GCSE

Computer Science

J277/02: Computational thinking, algorithms and programming

General Certificate of Secondary Education

Mark Scheme for June 2022

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.

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 5 PRACTICE AND 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.

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 SEEN annotation to confirm that the work has been seen.

7. Award No Response (NR) if:

- there is nothing written in the answer space

Award Zero '0' if:

- anything is written in the answer space and is not worthy of credit (this includes text and symbols).

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

11. **Annotations**

Annotation	Meaning
	Omission mark
	Benefit of doubt
	Cross
	Follow through
	Not answered question
	Benefit of doubt not given
	Repeat
	Tick
	Too vague
	Noted but credit not given, blank pages, pages with no annotation

<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 per correct row

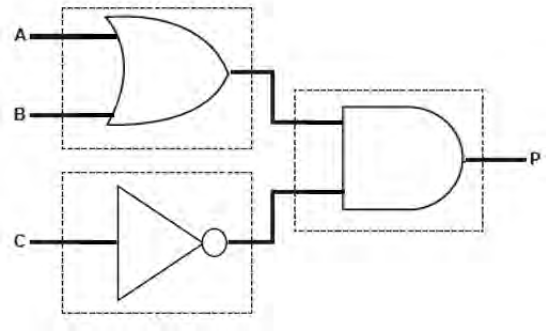
<table>
<tr><th>OCR Reference Language code</th><th>Selection</th><th>Iteration</th></tr>
<tr><td><pre>for i = 1 to 10
    print(i)
next i</pre></td><td></td><td>✓</td></tr>
<tr><td><pre>whilescore != 0
    playgame()
endwhile</pre></td><td></td><td>✓</td></tr>
<tr><td><pre>if playerHit() then
    score = score – 1
endif</pre></td><td>✓</td><td></td></tr>
<tr><td><pre>switch bonus:
    case 0:
        score = 9
    case 1:
        score = 7
    case 2:
        score = 5
endswitch</pre></td><td>✓</td><td></td></tr>
</table></td><td>4
(AO2 1b)</td><td>No mark given if both boxes in a row ticked.

Accept any response (ticks, crosses, etc) that clearly indicates candidate’s choice.</td></tr>
<tr><td>1</td><td>(b)</td><td></td><td>• <code>score = score + 1 // score +=1 // score++</code></td><td>1
(AO3 2b)</td><td>Allow other logically correct answers that result in <code>score</code> increasing by one and being overwritten. Do not accept <code>score + 1 / score = +1</code>

Accept valid structured English answers that refer to score increasing and overwriting the existing value by one. e.g. “score becomes/equals score plus one”

Ignore any superfluous code that does not affect the outcome</td></tr>
<tr><td>1</td><td>(c)</td><td></td><td>• Decomposition</td><td>2</td><td>Correct answer only. Ignore spelling errors.</td></tr>
</table>

			• Abstraction	(AO1 1a)	

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 <u>not</u> 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, gates 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)

2 (b)

- 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

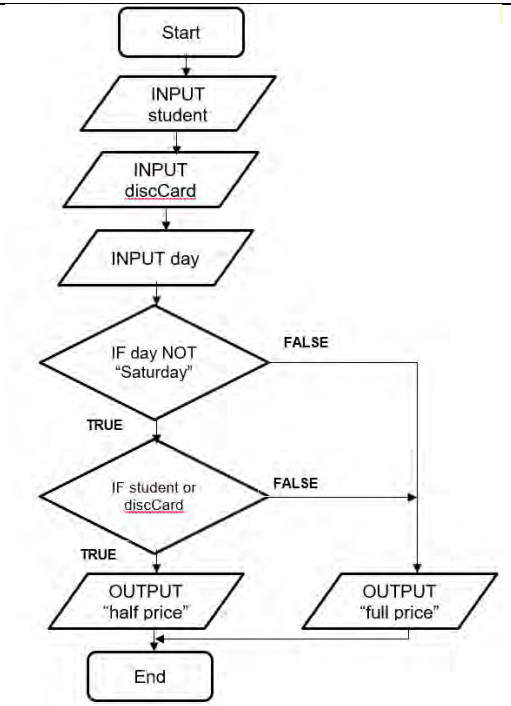
- **…Outputs** "full price" with correct conditions using **parallelogram shape**
- **…Outputs** "half price" with correct conditions using **parallelogram shape**

