Chhetri Academy

chhetriacademy.com

GCSE COMPUTER SCIENCE

2.4

OCR J277 | Specification point 2.4.1

Boolean logic

Untiered

Answers and examiner comments

5	2014-2024	Untiered
question groups	available papers	tier

Using this booklet

1. Answer numbers match the companion question booklet.
2. Follow the original question number on marking tables.
3. Examiner comments are included where available.

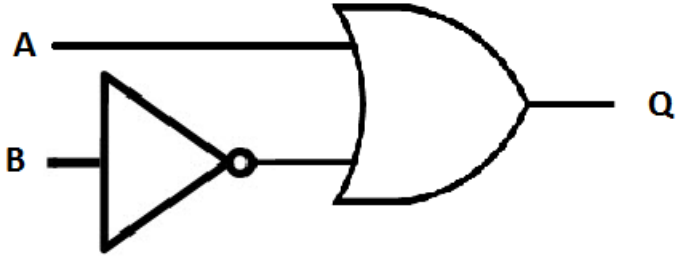
Name _____ **Date** _____

Some entries use full source marking material in an appendix. Any unavailable mark schemes are identified beside the relevant answer.

<table>
<tr><th colspan="3">Question</th><th>Answer</th><th>Mark</th><th>Guidance</th></tr>
<tr><td>7</td><td></td><td></td><td>
<table>
<tr><th>A</th><th>b</th><th>NOT(a AND b)</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>
1 mark for row two and three. For row 4, 1 mark for correctly identifying 1 1 as the inputs, and 1 mark for the correct output 0)</td><td>4</td><td>No follow through on row 4.</td></tr>
</table>

3	**(a)**	**(i)**	1 mark per bullet, max 2 • AND / conjunction • NOT / negation	2	Allow Boolean notation.

J276/02 Mark Scheme June 2018

<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.

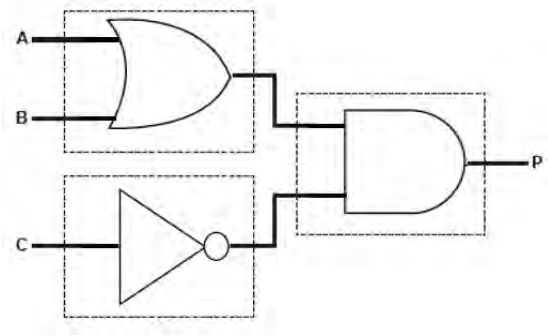
Second mark is only awarded of the logic system as shown is given with no other additional gates.

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>

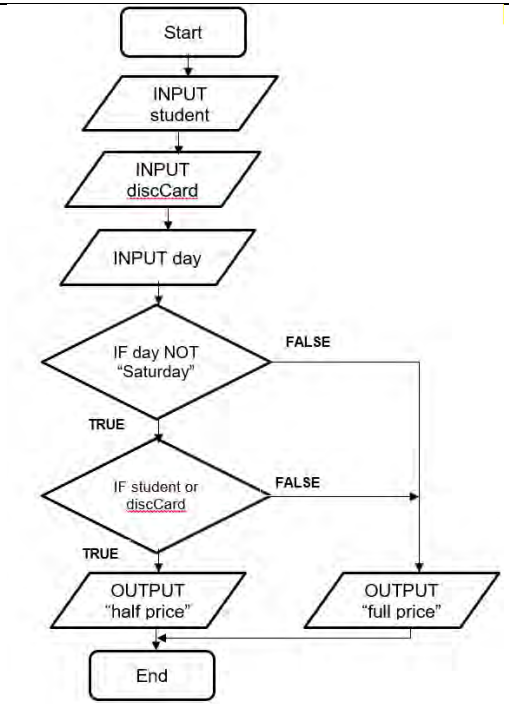
Original question 4

The full source mark scheme begins on page 10. Use the original paper code, tier and question number shown above.

Question			Answer	Mark	Guidance
2	(a)	(i)	• A **OR** B • **NOT** C • **AND** gate  	3 **(AO2 1b)**	1 mark per gate. Correct symbols must be used. NOT gate must have circle for inversion, OR and AND must no have a circle. Mark the **shape** of each gate, not the name written if given. Ignore any writing / notes. Lines do not have to be drawn or joined up, but if they are, gate must have the **correct number of inputs/outputs**. Penalise once then FT.
2	(a)	(ii)	• To show **all possible inputs** (to the logic circuit)… • **…**and the **associated/dependent** output (for each input)	2 **(AO1 1b)**	For 2nd BP, must be clear that the output is linked to the input values given. “All possible combinations of inputs and outputs” gains the first mark (all possible inputs) but not the second. “The output **for** each possible input” gains both marks
2	(a)	(iii)	• 8 // eight	1 **(AO2 1a)**	Accept other answers that equate to 8 (e.g. 2^3)

<table>
<tr>
<td>2</td>
<td>(b)</td>
<td></td>
<td>

Start and end/stop with all boxes connected, no boxes that do not lead to another box (no arrows needed)
Input three variables using parallelogram shape
Checks all three criteria (day, student, discount card) using diamond shape(s) with two lines from each
…Outputs "full price" with correct conditions using parallelogram shape
…Outputs "half price" with correct conditions using parallelogram shape

Guidance for correct outputs
<table>
<tr><th>Conditions</th><th>Outcome</th></tr>
<tr><td>Not Saturday and (either a student or has a discount card).</td><td>Half price</td></tr>
<tr><td>Saturday or (not a student and doesn't have a discount card).</td><td>Full price</td></tr>
</table>
<table>
<tr><th>Saturday</th><th>Student</th><th>Discount Card</th><th>Outcome</th></tr>
<tr><td>N</td><td>N</td><td>N</td><td>Full price</td></tr>
<tr><td>N</td><td>N</td><td>Y</td><td>Half price</td></tr>
<tr><td>N</td><td>Y</td><td>N</td><td>Half price</td></tr>
<tr><td>N</td><td>Y</td><td>Y</td><td>Half price</td></tr>
<tr><td>Y</td><td>N</td><td>N</td><td>Full price</td></tr>
<tr><td>Y</td><td>N</td><td>Y</td><td>Full price</td></tr>
<tr><td>Y</td><td>Y</td><td>N</td><td>Full price</td></tr>
<tr><td>Y</td><td>Y</td><td>Y</td><td>Full price</td></tr>
</table>
</td>
<td>5
(AO3 2a)</td>
<td>


Question asks for a flowchart. Answers as pseudocode, high level language or other forms are not acceptable 9 (NAQ).

BP 4 and 5 only to be awarded if all decisions ensure correct output and clear what the decisions are. FT for incorrect shapes used or no inputs as long as decisions are logically correct. Mu attempt all three decisions.

Allow calculation of half price / full price instead of message bu this must still be output.

Inputs / decisions may be presented as individual or combined boxes but must still store as three variables.

Penalise lack of parallelogram for input/output once only then F

BOD parallelogram shapes if not sure whether input or output a long as context is clear (e.g inputs at start, outputs at end)
</td>
</tr>
</table>

<table>
<tr><td>2</td><td>(c)</td><td></td><td>• Number of people (at the table) // whether there are more than 5 people or not
• Choice between percentage and value // actual value of both percentage, value</td><td>2
(AO3 2a)</td><td>Ignore additional inputs that would be sensible, such as cost of the meal.

Accept inputs in form of pseudocode / high-level language.

Max 1 if other irrelevant inputs given.

“Whether to leave a tip or not” or “Amount of tip” NE for BP2. Must address both the percentage and value of tip if asked for. BOD “type of tip” for BP2</td></tr>
<tr><td>2</td><td>(d)</td><td>(i)</td><td>• Convert/change one data type to another
• Line 03 // 3 // three</td><td>2
(AO1 1b, AO2 2b)</td><td>Do not accept “change to string” – this is the use in this examp but not a definition.</td></tr>
<tr><td>2</td><td>(d)</td><td>(ii)</td><td>• Kofi2021 as staffID on line 03
• Kofi2021x as staffID on line 05
• Kofi2021xx as staffID on line 05
• ID Kofi2021xx output on line 07 as first and only output</td><td>4
(AO3 2c)</td><td>Max 2 if incorrect order. Ignore misspelling of Kofi

Penalise lack of / errors with line numbers once then FT. Ignore capitalisation. Ignore additional lines unless outcome impacted.

