Y11 Computer Science (OCR - J277)

Unit 2 - Computational Thinking, Algorithms and Programming

Sample Past Paper Questions

Topic 1

Algorithms

1 There is a subroutine, HEX(), that takes a denary number between 10 and 15 and returns the corresponding hexadecimal number. E.g. HEX(10) would return “A”, HEX(15) would return “F”.

Write an algorithm, using the subroutine HEX(), to convert any whole decimal number between 0 and 255 into a 2 digit hexadecimal number.

[4]

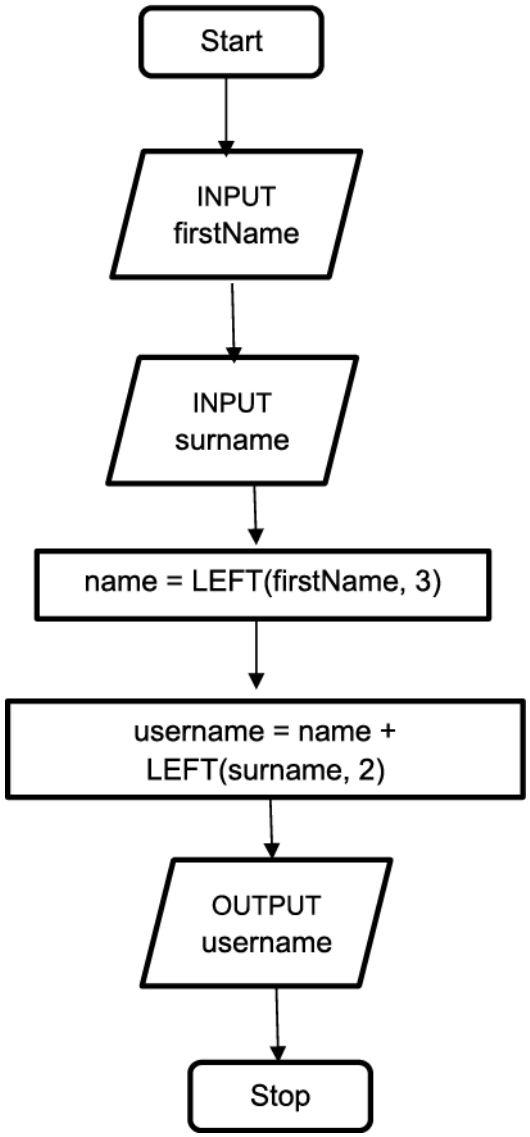
2 Complete the truth table below for the Boolean statement p = NOT (A AND B).

A	B	P
FALSE	FALSE	TRUE
FALSE	TRUE	
TRUE	FALSE	
TRUE	TRUE	FALSE

[2]

3(a) Johnny is writing a program to create usernames. The first process he has developed is shown in the flowchart in **Fig. 1**.

Fig. 1



For example, using the process in **Fig. 1**, Tom Ward's user name would be TomWa.

State, using the process in **Fig. 1**, the username for Rebecca Ellis.

---

[1]

(b) Johnny has updated the process used to create usernames as follows:

- If the person is male, then their username is the last 3 letters of their surname and the first 2 letters of their first name.
- If the person is female, then their username is the first 3 letters of their first name and the first 2 letters of their surname.

• What would be the username for a male called Fred Biscuit using the updated process?

---

[1]

• Write an algorithm for Johnny to output a username using the updated process.

---

[6]

4(a) Heath is researching how long, to the nearest minute, each student in his class spends playing computer games in one week (Monday to Friday). He is storing the data in a 2D array.

Fig. 2 shows part of the array, with 4 students.

Fig. 2

Days of the week	Students			
	0	**1**	**2**	**3**
0	60	30	45	0
1	180	60	0	60
2	200	30	0	20
3	60	10	15	15
4	100	35	30	45

For example, student 1, on Monday (day 0), played 30 minutes of computer games.

Heath wants to output the number of minutes student 3 played computer games on Wednesday (day 2). He writes the code:

```
print (hoursPlayed[3,2])
```

The output is 20.

(i) Write the code to output the number of minutes student 0 played computer games on Wednesday.

---

[1]

(ii) State the output if Heath runs the code:

```
print (hoursPlayed[2,1])
```

---

[1]

(iii) State the output if Heath runs the code:

```
print (hoursPlayed[3,1] + hoursPlayed[3,2])
```

---

[1]

(iv) Write an algorithm to output the total number of minutes student 0 played computer games from Monday (day 0) to Friday (day 4).

---

[3]

(b) Heath needs to work out the average number of minutes spent playing computer games each day for the class, which contains 30 students. Write an algorithm to output the average number of minutes the whole class spends playing computer games each day.

[8]

5(a) Willow has created a hangman program that uses a file to store the words the program can select from. A sample of this data is shown in **Fig. 3**.

Fig. 3

crime	bait	fright	victory	nymph	loose

Show the stages of a bubble sort when applied to data shown in **Fig. 3**.

---

[4]

(b)

A second sample of data is shown in **Fig. 4**.

Fig. 4

amber	house	kick	moose	orange	range	tent	wind	zebra

Show the stages of a binary search to find the word 'zebra' when applied to the data shown in **Fig. 4**.

---

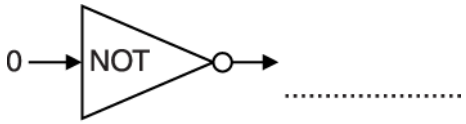
[4]

6 Charley is writing a program for music students. To make sure that there are no logic errors in the program, Charley uses a test plan.

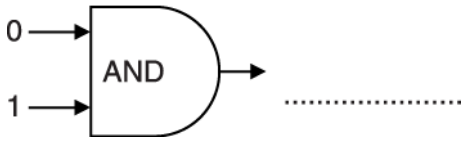
Describe what is meant by a logic error.

[2]

7(a) State the output of each of the following logic circuits for the inputs given.



.....



.....

[2]

(b) Fig. 1 is a circuit diagram.

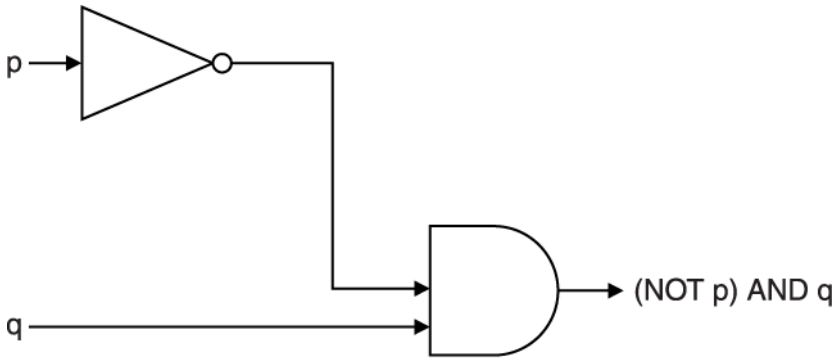


Fig. 1

Complete the truth table for Fig. 1.

p	q	(NOT p) AND q
0	0	0
1	0	0

[3]

8 Julie is writing a computer game that simulates a 100 m race. Each time the space bar is pressed, the position of the player moves up by 1. When the position reaches 100, the player has won.

Here is Julie's algorithm for the program

```
CONST PlayerKey = " "
Position = 0
REPEAT
  INPUT KeyPressed
  If KeyPressed = PlayerKey THEN
    Position = Position + 1
  END IF
UNTIL Position = 100
```

To make the game more interesting, Julie changes the rules. Each time the spacebar is pressed, the position of the player will now move up by a random number.

State **two** changes that need to be made to include this new rule. Justify each change.

Change 1 ---

Justification ---

Change 2 ---

Justification ---

---

[4]

9(a) Santos is writing a program that guesses the number of goals a team will score in a football match.

The algorithm for his program is shown below:

```
01  CONST Noise = 10
02  INPUT Wins
03  INPUT Losses
04  Goals = 0
05  Net = Wins - Losses
06  WHILE Net > Noise
07   Goals = Goals + 1
08   Net = Net - Noise
09  END WHILE
10  OUTPUT Goals
```

State what is meant by a constant and give an example from the algorithm above.

---

[2]

(b) State what is meant by a variable and give an example from the algorithm above.

---

[2]

(c) State the number of goals that will be output by this algorithm for the following inputs.
Explain how you obtained your answer in each case.

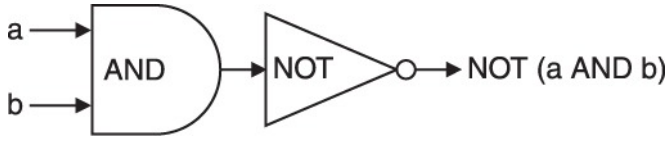
Wins = 30 Losses = 25

[2]

Wins = 20 Losses = 5

[3]

10 The following logic diagram shows the expression NOT (a AND b).

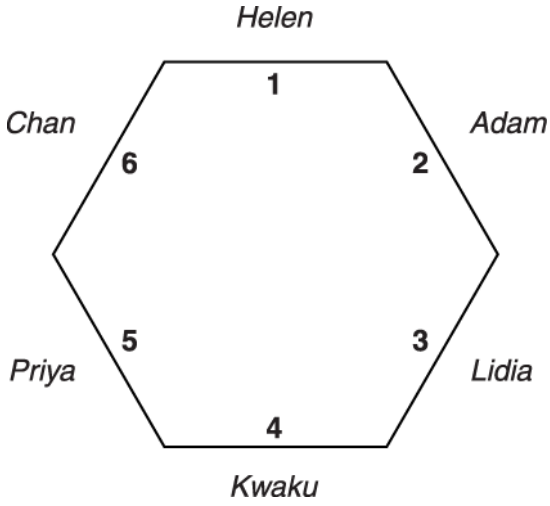


Complete the missing boxes in the truth table below to show the value of NOT (a AND b) that will be output for each possible set of values of a and b.

a	b	NOT (a AND b)
0	0	1
0		1
1	0	

[4]

11(a) A game on a computer shows six players around a table on seats. They are numbered 1 to 6 as shown below.



The names of the players are stored in an array with six elements called PlayerName. The index position of the array is used to indicate the seat number.

For example, the value of PlayerName(1) is “Helen”.

State the value of PlayerName(3).

---

[1]

(b) Describe what will happen if the code for the game includes an instruction to print the value of PlayerName(7).

---

[2]

(c) During the game, each player sometimes moves clockwise by a given number of places.

For example, if the number of places is 2, Helen will move to seat 3, Priya will move to seat 1 etc.