Guidance for correct outputs

Conditions	Outcome
Not Saturday **and** (either a student or has a discount card).	Half price
Saturday **or** (not a student and doesn't have a discount card).	Full price

Saturday	Student	Discount Card	Outcome
N	N	N	Full price
N	N	Y	**Half price**
N	Y	N	**Half price**
N	Y	Y	**Half price**
Y	N	N	Full price
Y	N	Y	Full price
Y	Y	N	Full price
Y	Y	Y	Full price

5 **(AO3 2a)**



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. Must attempt all three decisions.

Allow calculation of half price / full price instead of message but 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 FT

BOD parallelogram shapes if not sure whether input or output as long as context is clear (e.g inputs at start, outputs at end)

<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 example 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>

Question			Answer	Mark	Guidance
3	(a)		• Merge into correct **sorted** lists of **size 2** (12 45 **/** -99 100 **/** -13 0 **/** -27 17) • Merge into correct **sorted** lists of **size 4** (-99 12 45 100 **/** -27 -13 0 17) … • …Merge into correct **sorted** list of **size 8** (-99 -27 -13 0 12 17 45 100)	3 **(AO2 1b)**	Do not credit BP3 simply for a sorted list. Groups of numbers must clearly be the correct size. Do not all allow answers that show lists being merged and then sorting in place, this is incorrect.
3	(b)		Any four bullet points for 1 mark each • Select / choose / pick **middle** number (or left/right of middle as even number) and … • **…**check if selected number is **equal to / matches** target number *(not just compare)* • …if searched number is **larger**, discard left half // if searched number is **smaller**, discard right half • Repeat until **number found** • … or remaining list is of size 1 / 0 (number not found)	4 **(AO1 1b)**	Do not allow "split the list in half" on its own as first step, this is incorrect. Can get BP1 and 2 in one step (e.g. "check if the middle number is the one we're looking for") For BP3, accept focussing on correct half Repeat (BP4) must be in the context of an attempt at a binary search. Allow correct reference to recursion. "until number is not in the list" is NE for final BP. Need to explain how this is known.
3	(c)		1 mark each • Starting with the **first value** • Checking **all values in order**	2 **(AO1 1b)**	2nd bullet point must cover both ideas of checking all of the values AND being done in order. "Checks each value" / "one by one" / "step by step" by itself is NE, does not say in order. Do not accept "repeat until value found" for BP2 (question says number is not in the list) "Checks each value from beginning to end" implies order so gets both BP1 and BP2.

Question			Answer	Mark	Guidance
4	(a)		Any two bullet points for one mark each: • Add comments • Name variables sensibly • **Put into** subroutine / procedure / function • Use loop / iteration	2 **(AO2 1b)**	Do not accept indentation (no code to sensibly indent in this example) "Use a subroutine" is not enough. Must be clear that existing code will be put into a new subroutine.
4	(b)	(i)	• Multiplication • Division	2 **(AO1 1a)**	Accept other correct answers that mean the same Accept floor / integer division // division with no remainder (Python v2.x)
4	(b)	(ii)	• high-level • stops // crashes • no • executable • without	5 **(AO1 1b, AO2 1b)**	Ignore spelling errors.

4	**(c)**		• input and stores/uses value with message
			• attempt at repeating…
			• …correctly repeats number of times given as input
			• …correctly take number as input within loop and calculates total of these numbers
			• …correctly calculate an average (total/num)
			• Output both total and average
				6
				(AO3 2b, AO3 2c)
					e.g.
					`num = input("Enter how many numbers")`
					`for x = 1 to num`
					`  temp = input("Enter a number")`
					`  total = total + temp`
					`next x`
					`print(total)`
					`print(total / num)`
					
					If flow chart used, correct shapes needed.
					
					Allow tolerance of 1 with number of loops for BP3 with `for` loops
					
					BP1 requires input with a message (can be two statements, e.g. print and then input or combined. Input must be stored or used.
					
					BP3, 4, 5 must be logically correct to be credited
					Ignore non-initialisation of total
					
					BP 5 can be given as FT as long as an attempt has been made at working out a total within the loop.
					