<code>staffID</code> does not have space in. Output does have a space in. Penalise spaces once then FT. Do not penalise unless obvious

Quotes around answer is OK, but do not allow quotes around partial answers, e.g. <code>“ID” Kofi2021xx</code> is incorrect.
<table>
<tr><th>Line number</th><th><code>surname</code></th><th><code>year</code></th><th><code>staffID</code></th><th>Output</th></tr>
<tr><td>01</td><td>Kofi</td><td></td><td></td><td></td></tr>
<tr><td>02</td><td></td><td>2021</td><td></td><td></td></tr>
<tr><td>03</td><td></td><td></td><td>Kofi2021</td><td></td></tr>
<tr><td>05</td><td></td><td></td><td>Kofi2021x</td><td></td></tr>
<tr><td>05</td><td></td><td></td><td>Kofi2021xx</td><td></td></tr>
<tr><td>07</td><td></td><td></td><td></td><td>ID Kofi2021xx</td></tr>
</table>
</td></tr>
</table>

Original question 5

The full source mark scheme begins on page 23. Use the original paper code, tier and question number shown above.

PMT

GCSE (9–1)

Computer Science

J276/02: Computational thinking, algorithms and programming

General Certificate of Secondary Education

Mark Scheme for November 2020

Oxford Cambridge and RSA Examinations

PMT

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 2020

Oxford Cambridge and RSA Examinations

J276/02 Mark Scheme November 2020

Annotations

Annotation	Meaning
SEEN	Answer seen – 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
	Benefit of doubt
	Cross
	Follow through
	Not answered question
	Benefit of doubt not given
	Repeat
	Tick

Each question must include annotation. All marks credited on point-based marking must be given a tick.

J276/02 Mark Scheme November 2020

<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>1 mark for each letter

<table><tr><td>Decomposition</td><td>D</td></tr><tr><td>Abstraction</td><td>B</td></tr><tr><td>Input Sanitisation</td><td>A</td></tr><tr><td>Casting</td><td>F</td></tr></table></td><td>4

AO1
1a(4)</td><td>Accept answers that write the definition instead of the letter.</td></tr>
<tr><td>1</td><td>(b)</td><td>(i)</td><td>• <code>timer = 7.3</code></td><td>1

AO3
2b(1)</td><td>Ignore dim / define / etc and data types
Do not allow use of string delimiters or other unsuitable data types.
Allow other suitable assignment symbols (e.g. :=) Do not allow == for assignment.
Do not penalise case. Spelling must be accurate</td></tr>
<tr><td>1</td><td>(b)</td><td>(ii)</td><td>• Real // Float</td><td>1

AO2
1b(1)</td><td>Allow decimal, single, double or equivalent</td></tr>
<tr><td>2</td><td>(a)</td><td></td><td><table><tr><td>Line</td><td>Program code</td><td>Output</td></tr><tr><td>08</td><td><code>print score</code></td><td>18</td></tr><tr><td>09</td><td><code>print "name"</code></td><td>name</td></tr><tr><td>10</td><td><code>print newscore(score,2)</code></td><td>37</td></tr><tr><td>11</td><td><code>print score</code></td><td>18</td></tr></table></td><td>4

AO2
1b(4)</td><td></td></tr>
</table>

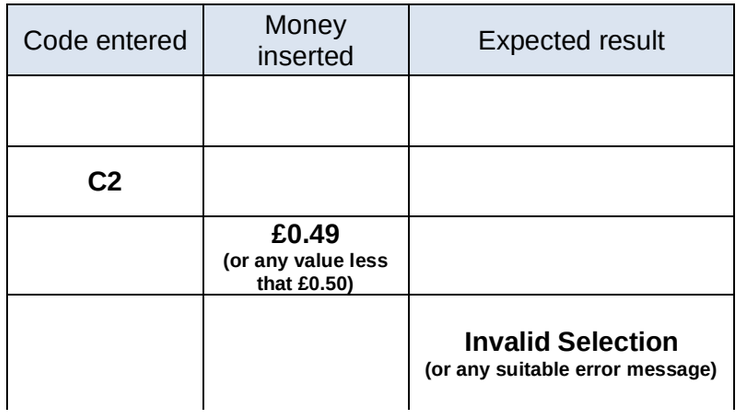
J276/02 Mark Scheme November 2020

2	**(b)**		1 mark per bullet to max 2 • **Easier/quicker** for humans to **write** • **Easier/quicker** to **read / understand / remember** • **Easier/quicker** to **maintain / debug / spot errors** • …because code is closer to English / uses English words • Less code to write • …because one HLL instruction represents many assembly instructions • Portable (between processors) // will work with different types of computer	2 AO1 1b(2)	Accept “human language” as English for BP4 “Easier to use” is too vague.
2	**(c)**		1 mark per bullet to max 2 • Each character (in character set) has a **unique** (binary) number/value • Each character in the **string** is assigned its associated number/value • The (binary) value of each character is stored/combined (in order) • … by example e.g. The binary value for D, then for r, then for u • Uses ASCII/Extended ASCII/Unicode	2 AO2 1a(2)	

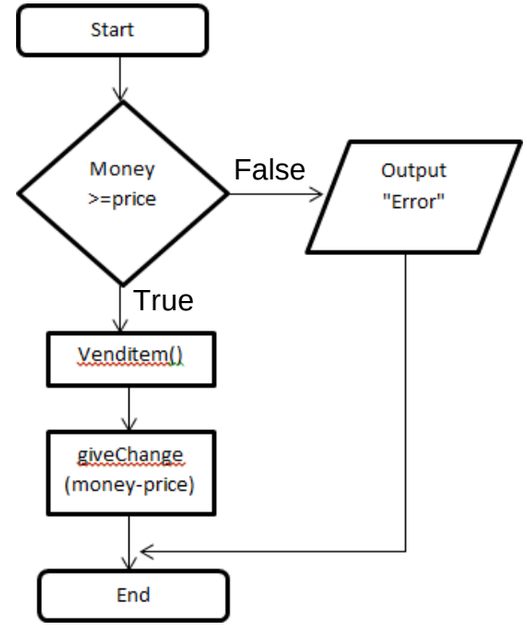
J276/02 Mark Scheme November 2020

Question			Answer	Mark	Guidance

3	(a)	(i)	1 mark per bullet to max 2
			e.g.
			• Check the program meets the **user requirements**
			• Check the program works (as intended) // detect logic / syntax errors
			• Check the program does not crash (under invalid entry) // check error messages are suitable
			• …allow these errors to be fixed
			• …make sure there are no problems when released
			• Any suitable example **related to the vending machine** e.g. gives correct change
				2
				
				AO1
				1b(2)
					Allow two any suitable examples for two marks
					
					BOD "find errors", "find bugs" for BP2
					
					"fix errors" by itself is one mark (BP4)

3	(a)	(ii)	1 mark per bullet to max 2 • Iterative is during development // repeatedly testing after/while making changes • Final is when the development is (almost) complete // done after iterative testing	2 AO1 1b(2)	Do not accept just "repeatedly testing" for iterative BOD "iterative testing tests modules/sections"
3	(a)	(iii)	 	3 AO3 2b(3)	For £0.49 accept any value <£0.50. Must be a specific value, not a description. Accept any suitable error message for invalid selection

J276/02 Mark Scheme November 2020

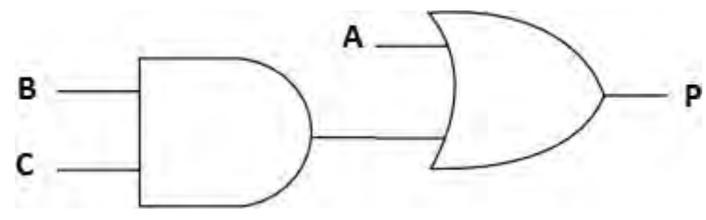
3	**(b)**	**(i)**	• `money` • `price`	1 AO1 1b(1)	Must be an identifier, not description. Ignore case.
3	**(b)**	**(ii)**	• one	1 AO2 1b(1)	
3	**(c)**		1 mark per bullet • Checking if `money>=price`… • …decision (diamond shape) used • …`venditem()` and `giveChange(money-price)` if True/Yes • …output an error if False / No • Terminator used to start and end the program and all paths terminated	5 AO3 2b(5)	Reasonable attempt at BP1 needed for credit BP2, 3 and 4 Ignore other additional code. BP3 and BP4 must follow on from True/False // Yes/No decision to be credited. Subroutines names and parameters must be correct. Ignore missing brackets on `venditem`. 