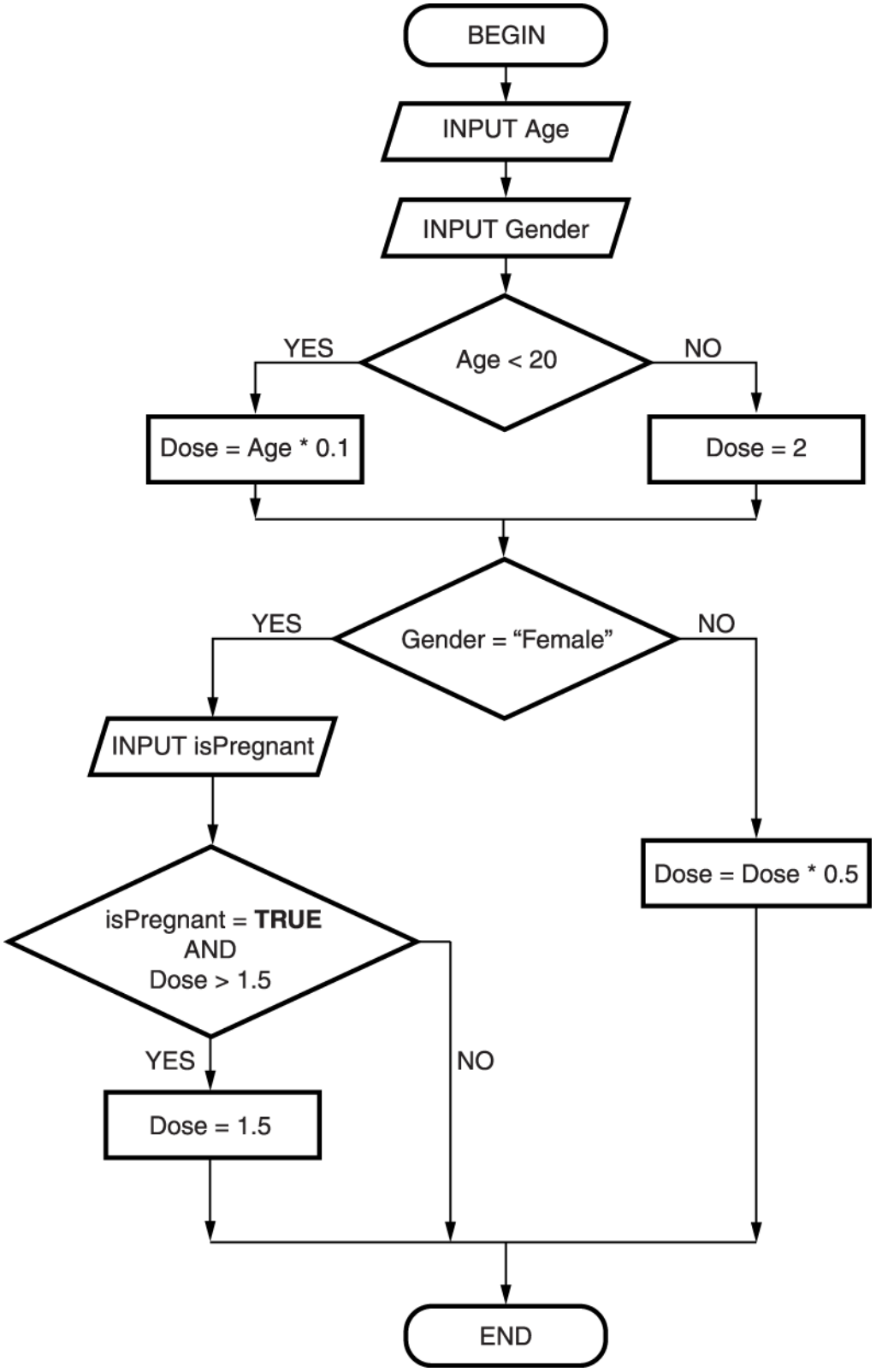
Write an algorithm that will update the contents of the array PlayerName after a move has occurred. Your algorithm should:

- allow the number of places to move to be input
- use iteration
- ensure that all of the existing players' names are moved to the correct position in the array.

[6]

12 A computer program calculates the correct dose in grams of a type of medicine.

The algorithm used is shown by the flow diagram below.



The data type of the variable Age is Integer.

State the data type of the following variables used in the flow diagram.

Variable	Data Type
Gender	
Dose	
isPregnant	

[3]

13 Joseph is an author and programmer, and he needs to estimate how many pages his new book will have.

Each page has an average of 300 words. Each chapter starts with a chapter title page.
The number of pages is estimated by;

- dividing the number of words by 300
- ignoring the decimal part of the division
- adding the number of chapters to this total.

Joseph uses the algorithm below to estimate the number of pages, but his algorithm does not give the correct result.

```
01 INPUT numberOfWords
02 INPUT numberOfChapters
03 CONST wordsPerPage = 300
04 numberOfPages = RoundDown(numberOfWords / wordsPerPage)
05 numberOfPages = numberOfWords + numberOfChapters
06 OUTPUT numberOfPages
```

Joseph has used a `RoundDown` function to remove the decimal part of the division, e.g. `RoundDown(6.2)` would return 6, `RoundDown(7.8)` would return 7.

There is an error in line `05` of the algorithm.

Write a corrected line of code to replace line `05`.

[1]

14(a) A programmer has written an algorithm to output a series of numbers. The algorithm is shown below:

```
01  for k = 1 to 3
02      for p = 1 to 5
03          print (k + p)
04      next p
05  next k
06  m = 7
07  print m * m
```

(i) Give the first **three** numbers that will be printed by this algorithm.

---

[1]

(ii) State how many times line **03** will be executed if the algorithm runs through once.

---

[1]

(b) Identify **two** basic programming constructs that have been used in this algorithm.

1 ---

---

2 ---

---

[2]

(c)

(i) Describe what is meant by a variable.

---[2]

(ii) Identify **two** variables that have been used in the algorithm above.

1 ---

2 ---

[2]

15 A library gives each book a code made from the first three letters of the book title in upper case, followed by the last two digits of the year the book was published.

For example, “Poetry from the War”, published in 2012 would be given the code `POE12`.

The library sorts their books based on the book code.

(i) Show the steps that a merge sort would take to put the following list of book codes into ascending alphabetical order (from A to Z).

```
POE12 , BAC97 , FLY77 , JAV16 , TAL86 , AND18 , ZAR09 , HOP86
```

---

[4]

(ii) Explain **one** advantage of a merge sort compared to a bubble sort.

---

[2]

16 Victoria is writing a program using a high level language to display the meaning of computer science acronyms that are entered. The code for her first attempt at this program is shown below.

```
01 a = input("Enter an acronym")
02 if a == "LAN" then
03     print("Local Area Network")
04 elseif a == "WAN" then
05     print("Wide Area Network")
06 ……
07 ……
08 endif
```

(i) Complete the code above to print out an "unknown" message if any other acronym is entered by the user. [2]

(ii) Describe what is meant by a "high level language".

---

[2]

17 Louise writes a program to work out if a number entered by the user is odd or even. Her first attempt at this program is shown.

```
01  num = input(“enter a number”)
02  if num MOD 2 >= 0 then
03      print(“even”)
04  else
05      pritn(“odd”)
06  endif
```

The program contains a syntax error on line **05**.

(i) State what is meant by a syntax error.

---

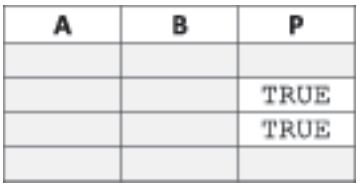
[1]

(ii) Give a corrected version of line **05** that fixes the syntax error.

---

[1]

END OF QUESTION PAPER

Question			Answer/Indicative content	Marks	Guidance
1			• Taking a number as input • Using HEX subroutine correctly • Calculating Digit 1 • Calculating Digit 2 INPUT decimal digitl = decimal DIV 16 IF digitl>=10 THEN digit1 = HEX(digit1) digit2 = decimal - (digit1*16) IF digit2>=10 THEN digit2=HEX(digit2)	4	1 mark for each bullet. There are no marks associated with data types or conversions of data types. If used, a flowchart should represent the bulleted steps in the answer column.
			Total	4	
2				2	1 mark for each correct answer in table.
			Total	2	
3	a		RebEl	1	Correct Answer Only (allow any case)
	b	i	• UitFr	1	Correct Answer Only (allow any case)

		ii	• Taking firstname, surname and gender as input
			• Checking IF gender is male / female (using appropriate selection)
			• For male …Generating last 3 letters of surname using appropriate string manipulation
			• …Generating first 2 of letters of firstname and adding to previous
			• For female…. correctly calculating as before
			• Correct concatenation **and** output
			
			input firstname, surname, gender
			if gender = “Male” then
			  username = RIGHT(surname, 3) + LEFT(firstname,2)
			else
			  username = LEFT (firstname,3) + LEFT(surname,2)
			end if
			print (username)
				6	1 mark for each correct bullet to a maximum of 6.
					
					If used, a flowchart should represent the bulleted steps in the answer column

			Total	8	

Question			Answer/Indicative content	Marks	Guidance
4	a	i	print (hoursPlayed[0,2])	1	Correct Answer Only
		ii		1	Correct Answer Only
		iii	80	1	Correct Answer Only
		iv	• Adding all correct elements • Outputting correctly • Using a loop e.g. total = 0 for x = 0 to 4 total = total + hoursPlayed[0,x] next x print (total)	3	1 mark per bullet to a maximum of 3. If used, a flowchart should represent the bulleted steps in the answer column

	b		• Loop 0 to 29
			• Loop 0 to 4
			• Accessing hoursplayed[x,y]
			• Addition of hoursplayed[x,y] to total
			• Calculating average correctly outside of loops
			• Outputting the results
			
			e.g.
			total = 0
			for x = 0 to 29
			  for y = 0 to 4
			  Total = total + hoursPlayed[x,y]
			  next y
			nextx
			average = total / (30*5)
			print (average)
				6	Accept any type of average calculation (mean, median, mode).
					
					If used, a flowchart should represent the bulleted steps in the answer column.