					BP6 can be given as FT long as attempt made at total and average (not necessarily in a loop)

J277/02 **Mark Scheme** **June 2022**
SECTION B

Note – algorithm questions in Section B (5bi, 5ci, 5cii, 5e) require candidates to answer **using OCR Reference Language or a high-level programming language**. Candidates **cannot** answer using structured English or flowcharts and responses of this nature should be marked as NAQ.

Candidates **do not** need to state which language they are using. Because of this, you must not assume use of a particular language. Mark for logical correctness and **not** for the syntax of any particular language. Ignore case sensitivity. Variable names must be correct/consistent.

Question			Answer	Mark	Guidance
5	(a)	(i)	• Integer • String	2 **(AO3 2a)**	Accept other valid data types from high-level languages (e.g. byte, short for integers) Do not accept descriptions (e.g. “whole number”, “text”). Do not accept “character(s)” for string.
5	(a)	(ii)	• stayComplete	1 **(AO3 2a)**	Ignore spaces or misspelling as long as recognisable.
5	(a)	(iii)	• `SELECT FirstName, Surname, Nights, Room, StayComplete // SELECT *` • `FROM TblBookings` • `WHERE` • `Nights > 1 // Nights >= 2 // Nights BETWEEN 2 AND 5`	4 **(AO3 1, AO3 2c)**	Order of fields for BP1 not important but must show all fields and be separated by commas. Ignore capitalisation and spacing. Spelling must be correct. Ignore quotes around numeric values or field/table names. Allow other logically valid SQL statements. Check with TL if required. Ignore reference to `stayComplete` or other valid SQL code that would not affect output. Max 3 if in wrong order or if includes any extra invalid code

5	**(b)**	**(i)**	• Checks that **both** `firstname` and `surname` are not empty…
			• Checks that `room` is either "basic" or "premium"…
			• Checks that `nights` is between 1 and 5 (inclusive)…
			
			• …Outputs "NOT ALLOWED" (or equivalent) if <u>any</u> of the 3 checks are invalid (must check all three)
			• …Outputs "ALLOWED" (or equivalent) <u>only</u> if all three checks are valid (must check all three)
			
			Note : output marks are given for if <u>entire system</u> produces the correct output. For example, If a user enters a valid name and room but an invalid number of nights, the system should say "NOT ALLOWED" (or equivalent). If this works and produces the correct response no matter which input is invalid, BP4 should be given.
			
			The same process holds for the valid output – if (and only if) three valid inputs results in an output saying "ALLOWED" (or equivalent), BP5 should be given. Do not give this if ALLOWED is printed when (for example) two inputs are valid and one is invalid.
			
			For any output marks to be given, a sensible attempt must have been made at all three checks. These may not be completely correct (and may have been penalised in BPs 1 to 3) but should be enough to allow the FT marks for output.
				5
				(AO3 2a)
					Must have some attempt at <u>all three checks</u> to give output mark(s). Check for `nights` must check both upper and lower limits.
					
					Iteration can be used as validation if input repeatedly asked for until valid answer given.
					
					<u>Do not accept</u> logically incorrect Boolean conditions such as
					`if firstname or surname == ""`
					
					Do **not** accept <u>></u> or <u><</u> for >=, <=. Ignore capitalisation
					
					e.g.
					`valid = True`
					`if firstname == "" or surname == "" then`
					`  valid = False`
					`end if`
					`if room != "basic" and room != "premium" then`
					`  valid = False`
					`endif`
					`if nights < 1 or nights > 5 then`
					`  valid = False`
					`endif`
					
					`if valid then`
					`  print("ALLOWED")`
					`else`
					`  print("NOT ALLOWED")`
					`endif`
					
					BP1 to 3 can check for valid or invalid inputs. . Pay particular attention to use of AND / OR. Only give marks for output if these work together correctly.
					