J276/02 Mark Scheme November 2020

Question			Answer	Mark	Guidance
3	d	i	1 mark per bullet to max 2 • Indentation // whitespace • Appropriately named variables / identifiers • Modularisation / use of subroutines	2 AO2 1b(2)	
3	d	ii	• Comments • Use of constants	1 AO2 1b(1)	
3	e		• `SELECT ItemCode // *` • `FROM ITEMS` • `WHERE` • …`Stock < 10`	4 AO3 2b(4)	Accept other fields shown in addition to `ItemCode` Accept `Stock <=9` / etc. Ignore case. Spelling of fields and table must be correct. If `WHERE` missing, `Stock < 10` must be after `FROM` clause.

3	f		1 mark per bullet
			• Input from user
			• Check IF input value is "**on**"…
			• … if so, assign **1** to `statevalue`
			• Correct assignment of **2 for "off"** and **3 for "suspended"** with correct state and `IF`
			• Correct logical check (else) to output "invalid state" <u>if no state set</u>
				5
				
				AO3
				2b(5)
					Accept alternative error messages. Variable names must not include obvious spaces.
					
					BP3 dependent on BP2. BP2 and BP4 must be a logical comparison using `IF` and not just the CASE statement. NE to simply replace CASE with `IF`.
					
					Penalise each error once then apply FT.
					
					e.g.
					`newstate = input("Enter the new state : ")`
					`if newstate == "on" then`
					`  statevalue = 1`
					`elseif newstate = "off" then`
					`  statevalue = 2`
					`elseif newstate = "suspended"`
					`  statevalue = 3`
					`else`
					`  print("Invalid state")`
					`endif`

J276/02 Mark Scheme November 2020

4	**(a)**		• E 3	2 AO1 1b(2)	1 mark per digit (mark right to left) Max 1 if any additional leading values
4	**(b)**		• 0110 1001 must be 8 bits	2 AO1 1b(2)	1 mark per nibble (mark right to left). Max 1 if any additional leading values
4	**(c)**		1 mark per bullet to max 2 • Easier/quicker to communicate / enter / write / read / remember • Less chance of input errors // easier to spot errors • They are smaller / shorter • Easy to convert between binary and Hexadecimal	2 AO1 1b(2)	Mark response as a whole. Do not accept answers simply describing what hexadecimal is. “easier to understand” or “easier to use” on its own is NE BP3 (smaller) must refer to size when written down, NOT size when stored which is unaffected)
4	**(d)**	**(i)**	• 3	1 AO1 1b(1)	CAO
4	**(d)**	**(ii)**	• 1	1 AO1 1b(1)	CAO
4	**e**		• 00001111	1 AO1 1b(1)	Ignore missing or additional leading zeros
4	**f**	i	1 mark per bullet point • B AND C • OR gate with two inputs, one of which is A • …correct connection of these two gates with no additional gates / connections	3 AO1 1b(3)	Shape must be accurate 

PMT

J276/02 Mark Scheme November 2020

4	**f**	ii	1 mark per bullet point • Correct completion of A and B inputs as **1 1** • 0 output for 01 input • 0 output for 10 input • 0 output for 11 input	4 AO1 1b(1) AO2 1b(3)	CAO

A	B	P
0	**0**	**1**
0	**1**	**0**
1	**0**	**0**
1	**1**	**0**

J276/02 Mark Scheme November 2020

Question			Answer	Mark	Guidance
5	a		• Number of pixels (in an image) • Height and width (of an image)	1 AO2 1b(1)	Accept pixels per inch / mm / unit area (density)
5	b		• 90 (pixels in an image) // 15 x 6 (pixels in image) • Multiply pixels x bits per pixel • …2 bits required per pixel (because 3 colours) • **180** bits overall answer	4 AO1 1b(2) AO1 1b(2)	Must clearly show multiplication for 3rd BP
5	c		• Reduce number of pixels / resolution • Reduce number of colours • Use lossy compression • Use lossless compression	2 AO2 1a(2)	Accept descriptive answers linked to given logo (e.g "change to black and white only") “Make image smaller” is NE Allow compression by itself for one answer.
5	d	i	• Data about data / the image/file // properties of the file	1 AO1 1b(2)	Do not accept examples without a definition.
5	d	Ii	e.g. • height • width • colour depth • resolution • geolocation • date/time **created/last edited //** timestamp • file type • author details	1 AO1 1a(2)	Accept any sensible data that could be stored alongside an image. Do not accept filename
6	(a)		• Access “Rob” / studentnames[0]… • …does **not** equal “Anna” // not desired item // move on • Access “Anna” / studentnames[1] • …does equal “Anna” // stop // item found	4 AO2 1b(4)	Answer must refer to this array, not a generic description of linear search. “Access first item” is NE for BP1 or BP3. Must refer to this scenario. Max 1 for “Compare ‘**Anna**’ to **each item** in list” if nothing else credited.

J276/02 Mark Scheme November 2020

6	**(b)**		• Anna inserted before Rob as first two elements… • …Huw correctly inserted into sorted list… • …Emma correctly inserted into sorted list … • …Patrice correctly inserted into sorted list … • …Iqbal correctly inserted into sorted list and **no further changes** made.	5 AO2 1b(5)	<table><tr><td>**Rob**</td><td>Anna</td><td>Huw</td><td>Emma</td><td>Patrice</td><td>Iqbal</td></tr><tr><td>**Anna**</td><td>Rob</td><td>Huw</td><td>Emma</td><td>Patrice</td><td>Iqbal</td></tr><tr><td>Anna</td><td>**Huw**</td><td>Rob</td><td>Emma</td><td>Patrice</td><td>Iqbal</td></tr><tr><td>Anna</td><td>**Emma**</td><td>Huw</td><td>Rob</td><td>Patrice</td><td>Iqbal</td></tr><tr><td>Anna</td><td>Emma</td><td>Huw</td><td>**Patrice**</td><td>Rob</td><td>Iqbal</td></tr><tr><td>Anna</td><td>Emma</td><td>Huw</td><td>**Iqbal**</td><td>Patrice</td><td>Rob</td></tr></table> Sorted list highlighted

6	**(c)**		• Use of iteration (any use) …
			• ...loops for each item in array // loops 6 times
			• …to print out each item in `studentnames`
			• …input attendance
			• Add up/calculate students present **and** absent
			• …Outputs present **and** absent (in suitable message)
				6
				
				AO3
				2b(6)
					BP 2 and 3 may be met together with suitable input statement. Both dependent on attempt at iteration.
					
					BP5 not dependent on correct previous parts.
					
					BP6 needs reasonable attempt at totalling present and absent figures.
					
					Ignore non-initialisation of counter variables.
					
					Flowcharts are acceptable but must show **how** to solve the problem, not simply repeat the question.
					
					<u>Example algorithm</u>
					
					`present=0`
					`absent=0`
					`for i = 0 to (studentnames.length) -1`
					`  print(studentnames[i])`
					`  attendance=input("absent or present?")`
					`  if attendance=="present" then`
					`  present=present+1`
					`  else`
					`  absent=absent+1`
					`  endif`
					`next i`
					`print ("Present students: " + present)`
					`print ("Absent students: " + absent)`

PMT

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

PMT

GCSE

Computer Science

J277/02: Computational thinking, algorithms and programming

General Certificate of Secondary Education

Mark Scheme for June 2024

Oxford Cambridge and RSA Examinations

PMT

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 2024

Oxford Cambridge and RSA Examinations

J277/02 Mark Scheme June 2024

MARKING INSTRUCTIONS

PREPARATION FOR MARKING
RM ASSESSOR

1. Make sure that you have accessed and completed the relevant training packages for on-screen marking: *RM Assessor Assessor Online Training*; *OCR Essential Guide to Marking*.