			Total	14	

<table>
<tr><th colspan="3">Question</th><th>Answer/Indicative content</th><th>Marks</th><th>Guidance</th></tr>
<tr><td>5</td><td>a</td><td></td><td>
<table>
<tr><td>crime</td><td>bait</td><td>fright</td><td>victory</td><td>nymph</td><td>loose</td></tr>
<tr><td>bait</td><td>crime</td><td>fright</td><td>victory</td><td>nymph</td><td>loose</td></tr>
<tr><td>bait</td><td>crime</td><td>fright</td><td>nymph</td><td>victory</td><td>loose</td></tr>
<tr><td>bait</td><td>crime</td><td>fright</td><td>nymph</td><td>loose</td><td>victory</td></tr>
<tr><td>bait</td><td>crime</td><td>fright</td><td>loose</td><td>nymph</td><td>victory</td></tr>
</table>
</td><td>4</td><td>1 mark for each row from row 2-5. Allow multiple swaps in one stage, where it is clear that a bubble sort has been applied.</td></tr>
<tr><td></td><td>b</td><td></td><td>• Comparing zebra to orange
• Greater, so split and take right side
• Further comparison (1 or 2 depending on choices made)
• Correct identification of zebra using methodology above

e.g.
compare zebra to orange

greater, split right

compare to wind

greater, split right

compare to zebra</td><td>4</td><td>1 mark per bullet (multiple ways through, marks awarded for appropriate comparison and creation of sub groups).</td></tr>
<tr><td></td><td></td><td></td><td>Total</td><td>8</td><td></td></tr>
<tr><td>6</td><td></td><td></td><td>• The error does not prevent program running…
• But it does not produce the expected output / it does not do what the programmer intended
• A reasonable example</td><td>2</td><td><u>Examiner's Comments</u>

Candidates who had learnt a definition for logic error were able to answer this more clearly and succinctly than candidates who were attempting to put it in their own words, often confusing the use of the term “logic” here with the everyday use of the term and giving answers such as “it doesn’t make sense”.</td></tr>
<tr><td></td><td></td><td></td><td>Total</td><td>2</td><td></td></tr>
</table>

<table>
<tr><th colspan="3">Question</th><th>Answer/Indicative content</th><th>Marks</th><th>Guidance</th></tr>
<tr><td>7</td><td>a</td><td></td><td>• 1
• 0.

(respectively)</td><td>2</td><td><u>Examiner's Comments</u>

Most candidates answered correctly. As expected, some weaker candidates were less able to work with the logic gates in combination</td></tr>
<tr><td></td><td>b</td><td></td><td>Correct answer:

<table><tr><td>p</td><td>q</td><td>(NOT p) AND q</td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>
Award marks for

• Correct missing input cases (0 1, 1 1 or 1 1, 0 1)
• Output of 1 for 0 1
• Output of 0 for 1 1.</td><td>3</td><td><u>Examiner's Comments</u>

As expected, some weaker candidates were less able to work with the logic gates in combination</td></tr>
<tr><td></td><td></td><td></td><td>Total</td><td>5</td><td></td></tr>
</table>

Question			Answer/Indicative content	Marks	Guidance
8			e.g. • Position = Position + 1 should be changed • …so the increment is a random number • The random number should be relatively small • …so the game remains interesting • The end condition of the loop should be changed to UNTIL Position > 100 / check if position > 100 and if so change to 100 • … as the position may not reach exactly 100 due to the random number. • seed / initialise random number generation • … so that numbers generated appear random Mark in pairs	4	Accept other suitable change and its justification ?Examiner's Comments?? This question worked well as a differentiator, testing problem solving and computational thinking skills. While the weakest candidates were not able to answer the question, most candidates were able to explain how to implement the change that was explicitly required in the question (a random increment at every step). Only the strongest candidates identified the consequences of this change on other parts of the algorithm and explained what needed to be done to address them. Many candidates expressed one change as two (for example, considering generating a random number and replacing the increment by 1 with an increment by this random number as two separate changes to the code).
			Total	4	

Question			Answer/Indicative content	Marks	Guidance
9	a		• A value that does not change (while the program is running) • eg Noise	2	For the example do not accept the whole line of code; candidate should show that they know where the constant is. Note that “A constant is a variable which does not change” is a contradictory answer (because by definition variables change) and when candidates give a contradictory answer award no marks. <u>Examiner's Comments</u> Given the good ability shown by candidates to follow the algorithm in (c) and the fact that prior to taking this examination, candidates would have completed the controlled assessment tasks in A453, one would have expected stronger answers for the definitions of constants and variables in (a) and (b) than those seen. Typically vague answers such as “something that does not change” and “something that can change” do not demonstrate to the examiner an understanding of the meaning of these terms in the context of programming as they more closely describe their everyday meaning, and were not awarded any marks. Some candidates stated that “a constant is a variable which does not change” which was considered a self-contradictory answer. Another common mistake was to state that constants and variables were numbers. Also, candidates needed to be more precise when identifying constants and variables in the pseudocode provided by stating the name only. By quoting the whole line in which a constant appears, such as “CONST noise = 10), candidates indicate to the examiner that they either do not know what a constant is, or they do not know precisely where it is in that line of code. In (c) many candidates followed the algorithm correctly and were awarded full marks. Weaker candidates demonstrated a misunderstanding of the abstractions used and seemed distracted by alternative possible meanings of the identifiers in the

<table>
<tr><th colspan="3">Question</th><th>Answer/Indicative content</th><th>Marks</th><th>Guidance</th></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td>question, for example by assuming that “Net” means the ball has touched the net and equating it to the number of goals.</td></tr>
<tr><td></td><td>b</td><td></td><td>• A location in memory to store / a value that may change (as the program is running)
• eg Wins / Losses/ Net / Goals</td><td>2</td><td></td></tr>
<tr><td></td><td>c</td><td></td><td>• Net = 5 which is less than Noise
• Goals = 0

• Net = 15 which is greater than Noise
• Runs Loop once {Goals = Goals + 1, Net = Net ? Noise}…
• Goals = 1</td><td>2

3</td><td>1 mark for the subtraction and result of the comparison
1 mark for correct result

1 mark for the subtraction and result of the comparison
1 mark for clearly indicating that the loop is executed once
1 mark for correct result

Remember to enter a total mark out of 5 for both sections.</td></tr>
<tr><td></td><td></td><td></td><td>Total</td><td>6</td><td></td></tr>
<tr><td>10</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.

<u>Examiner's Comments</u>

This part was well answered by the majority of candidates, indicating that logic and truth tables – a core concept in computer science – is understood by most candidates.</td></tr>
<tr><td></td><td></td><td></td><td>Total</td><td>4</td><td></td></tr>
</table>

<table>
<tr><th colspan="3">Question</th><th>Answer/Indicative content</th><th>Marks</th><th>Guidance</th></tr>
<tr><td>11</td><td>a</td><td></td><td>• Lidia</td><td>1</td><td>Accept incorrect spelling if intention is clear.</td></tr>
<tr><td></td><td>b</td><td></td><td>• Program finds there is no position 7 in the array / array index out of bounds
• An error will occur / an error message would be displayed / program will crash</td><td>2</td><td>Only award bullet 1 if answer is clearly about the contents of the array and not about the context.

Do not award bullet 2 if candidate specifically mentions syntax error.</td></tr>
<tr><td></td><td>c</td><td></td><td>Example
<pre>INPUT Num

For i = 1 to Num
  Temp = PlayerName(6)
  PlayerName(6) = PlayerName(5)
  PlayerName(5) = PlayerName(4)
  PlayerName(4) = PlayerName(3)
  PlayerName(3) = PlayerName(2)
  PlayerName(2) = PlayerName(1)
  PlayerName(1) = Temp
Next i</pre>
Award marks for:

• Input the number of places to move (e.g. Num)
• Use of temporary variable(s) or second array to avoid overwriting values in the array
• Sensible use of a loop
• … with correct end condition
• Correctly deals with moving from position 1 (e.g. 1 + Num)
• Correctly deals with moving from position 6 (e.g. Num)</td><td>6</td><td>If there is more than one loop, award bullets 3 and 4 for any non-trivial loop that contributes to the solution.

For bullet 3, “sensible” use of a loop, requires that the loop clearly address the problem (e.g. move every player from pos a to b). Although candidates can get partial marks here, candidates will only get full marks (incl bullet 6) if all conditions of all loops are correct.</td></tr>
<tr><td></td><td></td><td></td><td>Total</td><td>9</td><td></td></tr>
<tr><td>12</td><td></td><td></td><td><table><tr><th>Variable</th><th>Data Type</th></tr><tr><td>Gender</td><td>String</td></tr><tr><td>Dose</td><td>Real</td></tr><tr><td>isPregnant</td><td>Boolean</td></tr></table>
1 mark per row</td><td>3</td><td>Allow known equivalent names of data types:

String: alphanumeric / text. Do not accept character but accept an array of character or pointer to character. Real: single, double, float, decimal. Do not accept Number. Boolean: Yes / No, True / False</td></tr>
<tr><td></td><td></td><td></td><td>Total</td><td>3</td><td></td></tr>
</table>

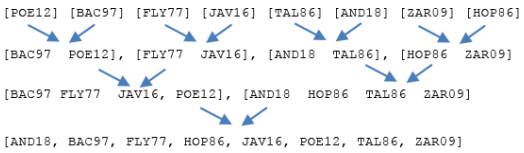
Question			Answer/Indicative content	Marks	Guidance
13			numberOfPages = numberOfPages+numberOfChapters	1	Accept: • += instead of = numberOfPages • numberOfPages=RoundDown(numberOfWords / wordsPerPage) +numberOfChapters • numberOfPages=RoundDown(numberOfWords / 300) +numberOfChapters Variable names must be spelt correctly, ignore case Examiner's Comments This question was appropriate programming theory and techniques. There was a mix of responses to this question, many candidates were able to get this correct, whilst others were unable to follow the code.
			Total	1	

Question			Answer/Indicative content	Marks	Guidance
14	a	i	• 2,3,4	1	All three numbers needed in the correct order (with no other numbers) for mark. Examiner's Comments The majority of candidates seemed to understand the use of count-controlled loops to repeat sections of an algorithm, which was pleasing. However, the vast majority were then unable to apply this to nested loops, understanding that the inner loop completes fully for each iteration of the outer loop. The correct answer for a(i) of 2, 3, 4 (with k remaining as 1 but p being 1 then 2 then 3) was only seen in candidates who generally achieved highly overall on the paper. Question part a(ii) tests the same understanding and so approximately the same number of candidates understood that the given lines would repeat 15 times in total.
		ii	• 15	1	Accept 3 × 5 Examiner's Comments The majority of candidates seemed to understand the use of count-controlled loops to repeat sections of an algorithm, which was pleasing. However, the vast majority were then unable to apply this to nested loops, understanding that the inner loop completes fully for each iteration of the outer loop. The correct answer for a(i) of 2, 3, 4 (with k remaining as 1 but p being 1 then 2 then 3) was only seen in candidates who generally achieved highly overall on the paper. Question part a(ii) tests the same understanding and so approximately the same number of candidates understood that the given lines would repeat 15 times in total.

