**		• 9	1 AO1 1b (1)	Correct answer only Do not accept 3^2 or 3 x 3
5	**(d)**		1 mark per nibble to max 2 • 1101 1101	2 AO1 1b (2)	Mark from right to left.

<table>
<tr>
<td>5</td>
<td>(e)</td>
<td></td>
<td>1 mark per missing bit
<table>
<tr><th>A</th><th>B</th><th>Q</th></tr>
<tr><td>0</td><td>0</td><td>0</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>1</td></tr>
</table>
</td>
<td>4
AO2 1b (4)</td>
<td>Accept T / True</td>
</tr>
</table>

6	**(a)**	**(i)**	1 mark per bullet to max 6
			
			• Function ticketprice() **defined**
			• … that accepts two parameters and has no other inputs
			• Works out total ticket price for adult (eg adult * 19.99)
			• Works out total ticket price for children (eg child * 8.99)
			• Adds on **correct** booking fee
			• Returns the calculated value.
				6
				AO3 2b (6)
					Bullet points 3, 4, 5 can be awarded even if no mention of a function / parameters (for example, if candidate has inputted the number of tickets needed.
					
					Do not award return value if no attempt at a function. Return mark can be given if a good attempt made at calculating the total, even if this is incorrect.
					
					Allow 2.50 booking fee to be per order or per ticket
					
					Ticket prices must be stored appropriately if needed.
					
					example algorithm
					
					`function ticketprice(numadult, numchild)`
					`  price = (numadult * 19.99) + (numchild * 8.99) + 2.50`
					`  return price`
					`end function`
					
					Allow alternatives in high level languages (e.g. def in Python).
					
					Allow return as assigning the value to the name of the function (VB syntax)

6	**(a)**	**(ii)**	1 mark per bullet to max 2 • Real… • …Returned value may not be a whole number / may have a decimal point in	2 AO2 1a (1) AO2 1b (1)	Allow String only if matching justification shows understanding (e.g. £ sign attached, message returned alongside value).

6	**(b)**	**(i)**	1 mark per bullet to max 1 • Check that the code is valid / real • Check it has been entered / sent / received correctly. • Makes it harder for people to make up discount codes	1 AO2 1b (1)	Mark first answer only
6	**(c)**	**(i)**	• Not in order / sorted	1 AO2 1b (1)	Mark first answer only
6	**(c)**	**(ii)**	• Linear (search)	1 AO1 1b (1)	Mark first answer only Allow other valid searching algorithms as long as they work on an unsorted list (e.g front and back search)
6	**(d)**	**(i)**	1 mark per bullet to max 2 • Flag / record whether a swap has taken place or not • checked as condition to decide **whether to repeat**	2 AO2 1b (2)	The variable records whether a swap has taken place; it does **no**t perform the swap.
6	**(d)**	**(ii)**	1 mark per bullet to max 2 • Swaps.. • …values of `queuesize[p]` and `queuesize[p+1]` • …when `queuesize[p]` is larger than `queuesize[p+1]` • using a temporary variable //doesn't overwrite numbers //explanation of process	2 AO2 1b (2)	Do not accept "sorts numbers" "swaps numbers" meets BP1. Explanation of which values in the array are swapped meets BP1 and BP2. Do not accept direct word for word repetition from the program (e.g. `temp = queuesize[p]`) , question asks for an explanation. Explanation of temporary variable must be logically correct.

6	**(d)**	**(iii)**	1 mark per bullet to max 2. • Comments • … to enable programmers to understand the purpose of each line / section • …by example (e.g. on line 4 add the comment…) • Naming variables sensibly • … to enable programmers to understand the purpose of each variable • …by example (e.g. change identifier p to …) • Modularise • …to allow reuse / makes easier to test / reduces errors • …by example (e.g. create as a function)	2 AO2 1a (1) AO2 1b (1)	Mark first answer only Do not accept indentation (already done) Accept “show what each line does” for comments.
6	**(d)**	**(iv)**	1 mark per bullet to max 2. • Insertion (sort) • Merge (sort)	2 AO1 1a (2)	Accept “insert”. Do not penalise spelling. Do not accept bubble sort (given in previous questions) Do not award searching algorithms Allow other valid sorting algorithms. (e.g. quick sort, heap sort, shell sort, selection sort, radix sort, bucket sort, tim sort, comb sort, pigeonhole sort, etc.)

<table>
<tr>
<td>6</td>
<td>(e)</td>
<td></td>
<td>1 mark per bullet to max 8.

• Input height
• Accepts riders > / >= 140 with suitable message
• Rejects riders < / <= 120 with suitable message
• Checks if height between 120 and 140…
• … If True, input whether accompanied
• … Suitable output message for True AND False
• Correctly counts number of riders <u>in all cases of being allowed to ride</u> (do not penalise candidates for counting or not counting accompanying adults)
• Attempt to loop based on 8 riders allowed

Some checks for rider height may be implicit (e.g. using ELSE after checking other heights). If the answer logically works to produce the correct output, it should be marked as correct.

Loop will almost certainly be condition controlled (WHILE/DO UNTIL) to gain BP8; count controlled (FOR) loop requires significant manipulation to work successfully.</td>
<td>8
AO3 2b (8)</td>
<td>Answers can be in any suitable format (including pseudocode, flowchart, etc). If flowchart used, accept any sensible shapes.

Do not penalise for lack of initialisation of variables.

Loop must repeat until 8 riders allowed, not just loop 8 times.

Do not credit asking whether accompanied if in the wrong place.

Condition for BP4 may be <code>120 < h < 140</code>

<u>Example algorithm</u>
<pre>
riders=0
while riders <8
      input height
      if height >= 140 then
          output "allowed"
          riders = riders + 1
      elif height >=120 then
          input withadult
          if withadult == "yes"
              output "allowed"
              riders = riders + 1
           else
              output "not allowed"
          end if
      else
          output "not allowed"
      end if
endwhile
</pre></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

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