2. Make sure that you have read and understood the mark scheme and the question paper for this unit. These are posted on the RM Cambridge Assessment Support Portal http://www.rm.com/support/ca

3. Log-in to RM Assessor and mark the **required number** of practice responses ("scripts") and the **number of required** standardisation responses.

   YOU MUST MARK 10 STANDARDISATION RESPONSES BEFORE YOU CAN BE APPROVED TO MARK LIVE SCRIPTS.

MARKING

1. Mark strictly to the mark scheme.

2. Marks awarded must relate directly to the marking criteria.

3. The schedule of dates is very important. It is essential that you meet the RM Assessor 50% and 100% (traditional 40% Batch 1 and 100% Batch 2) deadlines. If you experience problems, you must contact your Team Leader (Supervisor) without delay.

4. If you are in any doubt about applying the mark scheme, consult your Team Leader by telephone or the RM Assessor messaging system, or by email.

5. **Crossed Out Responses**
   Where a candidate has crossed out a response and provided a clear alternative then the crossed out response is not marked. Where no alternative response has been provided, examiners may give candidates the benefit of the doubt and mark the crossed out response where legible.

   Rubric Error Responses – Optional Questions
   Where candidates have a choice of question across a whole paper or a whole section and have provided more answers than required, then all responses are marked and the highest mark allowable within the rubric is given. Enter a mark for each question answered into RM assessor, which will select the highest mark from those awarded. *(The underlying assumption is that the candidate has penalised themselves by attempting more questions than necessary in the time allowed.)*

J277/02 Mark Scheme June 2024

Multiple Choice Question Responses
When a multiple choice question has only a single, correct response and a candidate provides two responses (even if one of these responses is correct), then no mark should be awarded (as it is not possible to determine which was the first response selected by the candidate).
When a question requires candidates to select more than one option/multiple options, then local marking arrangements need to ensure consistency of approach.

Contradictory Responses
When a candidate provides contradictory responses, then no mark should be awarded, even if one of the answers is correct.

Short Answer Questions (requiring only a list by way of a response, usually worth only **one mark per response**)
Where candidates are required to provide a set number of short answer responses then only the set number of responses should be marked. The response space should be marked from left to right on each line and then line by line until the required number of responses have been considered. The remaining responses should not then be marked. Examiners will have to apply judgement as to whether a 'second response' on a line is a development of the 'first response', rather than a separate, discrete response. *(The underlying assumption is that the candidate is attempting to hedge their bets and therefore getting undue benefit rather than engaging with the question and giving the most relevant/correct responses.)*

Short Answer Questions (requiring a more developed response, worth **two or more marks**)
If the candidates are required to provide a description of, say, three items or factors and four items or factors are provided, then mark on a similar basis – that is downwards (as it is unlikely in this situation that a candidate will provide more than one response in each section of the response space.)

Longer Answer Questions (requiring a developed response)
Where candidates have provided two (or more) responses to a medium or high tariff question which only required a single (developed) response and not crossed out the first response, then only the first response should be marked. Examiners will need to apply professional judgement as to whether the second (or a subsequent) response is a 'new start' or simply a poorly expressed continuation of the first response.

6. Always check the pages (and additional objects if present) at the end of the response in case any answers have been continued there. If the candidate has continued an answer there, then add a tick to confirm that the work has been seen.

J277/02 Mark Scheme June 2024

7. Award No Response (NR) if:
   - there is no attempt to answer the question (including blank responses and comments such as "I don't know")

   Award Zero '0' if:
   - An attempt is made in the answer space but this is not worthy of credit.

   Team Leaders must confirm the correct use of the NR button with their markers before live marking commences and should check this when reviewing scripts.

8. The RM Assessor **comments box** is used by your team leader to explain the marking of the practice responses. Please refer to these comments when checking your practice responses. **Do not use the comments box for any other reason.**
   If you have any questions or comments for your team leader, use the phone, the RM Assessor messaging system, or e-mail.

9. Assistant Examiners will send a brief report on the performance of candidates to their Team Leader (Supervisor) via email by the end of the marking period. The report should contain notes on particular strengths displayed as well as common errors or weaknesses. Constructive criticism of the question paper/mark scheme is also appreciated.

10. For answers marked by levels of response (not applicable in J277/02):
    a. **To determine the level** – start at the highest level and work down until you reach the level that matches the answer
    b. **To determine the mark within the level**, consider the following

Descriptor	Award mark
On the borderline of this level and the one below	At bottom of level
Just enough achievement on balance for this level	Above bottom and either below middle or at middle of level (depending on number of marks available)
Meets the criteria but with some slight inconsistency	Above middle and either below top of level or at middle of level (depending on number of marks available)
Consistently meets the criteria for this level	At top of level

11. Annotations

Annotation	Meaning
	Omission mark
	Benefit of doubt (must be accompanied with a tick)
	Cross
	Follow through (must be accompanied with a tick)
	Not answered question
	Benefit of doubt not given
	Repeat
	Tick
	Too vague
	Blank pages, pages with no annotation, no attempt to answer the question, page seen on QER

12. **Subject Specific Marking Instructions**

Mark scheme conventions:

- Each mark point is worth 1 mark unless stated otherwise
- Each mark point can only be awarded once
- A word/phrase that is underlined needs to be exact in the answer to award the mark point
- A word/phrase that is **bold** needs that concept to be in the answer (but can be given in multiple ways) to award the mark point
- 3 dots at the end of one mark point and at the start of the next mark point mean that the second mark point cannot be awarded without the first being awarded, unless the mark scheme states otherwise (for example a reasonable attempt with some inaccuracies)
- 3 dots at the start of a mark point, without 3 dots at the end of the mark point above, means the sentence carries on and there is no dependency
- Any text in brackets is not required to gain the mark point
- Single / means alternative word
- Double // means an alternative statement that is acceptable for the same mark point
- Enlarged font is used for visibility reasons only

Annotating scripts:

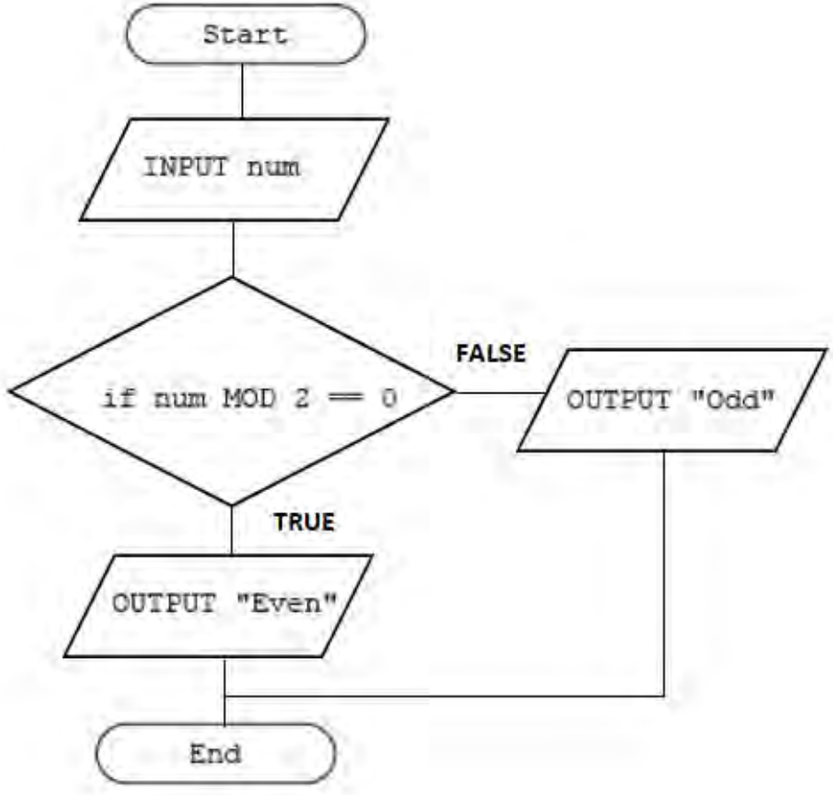
- Blank pages at the start of the script need SEEN annotation
- Any questions answered elsewhere (e.g. on the first blank pages, separately on the page) need to be linked within RM Assessor and annotated with ticks/crosses/SEEN as appropriate
- 1 tick for every mark awarded, if a question is given 3 marks there must be 3 ticks.
- A BOD or FT annotation needs to be accompanied by a tick
- Any answers with no candidate response need a SEEN annotation and NR entered as the mark.
- Any questions where the candidate has not attempted the question e.g. answered 'don't know' need a SEEN annotation and NR entered as the mark.
- All questions must be annotated throughout the marking process.