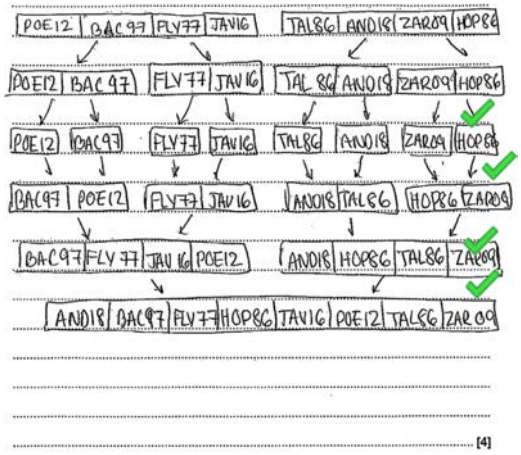
Question			Answer/Indicative content	Marks	Guidance
	b		1 mark per bullet, max 2 • Sequence • Iteration / loops / repetition	2	Ignore spelling. Do not allow examples (eg FOR loop / WHILE loop) <u>Examiner's Comments</u> Programming constructs are defined in section 2.2 ('Programming techniques') of the specification. These are given as sequence, selection and iteration. In the algorithm given in the question, sequence and iteration were both used and so candidates were credited if they gave either of these two answers, with loops being accepted as an alternative to iteration. Centres are encouraged to ensure that all of the specification is covered with candidates and that keywords from the specification are highlighted in lessons as appropriate.

Question			Answer/Indicative content	Marks	Guidance
	c	i	1 mark per bullet, max 2 • A (name/identifier for a) **memory location** • used to (temporarily) **holds/contains/stores** data / value // is assigned a value • that **can be** changed / **possible** to change (while the program is running)	2	Do not accept “will change” for bullet point 4. Do not allow “holds/stores something” or “holds/stores information” for bullet point 2. Do not accept name / identifier without reference to a memory location. Do not accept “a value given a name” or equivalent. Examiner’s Comments This question was pleasingly answered relatively well, perhaps because of similar questions regarding variables and constants in question on the previous GCSE Computing specification. It was pleasing how many candidates were able to relate variables to an identifier given to a memory location, showing a good understanding of a basic Computer Science concept. A mark was given quite generously for the ability for a variable to be ‘changed’, but candidates should be aware that ideally their answer should make reference to the value stored being able to be changed, not the variable itself changing somehow.

Question			Answer/Indicative content	Marks	Guidance
		ii	1 mark per bullet, max 2 • k • p • m	2	Ignore capitalisation. Correct answer only. Do not allow other code in answer. Examiner's Comments There were actually three variables in use (k, p and m) and so any two of these were able to be given to achieve full marks. This question was answered very well by the majority of candidates, with most achieving 2 marks. A common incorrect answer was to give full lines of pseudocode as the variable, e.g. 'm = 7' or 'next k'. In this case it was impossible to decide whether the candidate understood which part of their response was the variable and so no mark was given. As with question 1b(ii), candidates should be encouraged to look at the command word in the question (in this case 'identify') and make sure that their response is suitable.
			Total	8	

Question			Answer/Indicative content	Marks	Guidance
15		i	1 mark per bullet, max 4. • List split into individual elements (may be done over several steps or just as a starting point) • **Merge** individual elements into **sorted** lists of size 2 • **Merge** lists of size 2 into **sorted** lists of size 4 • **Merge** lists of size 4 into final sorted list.	4	Candidates **can** describe how the merge sort would work rather than showing output values at each stage. Ignore intermediate steps. **Do not** give final mark for simply showing the list sorted. Must have the (correct) idea of where it being merged from previous lists. Candidates' answers describing / showing other sorting algorithms (e.g. bubble sort, insertion sort) are worth 0 marks. 

Question			Answer/Indicative content	Marks	Guidance
		ii	1 mark per bullet, max 2. • Faster/quicker (to sort)… • …for large lists // for lists that are **more unordered** • Has a consistent running time (for a lists of same length)… • …doesn't depend on how ordered original list is	2	Accept (correct) reference to big O notation for 2nd mark on either mark point although this is beyond scope of GCSE specification. Allow "more efficient" for BOD on first bullet point. <u>Examiner's Comments</u> Large numbers of candidates understood the divide and conquer strategy applied by the merge sort algorithm; the list is continually divided in half until lists of size 1 are achieved before the lists are then merged together to achieve the sort, with the size of lists doubling with every iteration (i.e. when two lists of size 2 are merged, the result is a list of size 4). Many candidates achieved good marks on this question. Where mistakes were made, they generally fell into one of three categories. One was applying a different algorithm than had been asked (e.g. showing a bubble sort). Another was misunderstanding where the sorting takes place; some candidates showed the lists being split up correctly, but then these being merged and the list sorted in place afterwards – this is incorrect, it is the act of merging that sorts the values. A third and more common issue was that examiners found it extremely tough to decide where lists were split up or merged from candidate responses that seemed to list all values in one row with no seeming differentiation between one list of eight values and eight lists of one value. Where examiners were not able to see this, marks could not be given.

Question			Answer/Indicative content	Marks	Guidance
					Misconception When using merge sort to merge together [1, 3] and [2, 4] into ascending order, the algorithm will take the lowest value from the front of either list (in this case 1) and place it into the new, merged list. This will then repeat meaning that the new list will be [1, 2, 3, 4]. It does not merge the lists to be [1, 3, 2, 4] and then sort this list.  Exemplar 5 The exemplar above shows an ideal way to respond to this question. The list is obviously and clearly split up in successive passes into lists of a single value. Each pair of lists is then merged, with the process of merging resulting in sorted lists. Each time the lists are merged, the result is a larger list in ascending order. The candidate response shown here not only shows this process but makes it very clear that the values are held in separate lists and even includes arrows to illustrate the process of merging. This achieved full marks [4 out of 4].
			Total	6	

Question			Answer/Indicative content	Marks	Guidance
16			1 mark per bullet, max 2. • `else` • `print (“unknown”)`	2	Accept logically correct equivalents for else (e.g. `elseif a!=“LAN“ and/or a !=“WAN“`). Do not allow `elseif` on its own Accept other keywords for print (e.g. “output”) as long as the intention is clear. Accept other messages as equivalent to “uknown” (e.g. “not known” / “error”)) Message to be printed must be in quotes. Allow `“else then”`. Examiner’s Comments Most candidates were able to complete line 07 successfully, with an output/print of an acceptable ‘unknown’’ message being all that was required. Line 06 required candidates to understand that the message on line 07 should be printed out where the conditions in lines 02 and 04 were both false; the simplest way of achieving this was an ELSE (or equivalent). Where candidates put more logically complex statements, these were successful if they were logically correct. However, many candidates struggled with this.  Misconception To compare the contents of a variable against two possible values, it is incorrect to use : • `if a!=‘WAN’ or ‘LAN’` In the example above, the comparison of a against ‘WAN’ is logically correct, but it is unclear what ‘LAN’ is being compared against. A logically correct way to achieve the same thing would be : • `if a!=‘WAN’ and a!=‘LAN’`

Question			Answer/Indicative content	Marks	Guidance
			1 mark per bullet, max 2. • aimed at humans//understandable by humans / programmers • English like structure / syntax • Must be translated/compiled/interpreted (before it can be run) • Allows programmer to deal with the problem instead of considering the underlying hardware // an abstraction from the hardware // hardware independent // portable	2	Allow examples of keywords (eg IF / ELSE / WHILE) as 2nd bullet point. Do not award marks for naming languages such as Java , Python, etc. Do not award marks for stating what a high level language isn't (i.e. describing what low level code is). Do not allow "easy to use" Do not allow 'has to be converted' without into what i.e machine code etc. Examiner's Comments This question was answered well by many candidates, with responses relating to the use of English-like keywords but needing to be translated before the processor could execute it being the most popular. A minority of candidates confused 'high-level' with 'difficult' and gave incorrect answers regarding it being too hard to programmers to use. The opposite is in fact true, with 'high-level' referring to the level of abstraction away from the underlying hardware. Very few answers were seen that in any way discussed this abstraction or portability between different processors. Hopefully centres will become more confident with the delivery of the subject and so more complex and technically complete answers will be seen across all questions from high ability candidates.
			Total	4	

Question			Answer/Indicative content	Marks	Guidance
17		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). <u>Examiner's Comments</u> This question asked candidates to state the meaning of the term 'syntax error'. To be successful, candidates needed to show examiners that they knew how a syntax error differed from other types of errors and so answers such as 'when code does not work' were not precise enough. A very common incorrect answer here was to explain what caused the syntax error in this code or to give general examples of things that would cause syntax errors (e.g. 'a misspelling of a word') and not to give the wider answer of an error that breaks the grammatical rules of the programming language. **Exemplar 2** (i) State what is meant by a syntax error. when the programme is broken because ~~there~~ coding is incorrect. [1] In Exemplar 2, for example, the response given by the candidate above could equally apply to logic errors and is therefore assessed to be too vague. This was given 0 marks. **AfL** Candidates should make sure that they read questions carefully to ascertain whether a contextual basis is required in their answer. 'What is meant by a syntax error' is a very different question than 'what caused the syntax error in this code'. Examiners mark against a given mark

Question			Answer/Indicative content	Marks	Guidance
					scheme and even when candidates show understanding, this can only be credited if the question on the paper has been answered.
		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 Examiner's Comments This question was answered extremely well as the majority of candidates were able to identify and fix the misspelling of the keyword 'print' in their answer. The small number of candidates who did not manage to successfully answer this question tended to try to expand their answer beyond simply fixing the error. They then introduced new errors or unnecessary code. Where additional code was given, if it logically worked then this was accepted.
			Total	2	

Topic 2

Programming Fundamentals

1 There is a subroutine, HEX(), that takes a denary number between 10 and 15 and returns the corresponding hexadecimal number. E.g. HEX(10) would return “A”, HEX(15) would return “F”.

Write an algorithm, using the subroutine HEX(), to convert any whole decimal number between 0 and 255 into a 2 digit hexadecimal number.