					Example above shows checking for **invalid** data. Checks for **valid** data equally acceptable Examples shown below :
					
					• `if firstname != "" and surname != ""`
					• `if room == "basic" or room == "premium"`
					• `if nights >= 1 and nights <= 5`

<table>
<tr>
<td>5</td><td>(b)</td><td>(ii)</td>
<td>
• Normal
• 1 or 5 (not 0 or 6 as says allowed)
• Any numeric value except 1 to 5 // any non-numeric input (e.g. “bananas”)
</td>
<td>3
(AO3 2c)</td>
<td>
Allow other descriptions that mean normal (e.g. valid / typical / acceptable)
<table>
<tr><th>Test data (number of nights)</th><th>Type of test</th><th>Expected output</th></tr>
<tr><td>2</td><td>Normal</td><td>ALLOWED</td></tr>
<tr><td>1 // 5</td><td>Boundary</td><td>ALLOWED</td></tr>
<tr><td>e.g. 7</td><td>Erroneous/Invalid</td><td>NOT ALLOWED</td></tr>
</table>
</td>
</tr>
<tr>
<td>5</td><td>(c)</td><td>(i)</td>
<td>
• Function header for newPrice…
• …taking (at least) two parameters
• …correctly calculates price based on parameters (if present) within function …
• …returns this calculated price
</td>
<td>4
(AO3 2b)</td>
<td>
BP1 must be clear that a new function is being defined. E.g. <code>function</code> / <code>def</code> keyword. Allow FT for subsequent marks if not present.

Ignore any code outside attempt at function definition.

Ignore additional parameters. Ignore inputs or additional code as long as these do not overwrite parameters or affect operation of function.

If inputs used instead of parameters, FT for BP3. Allow use of <code>else</code> for second room type in BP3.

Attempt at calculation needed to award BP4. Must return (not output) value. Return can be done e.g. in VB by assigning to function name (e.g. <code>newPrice = price</code>)

e.g.
<pre>
function newPrice(nights, room)
  if room == "basic" then
    price = 60 * nights
  elseif room == "premium" then
     price = 80 * nights
  endif
  return price
endfunction
</pre>
</td>
</tr>
</table>

5	**(c)**	**(ii)**	• Call function newPrice…
			• …with (`"premium", 5`) as parameters
			• …Output **returned value**
				3
				(AO3 2b)
					Order of parameters not important
					
					"premium" must use string delimiters (e.g. "quotes")
					
					e.g.
					`print(newPrice("premium", 5))`
					
					`x = newPrice(5, "premium")`
					`print(x)`
					
					Do not allow function definitions for BP1
					
					Ignore capitalisation of `newPrice`
					
					Candidate could store returned value in a variable and then print this, or store parameters in variables before passing in – these are all acceptable
					
					Ignore any superfluous code given
					
					Do not credit answers where `newPrice` is overwritten prior to use.
					
					Ignore spaces. Allow function call if brackets missing (e.g. `newprice` instead of `newprice()`)

5	**(d)**		• For loop changed to **include 0** • total = 0 moved to **before** loop starts / removed	2 **(AO3 2c)**	Allow loop changed to 0 to 8 or 0 to 9 (Python) Do not accept moving total outside loop, NE (could be moved to after loop which would still be a logic error). Do not accept move to top of loop. Accept corrected code shown. Accept reference to count variable limits for BP1.

5	**(e)**		• Inputs hours AND electric (two separate inputs), storing or using these.
			• Checks if car is electric (IF/Select statement)…
			• …correctly calculates and outputs price (hours * 2 // price / 2) for electric
			• …correctly calculates and outputs price (hours * 4 // electric price * 2) for non-electric
			• Attempt at repetition of BP1 to 4…
			• …until 0 hours entered
				6
				
				(AO3 2c)
					Initialisation of price and hours not necessary, but if present hours must be non-zero for BP6 to be given.
					
					BP5 must include all points attempted. Can still be credited if any of BP1 to 4 not attempted / incorrect.
					
					BP6 can be given as FT even if BP5 (loop) is in the wrong place / does not include all required code
					
					BP6 could be achieved as repeated function calls / recursion
					
					Initial input outside of loop that is then <u>also</u> included within loop is fine. For example, input of hours outside of loop but input is then repeated again at end of loop.
					
					Do not accept `while hours > 0` (could be -1)
					
					Do not penalise answers where 0 is output when loop exits
					
					e.g.
					`while hours != 0`
					`  hours = input("Enter hours")`
					`  electric = input("enter Y for electric or N")`
					`  if electric == "Y" then`
					`  price = hours * 2`
					`  elseif electric == "N" then`
					`  price = hours * 4`
					`  endif`
					`  print(price)`
					`endwhile`

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.

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.