PMT

J277/02 Mark Scheme June 2024

<table>
<tr><th colspan="3">Question</th><th>Answer</th><th>Mark</th><th>Guidance</th></tr>
<tr><td>1</td><td></td><td></td><td>
<table>
<tr><td rowspan="2">Keyword</td><td colspan="2">Programming construct</td></tr>
<tr><td>selection</td><td>iteration</td></tr>
<tr><td>if</td><td>✓</td><td></td></tr>
<tr><td>for</td><td></td><td>✓</td></tr>
<tr><td>while</td><td></td><td>✓</td></tr>
</table>
</td><td>3
(AO1)</td><td></td></tr>
</table>

J277/02 Mark Scheme June 2024

2			• Correct shape for **all three** inputs AND outputs (parallelogram) • Correct shape for decision (diamond) • True and False // Yes and No labelled **correctly** *(true/Yes linking to "Even")* • **All** lines joined up correctly and link to End. 	4 (AO2)	No need for arrows – lines are acceptable. BOD for correct answers that include a loop back to the start

J277/02 Mark Scheme June 2024

3	(a)		Max 1 mark for definition that is clearly different from a logic error. • (an error that) breaks the **rules/grammar** of the **programming language** • Stops the program from **running** // does not allow program to **run** // crashes the program // does not allow program to **translate** Suitable example for 1 mark, e.g. • misspelling **key word** (e.g. **`printt`** instead of **`print`**) • Missing / extra symbol (e.g. missing bracket, missing semicolon) • Mismatched quotes • Invalid variable or function names (e.g. variable starting with a number or including a space) • Incorrect use of operators • Use of reserved keywords for variables (e.g. **`print = 3`**) • Incorrect capitalisation of keywords (e.g. **`P`**`rint` instead of `print`) • Incorrect indentation (of code blocks) • Missing concatenation (e.g. `print(score x)`)	2 (AO1)	BOD code/program etc for BP1 Do not allow answers linked to data types. "incorrect grammar" by itself is NE Do not allow “stop working”, "does not work", etc – TV. Do not accept missing quotation marks e.g. `print(hello)` (could be a variable name) BOD given code that could cause a syntax error in a high-level language.

J277/02 Mark Scheme June 2024

3	**(b)**		1 mark each
			• line 03
			• `total = num1 + num2`
			
			• Line 04
			• `if total >= 10 and total <=20 then`
			
			Allow other logical equivalent code e.g.
			`total = int(num1) + int(num2)`
			`if 10 <= total <= 20`
				4 (AO3)	Allow other logical corrections that will fix the problem identified and does not introduce any further errors.
					
					Allow descriptions of changes as long as clear exactly what will change. Do not allow ambiguous descriptions of changes to code.
					
					Ignore missing `then` from line 04.
					
					Ignore capitalisation.

3	**(c)**	**(i)**	1 mark each • Compare to / pick out middle value (which is 6) • **discard only left side // retain only right side** (because 6 < 10)… • …Compare to / pick out (middle value which is) **10**	3 (AO2)	BP1 can be given for generic answer. BP2 and 3 must be linked to data set given For BP2, must remove 1, 2,5 and 6 from list if discussing individual numbers. Allow FT for BP3 if this done incorrectly.

		(ii)	• Data must be sorted / in order	1 (AO1)	
		(iii)	• Merge sort	1 (AO1)	
4	**(a)**		Input e.g. • **Name / keyword** for video (to be searched for) // search **text** • **Controls** for watching video (e.g. play / pause) • **Rating** given to video Output e.g. • Video to be watched // audio • **Results** of search • (total / overall / average) rating of video • Number of views (of video) • Confirmation of data entry / data validity • Messages to user // example messages (e.g "enter a rating", "your rating has been saved") in **quotation marks**	2 (AO1)	1 mark for a suitable input, 1 mark for a suitable output Allow input / print pseudocode statements if meets mark point(s). Does not have to be valid pseudocode. Do not allow examples of inputs (e.g. "music videos")

J277/02 Mark Scheme June 2024

4	**(b)**		Only 1 method asked for. Could be name and description/example or description and example • Authentication • …checking users allowed to access the site / know identity of users • … by example (e.g. username and password) • Anticipating misuse // preventing misuse • ….stopping the user breaking / hacking into the system • …by example (e.g. restricting entry to integers) • Validation • …check / only allow sensible data to be entered / check data is sensible • …by example (e.g. restrict ratings to 1 to 10 / presence check / format check) • Input sanitisation • …removing invalid/special characters • …by example (e.g. remove quotation marks / semicolons) • Maintainability • ...ensuring program is able to be understood by others • ...by example (e.g. modularisation / comments)	2 (AO1)	Allow validation / input sanitisation / passwords as expansion of anticipating misuse. Allow mark for description with no / incorrect name Allow any 2 points from mark scheme as long as clearly linked to a single defensive design method.

PMT

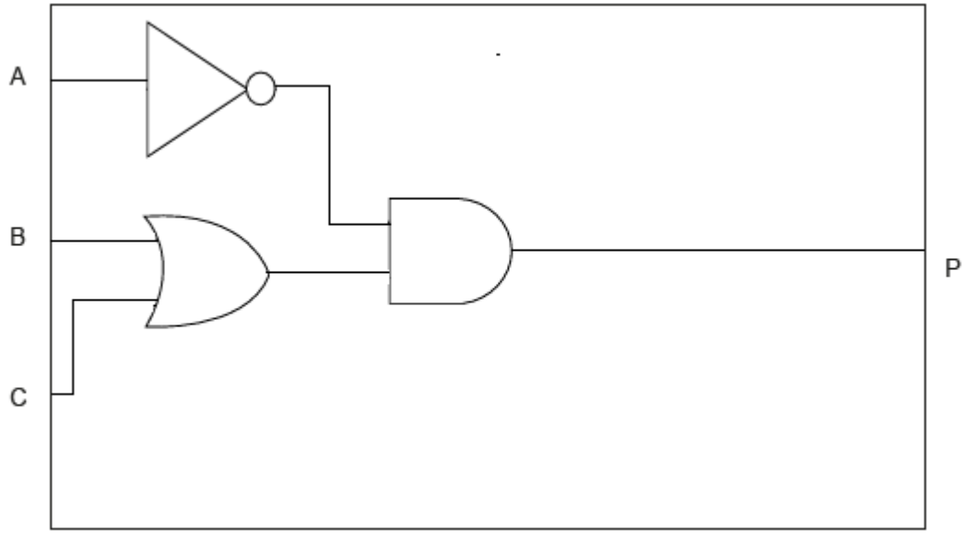
J277/02 Mark Scheme June 2024

5	**(a)**		1 mark per group of 2 rows	4 (AO2)	Accept True / False etc.

A	B	C	P	
0	0	0	**0**	1 mark
0	0	1	**1**	
0	1	0	**0**	1 mark
0	1	1	**1**	
1	0	0	**0**	1 mark
1	0	1	**1**	
1	1	0	**1**	1 mark
1	1	1	**1**	

PMT

J277/02 Mark Scheme June 2024

5	**(b)**		1 mark each • NOT A • B OR C • AND gate with **two inputs**  	3 (AO3)	**Max 2** if not logically correct or any additional / missing gates. Shapes of gates must be correct with correct number of inputs. Ignore annotation of gate names. NOT gate must include circle. Other gates must not include circle.