[4]

2 The area of a circle is calculated using the formula $\Pi \times r^2$, where Π is equal to 3.142 and r is the radius.

Finn has written a program to allow a user to enter the radius of a circle as a whole number, between 1 and 30, and output the area of the circle.

```
01    int radius = 0
02    real area = 0.0
03    input radius
04    if radius < 1 OR radius > 30 then
05    print ('Sorry, that radius is invalid')
06    else
07    area = 3.142 * (radius ^ 2)
08    print (area)
09    end if
```

Identify **two** variables used in the program.

--- [2]

3 Jim is writing a program to calculate the wages of workers in a teddy bear factory.

Workers sometimes get a £50 bonus.

Here is the algorithm used to calculate whether a worker should get a bonus.

```
Limit = 200
INPUT WagesEarned
IF WagesEarned < Limit THEN
    Pay = WagesEarned
ELSE
    Pay = WagesEarned + 50
END IF
```

State the value of Pay after this code is executed for each of the following values of WagesEarned.

WagesEarned = 50 Pay = _____

WagesEarned = 200 Pay = _____

[2]

4 Julie is writing a computer game that simulates a 100 m race. Each time the space bar is pressed, the position of the player moves up by 1. When the position reaches 100, the player has won.

Here is Julie's algorithm for the program

```
CONST PlayerKey = " "
Position = 0
REPEAT
  INPUT KeyPressed
  If KeyPressed = PlayerKey THEN
    Position = Position + 1
  END IF
UNTIL Position = 100
```

State what is meant by selection and iteration using examples from Julie's algorithm.

Selection _____

_____

Example _____

_____

Iteration _____

_____

Example _____

_____

[4]

5(a) Santos is writing a program that guesses the number of goals a team will score in a football match.

The algorithm for his program is shown below:

```
01  CONST Noise = 10
02  INPUT Wins
03  INPUT Losses
04  Goals = 0
05  Net = Wins - Losses
06  WHILE Net > Noise
07   Goals = Goals + 1
08   Net = Net - Noise
09  END WHILE
10  OUTPUT Goals
```

State what is meant by a constant and give an example from the algorithm above.

---

[2]

(b) State what is meant by a variable and give an example from the algorithm above.

---

[2]

(c) State the number of goals that will be output by this algorithm for the following inputs.
Explain how you obtained your answer in each case.

Wins = 30 Losses = 25

[2]

Wins = 20 Losses = 5

[3]

6 Joseph is an author and programmer, and he needs to estimate how many pages his new book will have.

Each page has an average of 300 words. Each chapter starts with a chapter title page.
The number of pages is estimated by;

- dividing the number of words by 300
- ignoring the decimal part of the division
- adding the number of chapters to this total.

Joseph uses the algorithm below to estimate the number of pages, but his algorithm does not give the correct result.

```
01 INPUT numberOfWords
02 INPUT numberOfChapters
03 CONST wordsPerPage = 300
04 numberOfPages = RoundDown(numberOfWords / wordsPerPage)
05 numberOfPages = numberOfWords + numberOfChapters
06 OUTPUT numberOfPages
```

Joseph has used a `RoundDown` function to remove the decimal part of the division, e.g. `RoundDown(6.2)` would return 6, `RoundDown(7.8)` would return 7.

State whether this algorithm uses selection, sequence or iteration.

[1]

7(a) OCR High School uses a computer system to store data about students' conduct. The system records good conduct as a positive number and poor conduct as a negative number. A TRUE or FALSE value is also used to record whether or not a letter has been sent home about each incident.

An example of the data held in this system is shown below in Fig. 1:

StudentName	Detail	Points	LetterSent
Kirstie	Homework forgotten	−2	FALSE
Byron	Good effort in class	1	TRUE
Grahame	100% in a test	2	FALSE
Marian	Bullying	−3	TRUE

Fig. 1

State the most appropriate data type used to store each of the following items of data.

- StudentName ----------
- Points ----------
- LetterSent ---------- [3]

(b) The data shown above in Fig. 1 is stored in a database table called **Conduct**.

(i) Write an SQL statement to select the StudentName field for all records that have negative Points.

---------- [3]

(ii) State the wildcard that can be used in SQL to show all fields from a table.

---------- [1]

8 Louise writes a program to work out if a number entered by the user is odd or even. Her first attempt at this program is shown.

```
01  num = input(“enter a number”)
02  if num MOD 2 >= 0 then
03      print(“even”)
04  else
05      pritn(“odd”)
06  endif
```

The program contains a logic error on line **02**.

(i) State what is meant by a logic error.

---

[1]

(ii) Give a corrected version of line **02** that fixes the logic error.

---

[1]

9 Elliott plays football for OCR FC. He wants to create a program to store the results of each football match they play and the names of the goal scorers. Elliott wants individual players from the team to be able to submit this information.

The number of goals scored in each football match is held in an array called `goals`. An example of this array is shown.

```
goals = [0, 1, 3, 0, 4, 5, 2, 0, 2, 1]
```

Elliott wants to count how many matches end with 0 goals.

Complete the following pseudocode for an algorithm to count up how many matches with 0 goals are stored in the array and then print out this value.

```
01  nogoalscount = 0
02  for count = 0 to (goals.length-1)
03      if goals[……] == 0 then
04        nogoalscount ……
05      endif
06  next count
07  print(……)
```

[3]

10 The symbol ^ is used for exponentiation.
Give the result of a^b when a = 3 and b = 2.

[1]

11 Complete the truth table in **Fig. 1** for the Boolean statement **`P = NOT(A AND B).`**

A	B	P
0	0	1
0	1	.....
1	0	.....
1	1	0

Fig. 1

[2]

12 The area of a circle is calculated using the formula $\pi \times r^2$ where π is equal to 3.142 and r is the radius.

A program is written to allow a user to enter the radius of a circle as a whole number between 1 and 30, then calculate and output the area of the circle.

```
01 radius = 0
02 area = 0.0
03 radius = input("Enter radius")
04 if radius < 1 OR radius > 30 then
05 print("Sorry, that radius is invalid")
06 else
07 area = 3.142 * (radius ^ 2)
08 print (area)
09 endif
```

Identify **two** variables used in the program.

1 ---

2 ---

[2]

13 A teacher researches the length of time students spend playing computer games each day.

Tick (✓) **one** box to identify the data type you would choose to store the data and explain why this is a suitable data type.

Data Type	Tick (✓) one box
String	
Integer	
Real	
Boolean	

Explanation: _____

_____

[2]

14 A cinema uses the following criteria to decide if a customer is allowed to see a film that has a 15 rating:

Customers have to be 15 years of age or older to see the film. They also need to either have a ticket or have the money to buy a ticket.

The table shows the inputs to the system that will output whether the customer can watch the film.

Input	Criteria (True / False)
A	The customer is 15 or over
B	The customer has a ticket
C	The customer has the money to buy a ticket

The cinema has three screens: "Red", "Black" and "Yellow".

The function `freeseats()` counts how many seats are available in each screen. The name of the screen is passed in as a string parameter and the number of free seats is returned as an integer.

Write code using the function `freeseats()` to find the number of seats available in screen Red and assign this to a variable with identifier `redseats`.

---

[2]

---

15 `DIV` and `MOD` are both operators used in computing-related mathematics.

(i) State the value of `13 DIV 4`

---

[1]

(ii) State the value of `13 MOD 4`

---

[1]

END OF QUESTION PAPER

Question			Answer/Indicative content	Marks	Guidance
1			• Taking a number as input • Using HEX subroutine correctly • Calculating Digit 1 • Calculating Digit 2 INPUT decimal digitl = decimal DIV 16 IF digitl>=10 THEN digit1 = HEX(digit1) digit2 = decimal - (digit1*16) IF digit2>=10 THEN digit2=HEX(digit2)	4	1 mark for each bullet. There are no marks associated with data types or conversions of data types. If used, a flowchart should represent the bulleted steps in the answer column.
			Total	4	
2			• radius • area	2	
			Total	2	
3			• 50 • 250.	2	Examiner's Comments This was generally well answered although a few candidates lost both marks by confusing the less than and greater than symbols.
			Total	2	

Question			Answer/Indicative content	Marks	Guidance
4			Selection • A condition is used to decide whether code should be executed • Position = Position + 1 is only run if the IF condition is met. Iteration • code is executed repeatedly • The code in the repeat loop will be run several times (until Position = 100).	4	?Examiner's Comments?? Candidates who have completed the course will have used iteration and selection extensively as part of their programming, and so it was disappointing that many appeared not to recognise these terms, as they apply to programming, for this question, and made guesses based on the normal everyday meaning of the terms (and hence often getting better marks for iteration). Centres are advised not to teach the theoretical concepts relating to programming (which are assessed in this examination) separately from the programming practice which is assessed in A453, but rather to use their programming lessons as a context to bring this content alive. Again, it would be appropriate here for candidates to learn basic definitions by rote – their understanding of such definitions was tested separately by their being asked to apply them to the algorithm given.
			Total	4	

Question			Answer/Indicative content	Marks	Guidance
5	a		• A value that does not change (while the program is running) • eg Noise	2	For the example do not accept the whole line of code; candidate should show that they know where the constant is. Note that “A constant is a variable which does not change” is a contradictory answer (because by definition variables change) and when candidates give a contradictory answer award no marks. <u>Examiner's Comments</u> Given the good ability shown by candidates to follow the algorithm in (c) and the fact that prior to taking this examination, candidates would have completed the controlled assessment tasks in A453, one would have expected stronger answers for the definitions of constants and variables in (a) and (b) than those seen. Typically vague answers such as “something that does not change” and “something that can change” do not demonstrate to the examiner an understanding of the meaning of these terms in the context of programming as they more closely describe their everyday meaning, and were not awarded any marks. Some candidates stated that “a constant is a variable which does not change” which was considered a self-contradictory answer. Another common mistake was to state that constants and variables were numbers. Also, candidates needed to be more precise when identifying constants and variables in the pseudocode provided by stating the name only. By quoting the whole line in which a constant appears, such as “CONST noise = 10), candidates indicate to the examiner that they either do not know what a constant is, or they do not know precisely where it is in that line of code. In (c) many candidates followed the algorithm correctly and were awarded full marks. Weaker candidates demonstrated a misunderstanding of the abstractions used and seemed distracted by alternative possible meanings of the identifiers in the