J277/02 Mark Scheme June 2024

6	(a)		1 mark for each output `print(message.upper)` — **ABCD1234** (upper case) `print(message.left(4))` — **abcd** (lower case) `print(int(message.right(4))*2)` — **2468**	3 (AO2)	Case must be correct but BOD if ambiguous. Allow quotation marks in answer.

6	(b)		1 mark per bullet point :
			• storing both strings correctly in `word1` and `word2`
			• correct concatenation (`word1` **then** `word2`)...
			• …storing in variable `message`
			
			Example
			`word1 = "Hello"`
			`word2 = "Everyone"`
			`message = word1 + word2`
				3 (AO3)	Accept & / + / . etc as valid methods of concatenation. Allow use of sensible concatenation functions e.g. concat() . **Do not allow commas**.
					
					Do not allow == for assigning value to string. Do not allow spaces in variable names. Penalise once then FT.
					
					Ignore additional code given. Ignore case.
					
					Reasonable attempt at BP2 needed to access BP3.

J277/02 Mark Scheme June 2024

7	(a)		1 mark each to max 2 • (machine code) does not need to be translated / compiled / interpreted • Direct control of **hardware / memory** • Faster execution time • Code can be optimised / shorter code / use less memory • Can program for specific hardware • **Assembly language** is fast to translate.	2 (AO1)	"More efficient" by itself is TV. Mark first answer on each line. BP6 relates to Assembly language being a one-to-one direct mapping to machine code.
7	(b)		1 mark each to max 3 • Can produce an **executable file** • program/code **runs/executes** faster (than interpreted version) • end users do not need translator • Can be run **again/multiple times** without re-translating // only needs to translate **once** • **End users** have no access to **source code** // distributed **with no source code**... • …cannot steal/copy/modify code/program • Shows **all/multiple** errors // shows errors **at the end** (of compilation) // creates **error file** • Compiler can optimise the code	3 (AO1)	Allow in reverse (e.g. "interpreter translates every time") Do not allow "no access to source code" unless clearly talking about end user. Allow if in context of distribution. Do not allow descriptions of how a compiler translates (e.g. "translates whole code in one go") "Faster / quicker" by itself is TV

PMT

J277/02 Mark Scheme June 2024

8	(a)		2 marks max per group • Meaningful **identifiers** // meaningful variable names • ...to describe/show what they **store // purpose** of variable • An example of a meaningful variable identifier for this algorithm • Comments • ...to make it easier for other programmers to **follow** / **understand** (part of) the code // explains what the code does // easier to **debug** • An example of a suitable comment for this algorithm • Use of subroutines • ...to **reuse** blocks of code // make code easier to follow • An example of a subroutine for this algorithm • Use of constants • ...to store data that will not change (during program execution) // so data can be changed in one place only • An example of a constant for this algorithm (e.g. store 512 as a constant)	4 (AO2)	Do not accept "what variables do" – incorrect verb, variables store/hold data. BOD notes (and alternatives) for comments. Do not allow instructions. Do not allow indentation (already done in program given) Allow whitespace / **blank** lines (same expansions as comments) Do not award expansion without being clear which method is being discussed. "Makes it easier to understand" by itself is TV.

J277/02 Mark Scheme June 2024

8	(b)		1 mark each to max 6
			• Appropriate use of **both parameters** and **no additional inputs** / **incorrect overwrites** that affect outcome of algorithm
			• Attempt at selection…
			• …correctly checking if `direction` is "left" **and** subtracting 5 from `position` (or equivalent)
			• …correctly checking if `direction` is "right" **and** adding 5 to `position` (or equivalent)
			• Ensuring position (or equivalent) is between 1 and 512 inclusive
			• **Returning** the updated position
			
			Example
			`if direction == "left" then`
			`  position = position - 5`
			`elseif direction == "right" then`
			`  position = position + 5`
			`endif`
			
			`if position < 1 then`
			`  position = 1`
			`elseif position > 512 then`
			`  position = 512`
			`endif`
			
			`return position`
				6 (AO3)	Allow `else` for BP3/4 (validated in question 8a)
					
					Allow <=, >= and equivalents (e.g. <= 0) for BP5.
					
					Do not award BP5 if before BP3 and 4 (otherwise will alter position value)
					
					BP6 only to be given if attempt made at calculating new position. Calculation can be partial/incorrect.
					
					Ignore repeat of function header / end.
					
					Accept flowchart / structured English but must not just repeat the question.
					
					If response uses loop to incorrectly change position multiple times, do not award BP1 (incorrect overwrite)
					
					For minor syntax errors (e.g. missing quotation marks or == for assignment, spaces in variable names) penalise once then FT.

J277/02 Mark Scheme June 2024

Section B

Question			Answer	Mark	Guidance
9	(a)	(i)	• String • Integer • Real / Float	3 (AO3)	Accept alternative equivalent correct data types (e.g. single/double/decimal for BP3) Do not accept char for BP1
		(ii)	• `theTeam.length() - 1 // 5` • `count` • `studentName` • `True`	4 (AO3)	Accept 6 // `theTeam.length()` for BP1 (Python). Accept alternative length functions e.g. len() Accept **`count = 5`** (and equivalents) for BP1. Accept "True" for BP4. Do not allow obvious spaces in variable names. Ignore capitalisation.

J277/02 Mark Scheme June 2024

<table>
<tr><th colspan="3">Question</th><th>Answer</th><th>Mark</th><th>Guidance</th></tr>
<tr><td></td><td>(b)</td><td></td><td>

<code>javelinThrow</code> set to 14.3 on line 01 and <code>yearGroup</code> set to 10 on line 02
<code>score</code> set to 2 on line 06
<code>score</code> set to 4 on line 11
"The score is 4" output on line 13 with no additional outputs (allow input statements)

Example
<table>
<tr><th>Line number</th><th>javelinThrow</th><th>yearGroup</th><th>score</th><th>Output</th></tr>
<tr><td>01</td><td>14.3</td><td></td><td></td><td></td></tr>
<tr><td>02</td><td></td><td>10</td><td></td><td></td></tr>
<tr><td>06</td><td></td><td></td><td>2</td><td></td></tr>
<tr><td>11</td><td></td><td></td><td>4</td><td></td></tr>
<tr><td>13</td><td></td><td></td><td></td><td>The score is 4</td></tr>
</table>
Answer may include lines where no changes or output happens (i.e. lines 3, 4, 5, 7, 8, 9, 10, 12).

Where variable doesn't change, current value may be repeated on subsequent lines.
</td><td>4 (AO3)</td><td>Max 3 if in wrong order or additional (incorrect) changes. Penalise line numbers once then FT.

Allow FT for BP4 for current value of <code>score</code>.

BP4 must not include comma. Ignore superfluous spaces. Ignore quotation marks.

Treat any entry in output column as an output, even if "x", "-" or "0".</td></tr>
</table>

PMT

J277/02 Mark Scheme June 2024

Question			Answer	Mark	Guidance

	(c)	(i)	• inputs a value from the user and stores/uses
			• checks min value (>= 40.0 // < 40)
			• checks max value (<=180.0 // > 180)
			• …outputs **both** valid / not valid correctly based on checks
			
			Example 1 (checking for valid input)
			`h = input("Enter height jumped")`
			`if h >= 40 and h <= 180 then`
			`  print("valid")`
			`else`
			`  print("not valid")`
			`endif`
			
			Example 2 (checking for invalid input)
			`h = input("Enter height jumped")`
			`if h < 40 or h > 180 then`
			`  print("not valid")`
			`else`
			`  print("valid")`
			`endif`
				4 (AO3)	Answers using `AND/OR` for BP2 and BP3 must be logically correct e.g. `if height >=40 and height <=180.` Do not accept `if height >=40 and <=180`
					
					Answers using OR will reverse output for BP4 (see examples).
					
					BP4 needs reasonable attempt at **either** BP2 or BP3. Need to be sure what is being checked to be able to decide which way around valid/invalid should be.
					
					Allow FT for BP4 if reasonable attempt at validating (must include at least one boundary)
					
					Ignore conversion to int on input. `input` cannot be used as a variable name.
					
					Greater than / less than symbols must be appropriate for a high-level language / ERL. Do not accept => (wrong way around) or ≥ (not available on keyboard). No obvious spaces in variable names. Penalise once and then FT.

J277/02 Mark Scheme June 2024

Question			Answer	Mark	Guidance
	(c)	(ii)	• Any normal value (between 40 and 180 inclusive) • 40.0 // 180.0 • Any value less then 40 // any value greater that 180 // any non-numeric value	3 (AO3)	No need to include decimals, e.g. accept 50. Ignore cm if given. Answer must be actual data (e.g. 50) and **not** description of data (e.g. "a value between 40 and 180"). If descriptions given, do not accept this as non-numeric for BP3
	(d)		• `TeamName` only in first space • `TblResult` in second space • `WHERE` • `...YearGroup = 11`	4 (AO3)	Max 3 if not in correct order / includes other logical errors. Ignore capitals. Do not accept * or additional fields for BP1 Spelling must be accurate (e.g. not TblResult**s**). No spaces in field names, penalise obvious spaces once and then FT. Allow quotation marks around field names, table name and 11 Accept == for BP4 (invalid SQL but works in some environments)

J277/02 Mark Scheme June 2024

Question			Answer	Mark	Guidance
	(e)	(i)	• any **example** of simplification/removing data or focussing on data (in the design) Examples : - "focus on student names and events" - "ignore data such as students' favourite subjects" - "store year groups instead of ages or DOB" - "shows student IDs instead of full student details"	1 (AO3)	Must be applicable to this program (in the context of students and a sports day), **not** a generic description of what abstraction is. Give BOD where this is unclear.
		(ii)	• any **example** of breaking down the **program** into sections/subroutines • any **example** of breaking down the **database** into tables Examples : - "splits the program up into different events" - "separates the validation routines into subroutines" - "breaks the database down into a table per event"	1 (AO3)	Must be applicable to this program, not a generic description of what decomposition is. Give BOD where this is unclear. Do not give answers discussing splitting into **fields** (e.g. split into StudentID, YearGroup, etc). BOD if answer discusses one table but suggests other tables could be used. Do not give answers relating simply to data being split into smaller groups unless this clearly relates to how data is decomposed into **tables** in the DB. Allow reference to sports day to mean sports day program.

J277/02 Mark Scheme June 2024

Question			Answer	Mark	Guidance

	(f)		• Input team name AND score **and** store / use separately
			• Attempt at using iteration…
			• …to enter team/score until "stop" entered
			• Calculates highest score
			• Calculates winning team name…
			• …Outputs highest score **and** team name
			
			Example 1
			`highscore = 0`
			`while team != "stop"`
			`  team = input("enter team name")`
			`  score = input("enter score")`
			`  if score > highscore then`
			`  highscore = score`
			`  highteam = team`
			`  endif`
			`endwhile`
			`print(highscore)`
			`print(highteam)`
			
			Example 2 (alternative)
			`scores = []`
			`teams = []`
			`while team != "stop"`
			`  team = input("enter team name")`
			`  score = input("enter score")`
			`  scores.append(score)`
			`  teams.append(team)`
			`endwhile`
			`highscore = max[scores]`
			`highteam = teams[scores.index(highscore)]`
				6 (AO3)	For BP3, allow "stop" to be entered for either team name or score (or both). Allow third input (e.g. "do you wish to stop?")
					
					Allow use of `break` (or equivalent) to exit loop for BP3.
					
					Allow use of recursive function(s) for BP2/3.
					
					Initialisation of variables not needed - assume variables are 0 or empty string if not set.
					
					Ignore that multiple teams could get the same high score, assume only one team has the highest score.
					
					BP4/5 could be done in many ways – see examples. Allow any logically valid method. Allow use of max/sum functions and use of arrays/lists.
					
					FT for BP6 if attempt made at calculating highest score/name
					
					If answer simply asks for multiple entries (not using iteration), BP2 and 3 cannot be accessed but all others available.

PMT

J277/02 Mark Scheme June 2024

Question			Answer	Mark	Guidance
			`print(highscore)` `print(highteam)`		For minor syntax errors (e.g. missing quotation marks or == for assignment, spaces in variable names) penalise once then FT. `input` cannot be used as a variable name.

PMT

Need to get in touch?

If you ever have any questions about OCR qualifications or services (including administration, logistics and teaching) please feel free to get in touch with our customer support centre.

Call us on

01223 553998

Alternatively, you can email us on

support@ocr.org.uk

For more information visit

ocr.org.uk/qualifications/resource-finder
ocr.org.uk
Twitter/ocrexams
/ocrexams
/company/ocr
/ocrexams

CAMBRIDGE
UNIVERSITY PRESS & ASSESSMENT

OCR is part of Cambridge University Press & Assessment, a department of the University of Cambridge.

For staff training purposes and as part of our quality assurance programme your call may be recorded or monitored. © OCR 2024 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 operates academic and vocational qualifications regulated by Ofqual, Qualifications Wales and CCEA as listed in their qualifications registers including A Levels, GCSEs, Cambridge Technicals and Cambridge Nationals.

OCR provides resources to help you deliver our qualifications. These resources do not represent any particular teaching method we expect you to use. We update our resources regularly and aim to make sure content is accurate but please check the OCR website so that you have the most up-to-date version. OCR cannot be held responsible for any errors or omissions in these resources.

Though we make every effort to check our resources, there may be contradictions between published support and the specification, so it is important that you always use information in the latest specification. We indicate any specification changes within the document itself, change the version number and provide a summary of the changes. If you do notice a discrepancy between the specification and a resource, please contact us.

Whether you already offer OCR qualifications, are new to OCR or are thinking about switching, you can request more information using our Expression of Interest form.

Please get in touch if you want to discuss the accessibility of resources we offer to support you in delivering our qualifications.

Question 2 (a) (i)

2 A fast food restaurant offers half-price meals if the customer is a student or has a discount card. The offer is not valid on Saturdays.

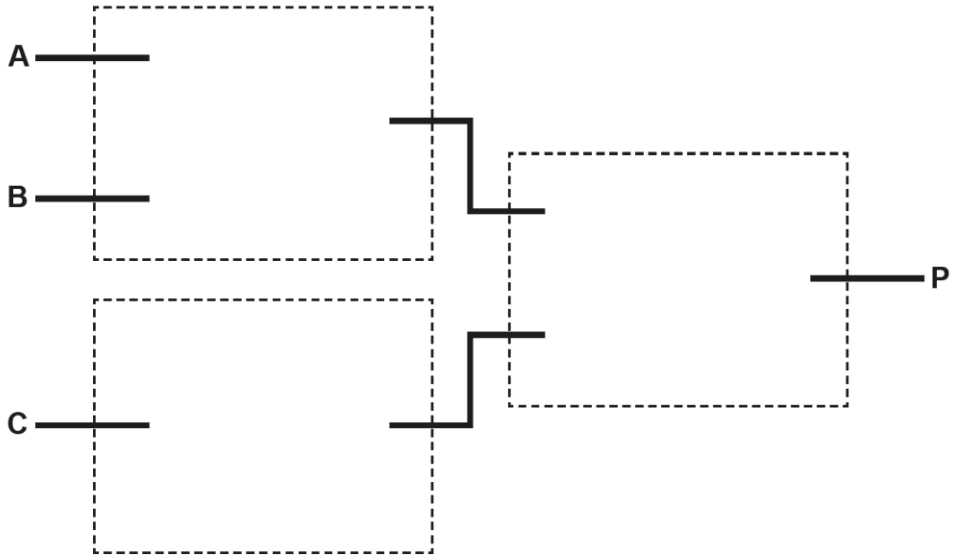
A computer system is used to identify whether the customer can have a half-price meal.

The table identifies the three inputs to the computer system:

Input	Value
A	Is a student
B	Has a discount card
C	The current day is Saturday

(a) The logic system **P = (A OR B) AND NOT C** is used.

(i) Complete the following logic diagram for **P = (A OR B) AND NOT C** by drawing one logic gate in each box.



[3]

This question was answered extremely well. Candidates were familiar with the logic symbols required for each gate.

A few candidates missed the circle from the NOT gate symbol (which turned this into a buffer instead) or added a circle to the AND and OR gates. This turned them into NAND and NOR gates. These logic gates are not on the GCSE specification, but are covered at A Level. Teachers may find it beneficial to briefly discuss these in order to anticipate mistakes like these.