Question			Answer/Indicative content	Marks	Guidance
					question, for example by assuming that "Net" means the ball has touched the net and equating it to the number of goals.
	b		• A location in memory to store / a value that may change (as the program is running) • eg Wins / Losses/ Net / Goals	2	
	c		• Net = 5 which is less than Noise • Goals = 0 • Net = 15 which is greater than Noise • Runs Loop once {Goals = Goals + 1, Net = Net ? Noise}… • Goals = 1	2 3	1 mark for the subtraction **and** result of the comparison 1 mark for correct result 1 mark for the subtraction and result of the comparison 1 mark for clearly indicating that the loop is executed once 1 mark for correct result Remember to enter a total mark out of 5 for both sections.
			Total	**6**	
6			Sequence	1	**Examiner's Comments** This question was appropriate programming theory and techniques. This question was answered well, with many candidates correctly getting sequence. Some candidates did not read the question, and gave a response other than the three options the question gave.
			Total	**1**	

Question			Answer/Indicative content	Marks	Guidance
7	a		1 mark per bullet, to max 3 • String • Integer / Int • Boolean	3	Accept text / varchar for string. Do not accept character. Do not accept number/numeric for integer Accept yes/no, true/false for Boolean. <u>Examiner's Comments</u> This question was answered very well by the majority of candidates, with most achieving full marks. Where candidates did fail to do well, it tended to be through a misunderstanding of the term 'data type', leading the answers such as 'Kirstie' or -3.
	b	i	1 mark per bullet, max 2 if not in correct order or additional statements given. • SELECT StudentName • FROM conduct • WHERE Points < 0	3	Capitalisation does not affect the mark. Spellings of fields, tables must be correct. Ignore brackets. Ignore quotes around StudentName, Conduct or Points. Mark quotes around 0 in WHERE clause as incorrect. StudentName must not include space Accept <= –1 or equivalent for 3rd bullet point. <u>Examiner's Comments</u> Despite 'the use of SQL to search for data' being listed under section 2.2 of the specification, the vast majority of candidate answers showed a lack of understanding of SQL in general. It was common to see incorrect answers that attempted to use pseudocode to respond to this and a large number leaving this question blank. Where candidates did use the keywords of SELECT, FROM and WHERE answers were much more successful. The SQL keywords and terms that candidates are required to know are listed in appendix 5f of the specification.

Question			Answer/Indicative content	Marks	Guidance
		ii	• */ star / asterisk	1	Wildcard (*) must be clearly identified as the answer. Do not allow any other SQL statements alongside this unless this is given as an example. Examiner's Comments As with the previous question, where candidates were familiar with SQL and understood the term 'wildcard', this question was answered successfully. However, the majority of candidates did not appear to be confident with either of these and so gave answers more suited to algorithm questions. Candidates should also be encouraged to provide answers which match the keyword in the question; in this case 'state'' means to give a specific name, value or other brief answer without explanation. The wildcard in question (an asterisk / *) was sometimes surrounded by other SQL code and so it was impossible to tell whether the candidate understood which part of this was the wildcard. A small number of candidates incorrectly identified '%' as the wildcard; although this would be a valid wildcard in the WHERE clause, it is not used to select all fields from a table. SQL coverage Appendix 5f of the specification lists all of the SQL keywords and wildcards that candidates are required to understand. This is available from http://www.ocr.org.uk
			Total	7	

Question			Answer/Indicative content	Marks	Guidance
8		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. Examiner's Comments This question asked candidates to state the meaning of the term 'logic error'. To be successful, candidates needed to show examiners that they knew how a logic error differed from other types of errors and so answers such as 'when code does not work' were not precise enough. Successful answers conveyed the idea of incorrect output without stopping the execution of the program.

		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`)
					
					Examiner's Comments
					
					The logic error present in the program was that any number entered would print out 'even' because line 02 checked if `num MOD 2` was greater than or equal to 0 rather than simply equal to 0. This could be fixed by replacing the greater than comparison with an = (or ==).
					
					Few candidates achieved this, suggesting a lack of confidence either with the MOD operator or the program as a whole.

			Total	2	

Question			Answer/Indicative content	Marks	Guidance
9			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` Examiner's Comments This question assessed whether candidates were able to complete an algorithm to count the number of times that the value 0 appeared in a given array. The first blank to be filled in tested candidates' understanding of the use of a for loop counter to access each index in an array sequentially; this was not answered successfully by many candidates and perhaps shows a lack of understanding. The next blank was more successfully answered and required candidates to increment the variable by 1 which could have been done in several ways, all of which were accepted. Finally, the simplest and most commonly correct answer simply required candidates to print out the value of the variable.
			Total	3	

<table>
<tr><th colspan="3">Question</th><th>Answer/Indicative content</th><th>Marks</th><th>Guidance</th></tr>
<tr><td>10</td><td></td><td></td><td>• 9</td><td>1
AO1 1b
(1)</td><td>Correct answer only
Do not accept 3^2 or 3 x 3

<u>Examiner's Comments</u>

Exponentiation is covered in the specification in section 2.4, Computational logic under the subheading 'applying computing-related mathematics'.

Approximately half of all candidates did not understand the use of this symbol, even when the meaning was given in the question.

The most common incorrect answer was to treat exponentiation as synonymous to multiplication.</td></tr>
<tr><td></td><td></td><td></td><td>Total</td><td>1</td><td></td></tr>
<tr><td>11</td><td></td><td></td><td><table><tr><th>A</th><th>B</th><th>P</th></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td>1</td></tr><tr><td></td><td></td><td>1</td></tr><tr><td></td><td></td><td></td></tr></table></td><td>2
(AO1 1b)</td><td>1 mark for each correct answer in table

'True' or 'T' are also credit worthy.</td></tr>
<tr><td></td><td></td><td></td><td>Total</td><td>2</td><td></td></tr>
<tr><td>12</td><td></td><td></td><td>• radius
• area</td><td>2
(AO1 1b)</td><td>1 mark per bullet up to a maximum of 2 marks.</td></tr>
<tr><td></td><td></td><td></td><td>Total</td><td>2</td><td></td></tr>
</table>

Question			Answer/Indicative content	Marks	Guidance
13			Integer (1)… • …number of seconds not important (1) • … level of accuracy not needed so round to nearest minute (1) • …using a decimal to store seconds (0-60) is not appropriate (1) Real (1)… • … number of seconds may be important (1) • … allows parts/fractions to be stored over integers (1)	1 (AO3 2a) 1 (AO3 1)	One mark for appropriate data type identified. One mark for appropriate justification linked to the data type chosen.
			Total	**2**	
14			• `freeseats` called with `"Red"` • …returned value assigned to variable `redseats`	2	`redseats = freeseats("Red")` "`Red`" must use suitable string delimiters (e.g. speech marks) if directly passing the string. Do not penalise case.
			Total	**2**	
15		i	• 3	1 AO1 1b(1)	CAO
		ii	• 1	1 AO1 1b(1)	CAO
			Total	**2**	

Topic 3

Producing Robust Programs

1(a) The algorithm for one section of a vending machine program is shown in pseudocode.

```
if money >= price then
    venditem()
    giveChange(money – price)
else
    print("Error – not enough money inserted")
endif
```

(i) Give the identifier of **one** variable used in the algorithm.

---

[1]

(ii) State how many parameters are passed into the `giveChange()` subroutine.

---

[1]

(b) A vending machine has the following options available.

Item code	Item name	Price
A1	Crisps, bacon flavour	£0.75
A2	Crisps, salted	£0.75
B1	Chocolate bar	£0.90
C1	Apple pieces	£0.50
C2	Raisins	£0.85

Users insert coins into the vending machine and then enter the two character item code of their selection. If the user has inserted enough money, the vending machine will release the chosen item and output any change required. If the user enters an invalid item code then a suitable error message is displayed.

Draw the vending machine algorithm in the part above as a flowchart.

[5]

(c) When writing the program for the vending machine, maintainability was considered.

(i) Identify **two** ways that the program in the part above has been made more maintainable.

1 ---

---

2 ---

---

[2]

(ii) Give **one** additional way that the maintainability of the program can be improved.

---

[1]

(d) The vending machine is tested before it is released.

(i) Explain the purpose of testing the vending machine.

---

[2]

(ii) Describe the difference between iterative testing and final testing.

---

[2]

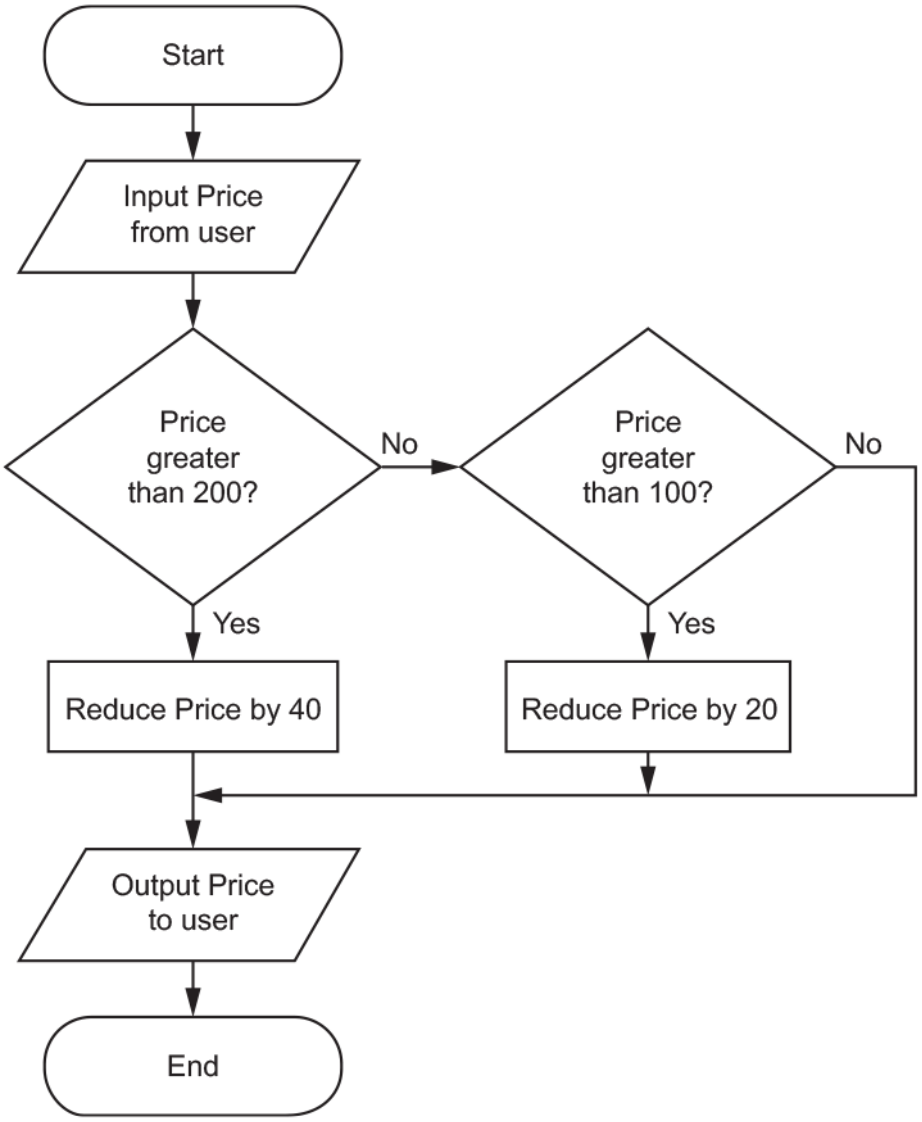
(iii) Complete the following test plan for the vending machine.