A small number of candidates drew indistinct logic gate symbols and named them instead, such as a rectangle with AND written in; this was not credited with a mark.

Question 2 (a) (ii)

(ii) A truth table can be produced for this logic circuit.

Describe the purpose of a truth table.

.....

.....

.....

..... **[2]**

Question 2 (a) (iii)

(iii) State how many rows (excluding any headings) would be required in a truth table for the logic expression:

P = (A OR B) AND NOT C

..... **[1]**

It is clear that candidates have experienced and used truth tables with logic circuits. However, the majority were not able to describe the truth table's purpose with precision.

A truth table defines the expected outputs for a logic circuit depending on the inputs given. Furthermore, the truth table covers all possible permutations of inputs.

A logic circuit with three inputs (A,B and C) will have **8** possible rows in a truth table to cover the 8 possible ways that True and False values for three inputs can be arranged.

Misconception

? The number of rows in a truth table depends on the number of inputs. The number of rows can be given by 2^x, where x is the number of inputs. Therefore, a truth table with 3 inputs would have 2^3 rows, or 2 x 2 x 2 = 8 rows.

Exemplar 1

To show what output happens for every variation of inputs, and check they are all what was intended

Here the candidate has achieved both marks available; it is clear that every variation of input is included in the truth table and the output is very clearly linked to these inputs.

Question 2 (b)

(b) The restaurant needs an algorithm designing to help employees work out if a customer can have a half price meal or not. It should:

- input required data
- decide if the customer is entitled to a discount
- output the result of the calculation.

Design the algorithm using a flowchart.

[5]

Most candidates who answered this question were comfortable using the correct flowchart symbols listed in the specification. A mark was available for including suitable start / end symbols and connecting all other symbols, so even candidates who may struggle should feel confident of being able to access some marks.

Marks were dropped when responses did not include suitable inputs to the algorithm. The first bullet point in the question stem was clear that these were required.

Candidates achieving lower scores tended to group decisions so that any processing was removed (such as “can they have a discount?”). More successful responses decomposed the problem and used a succession of smaller decisions (“do they have a discount card?”, “are they a student?”, “is it Saturday?”). These decisions then point towards the correct outputs.

Question 2 (c)

(c) The restaurant adds a service charge to the cost of a meal depending on the number of people at a table. If there are more than five people 5% is added to the total cost of each meal.

Customers can also choose to leave a tip, this is optional and the customer can choose between a percentage of the cost, or a set amount.

Identify **all** the additional inputs that will be required for this change to the algorithm.

.....

.....

.....

..... **[2]**

This question type is new to the J277 specification and asked candidates to list inputs that will be required as part of the planning stage for an algorithm. Successful candidates were able to identify the raw data needed from the user in order to be able to solve the problem.

Unsuccessful responses tended to simply rewrite the question or miss out key information.

Question 2 (d) (i)

(d) Each member of staff that works in the restaurant is given a Staff ID. This is calculated using the following algorithm.

```
01  surname = input("Enter surname")
02  year = input("Enter starting year")
03  staffID = surname + str(year)
04  while staffID.length < 10
05       staffID = staffID + "x"
06  endwhile
07  print("ID " + staffID)
```

(i) Define the term **casting and** give the line number where casting has been used in the algorithm.

Definition .....

.....

Line number .....

[2]

Many candidates correctly defined casting as changing data from one data type to another. Some candidates defined this term as changing a variable from an integer to a string, which is only one example of what can be done and not a definition.

The majority of candidates then gave the correct line number (line 03) for there this was shown the example code given.

Question 2 (d) (ii)

(ii) Complete the following trace table for the given algorithm when the surname "Kofi" and the year 2021 are entered.

You may not need to use all rows in the table.

Line number	surname	year	staffID	Output
01	Kofi			
02		2021		

[4]

This question asked candidates to trace through a given algorithm to show the value of three variables at various points in the algorithm.

The algorithm itself was relatively simple. It used condition-controlled iteration to repeat while the length of the username was less than 10 characters.

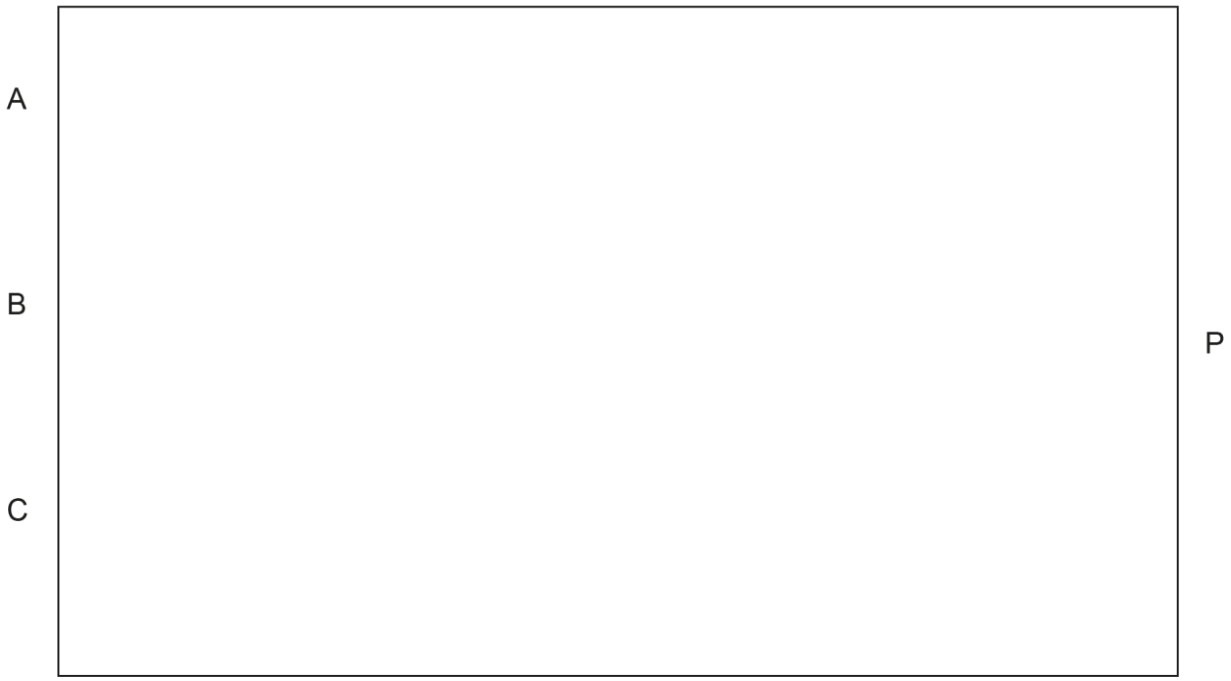
Most candidates gained the first 2 marks for the initial changes to `staffID`. However few candidates were able to trace through the iteration and conclude that the final username should end up as **`ID Kofi2021xx`**.

Marking this question considered the spaces within the username at various points. The algorithm results in one space only, in between **`ID`** and **`Kofi2021xx`**. Where extra spaces appeared or were missed, this was penalised. However, examiners were instructed to give clear benefit of doubt, and to only do this if the space was clearly present/missing.

It is important to understand that "`ab`" and "`a b`" are two strings that are not the same. This level of precision should be encouraged within GCSE Computer Science. Experience of practical programming will help reinforce the impact of spaces within programming and algorithms.

Question 5 (b)

(b) Draw a logic circuit for **P = NOT A AND (B OR C)**



[3]

Drawing logic circuits is now a commonly asked question and candidates are generally competent at doing this. Where issues did arise, they tended to be missing the circle from the NOT gate (and therefore turning it into a buffer, not a logic gate) or including the incorrect number of inputs into a gate.

Key point – logic gates

Candidates are expected to be able to draw the correct shapes for each gate. A number of candidates labelled their gates up, but this is not necessary; examiners are instructed to mark the shape of each gate and ignore any labelling.

Candidates are **not** allowed to take stencils or other tools that allow them to more easily draw these gates into an examination unless as part of a specific access arrangement agreed by OCR.