Code entered	Money inserted	Expected result
B1	£1	Chocolate bar served, £0.10 change given
	£0.85	Raisins served, no change given
C1		Error – not enough money inserted
C3	£0.75	

[3]

2 OCR Tech is an online shop that sells electronics such as TVs and game consoles.

The following flowchart shows an algorithm to calculate the price of an item during a sale period.



(i) Complete the following test plan for the algorithm.

Price input	Test type	Expected price output
50	Normal	
100	Boundary	
150	Normal	

200	Boundary	
250	Normal	

[3]

(ii) Rewrite the algorithm above.

You must use **either:**

- OCR Exam Reference Language, **or**
- A high-level programming language that you have studied

[6]

3 A car dealership uses a computer system to record details of the cars that it has for sale. Each car has a make, model, age and number of miles driven.

Each car is given a star rating of 1 to 5, based on the age of the car and the number of miles it has been driven. This rating is recorded in the computer system.

(i) Define the term abstraction.

---

[1]

(ii) Give **one** example of how abstraction has been used in the design of this star rating system.

---

[1]

(iii) Explain how authentication could be used as part of the defensive design considerations for this computer system.

---

[2]

4(a) A teacher researches the length of time students spend playing computer games each day.

The teacher writes a program to add up and print out the total number of minutes student 2 played computer games over 5 days (Monday to Friday).

```
total = 0
total = total + minsPlayed[2,0]
total = total + minsPlayed[2,1]
total = total + minsPlayed[2,2]
total = total + minsPlayed[2,3]
total = total + minsPlayed[2,4]
print(total)
```

Refine the program to be more efficient. Write the refined version of the algorithm.

You must use **either**:

- OCR Exam Reference Language, **or**
- a high-level programming language that you have studied.

[4]

(b) The program should only allow values from **0** to **300** inclusive as valid inputs. If the data entered breaks this validation rule, an error message is displayed.

(i) Complete the following program to output "`Invalid input`" if the data does not meet the validation rule.

You must use **either**:

- OCR Exam Reference Language, **or**
- a high-level programming language that you have studied.

```
mins = input("Enter minutes played: ")

if mins < 0 ..................... mins ..................... then

    ..................... ("Invalid input")

endif
```

[3]

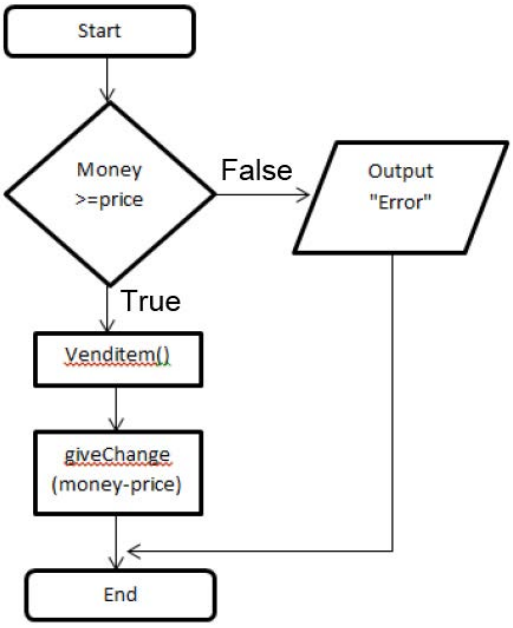
(ii) Complete the following test plan for the program in **(i)**.

Test data	Test type	Expected result
25	Normal	Value accepted
	Invalid	`Invalid input` message displayed
	Boundary	

[3]

END OF QUESTION PAPER

Question			Answer/Indicative content	Marks	Guidance
1	a	i	• `money` • `price`	1 AO1 1b(1)	Must be an identifier, not description. Ignore case.
		ii	• one	1 AO2 1b(1)	<u>Examiner's Comments</u> Question (b)(i) was answered very well. However, (b)(ii) was only answered correctly by a small proportion of candidates. These were also the candidates who generally went on to achieve high marks on this paper. The question asked candidates to state how many parameters are passed into the function from the line `giveChange(money - price)`. Two variables are inside the brackets. Candidates did not recognise this as a calculation. This calculation would be completed before the function call. Only the result is passed into the function as the parameter. This means that the correct answer is one. Most candidates gave the (incorrect) answer of two. **Misconception** Where a sub program (function or procedure) has multiple parameters passed into it, there will be separated by commas – for example `testfunction(x,y)`. A subprogram with a calculation as a parameter will pass the result of this calculation into the sub program.

Question			Answer/Indicative content	Marks	Guidance
	b		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`.  Examiner's Comments Question (c) asked candidates to draw flowchart and this was done particularly well. Many candidates changed the given pseudocode algorithm into a well-defined process that covered the same steps as the pseudocode. A typical answer for this question would be good to use for centres when teaching about using flowcharts. Candidates who use flowcharts for algorithm answers are often not specific enough with the steps required. This gives answers that are generic or high-level. The structure of this question is a good example of the level of detail required.

Question			Answer/Indicative content	Marks	Guidance
	c	i	1 mark per bullet to max 2 • Indentation // whitespace • Appropriately named variables / identifiers • Modularisation / use of subroutines	2 AO2 1b(2)	
		ii	• Comments • Use of constants	1 AO2 1b(1)	

	d	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)

		ii	1 mark per bullet to max 2 • Iterative is during development // repeatedly testing <u>after/while making changes</u> • 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”

<table>
<tr><th colspan="3">Question</th><th>Answer/Indicative content</th><th>Marks</th><th>Guidance</th></tr>
<tr><td></td><td></td><td>iii</td><td>
<table>
<tr><th>Code entered</th><th>Money inserted</th><th>Expected result</th></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td>C2</td><td></td><td></td></tr>
<tr><td></td><td>£0.49 (or any value less that £0.50)</td><td></td></tr>
<tr><td></td><td></td><td>Invalid Selection (or any suitable error message)</td></tr>
</table>
</td><td>3

AO3
2b(3)</td><td>For £0.49 accept any value <£0.50. Must be a specific value, not a description.

Accept any suitable error message for invalid selection</td></tr>
<tr><td></td><td></td><td></td><td>Total</td><td>17</td><td></td></tr>
</table>

<table>
<tr><th colspan="3">Question</th><th>Answer/Indicative content</th><th>Marks</th><th>Guidance</th></tr>
<tr><td>2</td><td></td><td>i</td><td>One mark if two correct, two marks if four correct, three marks if all correct.

<table><tr><th>Price input</th><th>Test type</th><th>Expected price output</th></tr><tr><td>50</td><td>Normal</td><td>50</td></tr><tr><td>100</td><td>Boundary</td><td>100</td></tr><tr><td>150</td><td>Normal</td><td>130</td></tr><tr><td>200</td><td>Boundary</td><td>180</td></tr><tr><td>250</td><td>Normal</td><td>210</td></tr></table></td><td>3</td><td></td></tr>
<tr><td></td><td></td><td>ii</td><td>One mark per bullet point

• Input and store price
• Check if price is > 200…
• …if true, reduce price by 40
• Check if price is >100 and not >200…
• ...if true, reduce price by 20
• Output price</td><td>6</td><td>High-level programming language / OCR Exam Reference Language response required

Do not accept pseudocode / natural language.

BP3 and BP5 only to be given if sensible check for price being over the appropriate threshold. BP4 must check that price is both larger than 100 and not larger than 200; do not give mark for simply checking price is larger than 100. This may be implicit (e.g. using <code>elseif</code>).

e.g.
<pre>price = input("enter price")
if price > 200 then
    price = price – 40
elseif price > 100 then
    price = price - 20
endif
print(price)</pre></td></tr>
<tr><td></td><td></td><td></td><td>Total</td><td>9</td><td></td></tr>
</table>

Question			Answer/Indicative content	Marks	Guidance
3		i	• Hiding / ignoring / removing detail // focussing on certain parts of a problem	1	
		ii	• Focus on age / number of miles • Ignore other factors (such as make, model, etc)	1	Allow other examples of factors to ignore / remove for BP2
		iii	• Ensures only certain users can access the system • Using password / other example of authentication technique	2	Allow other examples of authentication for BP2
			Total	4	

Question			Answer/Indicative content	Marks	Guidance

4	a		• Initialises total as 0 and prints out total the end (as per original program)
			• Uses iteration, e.g. FOR, WHILE
			• …that repeats 5 times
			• …correctly adds up values using loop index
			
			e.g.
			`total = 0`
			`for x = 0 to 4`
			`  total = total + hoursplayed[2, x]`
			`next x`
			`console.writeline(total)`
			
			e.g.
			`total = 0`
			`for x in range (0, 4)`
			`  total += hoursplayed[2][x]`
			`next x`
			`print (total)`
			
			e.g.
			`total = 0;`
			`for (int x = 0; x <= 4; x++){`
			`  total = total + hoursplayed[2][x];`
			`}`
			`System.out.println (total);`
				4 (AO3 2c)	High-level programming language / OCR Exam Reference Language response required
					
					Do not accept pseudocode / natural English.
					
					MP1 must have appropriate identifier, = and then the numeric 0
					MP2 must have `for` or `while`
					MP3 must have the for stopping condition 4/5
					MP4 must have the same identifier for MP1 and equal and + to add the data in the array (using either `[x,y]` or `[x][y]`. This could be `total = total + ….` Or `total += ….`

	b	i	• `or` • `>300 // >= 301` • `print`	3 (AO3 2b)	High-level programming language / OCR Exam Reference Language response required Do not accept pseudocode / natural English. MP2 do not accept ‘greater than’, must use the HLL syntax > or >= MP3 must be a suitable output command word that could be found in a HLL e.g. `print` (Python), `console.writeline` (VB), `cout` (C++)

Question			Answer/Indicative content	Marks	Guidance
		ii	• Suitable invalid test data (i.e. > 300, e.g. 350) • Suitable boundary test data (e.g. 0, 300) • "Value accepted" or equivalent if boundary data 0 or 300 // "Invalid input displayed" or equivalent if boundary data -1 or 301	3 (AO3 2c)	
			Total	10	

Topic 4

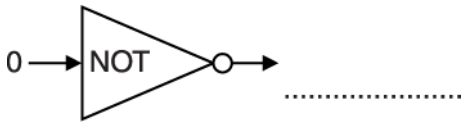
Boolean logic

1 Complete the truth table below for the Boolean statement p = NOT (A AND B).

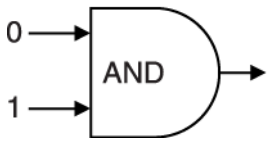
A	B	P
FALSE	FALSE	TRUE
FALSE	TRUE	
TRUE	FALSE	
TRUE	TRUE	FALSE

[2]

2(a) State the output of each of the following logic circuits for the inputs given.



.....



.....

[2]

(b) Fig. 1 is a circuit diagram.

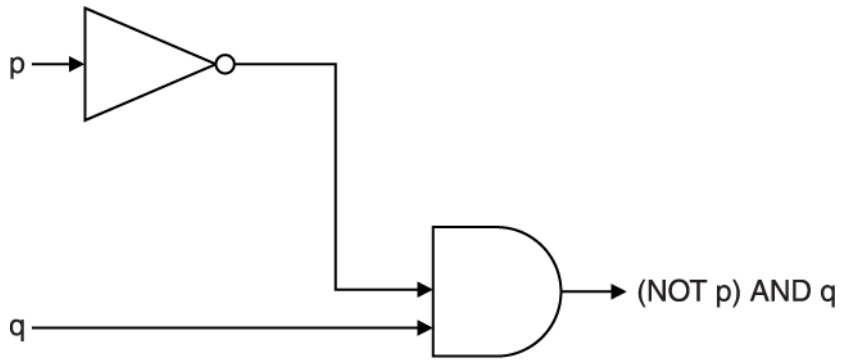


Fig. 1

Complete the truth table for Fig. 1.

p	q	(NOT p) AND q
0	0	0
1	0	0

[3]

3 A cinema uses the following criteria to decide if a customer is allowed to see a film that has a 15 rating:

Customers have to be 15 years of age or older to see the film. They also need to either have a ticket or have the money to buy a ticket.

The table shows the inputs to the system that will output whether the customer can watch the film.

Input	Criteria (True / False)
A	The customer is 15 or over
B	The customer has a ticket
C	The customer has the money to buy a ticket

Draw this system using logic gates.

[2]

4

(i) Draw the logic diagram for the logic system **P = A OR (B AND C)**

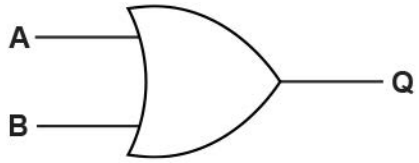
[3]

(ii) Complete the truth table for the logic system **P = NOT (A OR B)**

A	B	P
0	0	1
0	1	
1	0	

[4]

5 Complete the truth table for the following logic gate.



A	B	Q
0	0	0
0	1	1
	0	
1		

[4]

END OF QUESTION PAPER

<table>
<tr><th colspan="3">Question</th><th>Answer/Indicative content</th><th>Marks</th><th>Guidance</th></tr>
<tr><td>1</td><td></td><td></td><td>
<table>
<tr><th>A</th><th>B</th><th>P</th></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td>TRUE</td></tr>
<tr><td></td><td></td><td>TRUE</td></tr>
<tr><td></td><td></td><td></td></tr>
</table>
</td><td>2</td><td>1 mark for each correct answer in table.</td></tr>
<tr><td></td><td></td><td></td><td>Total</td><td>2</td><td></td></tr>
<tr><td>2</td><td>a</td><td></td><td>• 1
• 0.

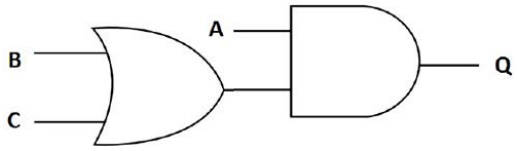
(respectively)</td><td>2</td><td><u>Examiner's Comments</u>

Most candidates answered correctly. As expected, some weaker candidates were less able to work with the logic gates in combination</td></tr>
<tr><td></td><td>b</td><td></td><td>Correct answer:

<table>
<tr><td>p</td><td>q</td><td>(NOT p) AND q</td></tr>
<tr><td>0</td><td>0</td><td>0</td></tr>
<tr><td>1</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>1</td></tr>
<tr><td>1</td><td>1</td><td>0</td></tr>
</table>

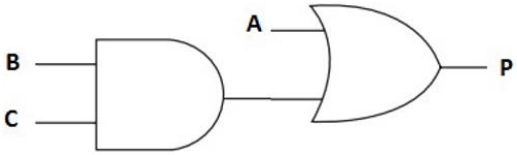
Award marks for

• Correct missing input cases (0 1, 1 1 or 1 1, 0 1)
• Output of 1 for 0 1
• Output of 0 for 1 1.</td><td>3</td><td><u>Examiner's Comments</u>

As expected, some weaker candidates were less able to work with the logic gates in combination</td></tr>
<tr><td></td><td></td><td></td><td>Total</td><td>5</td><td></td></tr>
<tr><td>3</td><td></td><td></td><td>• OR gate with two inputs // AND gate with two inputs
• Diagram as shown in guidance with no additional gates</td><td>2</td><td> 
</td></tr>
<tr><td></td><td></td><td></td><td>Total</td><td>2</td><td></td></tr>
<tr><td>4</td><td></td><td>i</td><td>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</td><td>3

AO1
1b(3)</td><td>Shape must be accurate

 
</td></tr>
</table>

<table>
<tr><th colspan="3">Question</th><th>Answer/Indicative content</th><th>Marks</th><th>Guidance</th></tr>
<tr><td></td><td></td><td>ii</td><td>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</td><td>4

AO1
1b(1)

AO2
1b(3)</td><td>CAO

<table><tr><th>A</th><th>B</th><th>P</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table></td></tr>
<tr><td></td><td></td><td></td><td>Total</td><td>7</td><td></td></tr>
<tr><td>5</td><td></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

<u>Examiner’s Comments</u>

This question asked candidates to complete a truth table for an OR logic gate, with the correct answer being all 1s.

This relatively simple question firstly assessed whether candidates understood the use of truth tables in general, with half the marks being given for the inputs, regardless of the gate.

Secondly, this question assessed whether candidates understood the outputs given by an OR gate with these inputs.

As expected, the majority of candidates were able to gain full marks for both of these areas.</td></tr>
<tr><td></td><td></td><td></td><td>Total</td><td>4</td><td></td></tr>
</table>

Topic 5

Programming Languages and Integrated Development Environments

1(a) Harry is planning to create a computer game using a high-level programming language.

State why the computer needs to translate the code before it is executed.

---

[1]

(b) Harry can use either a compiler or an interpreter to translate the code.

Describe **two** differences between how a compiler and an interpreter would translate Harry's computer game.

---

[4]

2 The area of a circle is calculated using the formula $\Pi \times r^2$, where Π is equal to 3.142 and r is the radius.

Finn has written a program to allow a user to enter the radius of a circle as a whole number, between 1 and 30, and output the area of the circle.

```
01    int radius = 0
02    real area = 0.0
03    input radius
04    if radius < 1 OR radius > 30 then
05    print ('Sorry, that radius is invalid')
06    else
07    area = 3.142 * (radius ^ 2)
08    print (area)
09    end if
```

Finn uses an IDE (Integrated Development Environment) to write his programs. Identify **two** features of an IDE that Finn might use.

---

[2]

3 Describe the advantages of writing a program in a high-level language instead of in assembly language.

[2]

4(a)

A translator is a common tool found in an Integrated Development Environment (IDE).

Describe **two** other common tools or facilities that an IDE can provide.

1 ----------

---

2 ----------

---

[4]

(b) Tick (✓) **one** box in each row to identify whether the statement refers to a high-level language or a low-level language.

Statement	High-level language	Low-level language
Uses English-like keywords such as `print` and `while`		
Must be translated before the processor can execute code		
Code written is portable between different processors		
Requires the programmer to understand the processor's registers and structure		

[4]

5(a) A computer game is written in a high-level programming language.

State why the computer needs to translate the code before it is executed.

[1]

(b) Either a compiler or an interpreter can translate the code.

Describe **two** differences between how a compiler and an interpreter would translate the code.

1

2

[4]

END OF QUESTION PAPER

Question			Answer/Indicative content	Marks	Guidance
1	a		• To convert it to binary / machine code • The processor can only understand machine code	1	Maximum 1 mark
	b		• Compiler translates all the code in one go… • … whereas an interpreter translates one line at a time • Compiler creates an executable… • …whereas an interpreter does not / executes one line at a time • Compiler reports all errors at the end… • … whereas an interpreter stops when it finds an error	4	1 mark to be awarded for the correct identification and one for a valid description up to a maximum of 4 marks. No more than 2 marks for answers relating only to interpreters and no more than 2 marks for answers only relating to compilers.
			Total	5	
2			• Error diagnostics (any example) • Run-time environment • Editor (any feature such as auto-correct, auto-indent) • Translator • Version control • Break point • Stepping	2	1 mark per bullet to a maximum of 2 marks. Only 1 example per bullet, e.g. auto-correct and auto-indent would only gain 1 mark.
			Total	2	
3			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 compute	2 AO1 1b(2)	Accept “human language” as English for BP4 “Easier to use” is too vague. <u>Examiner’s Comments</u> Question (b) was answered extremely well. Many candidates were able to articulate the advantages of high-level languages over assembly language. Even more pleasing were the many candidates able to discuss issues such as portability of code and that a high-level instruction translates into many assembly instructions.
			Total	2	

<table>
<tr><th colspan="3">Question</th><th>Answer/Indicative content</th><th>Marks</th><th>Guidance</th></tr>
<tr><td>4</td><td>a</td><td></td><td>1 mark per bullet, max 4

e.g.

• Editor
• …to enable program code to be entered / edited
• Error diagnostics / debugger
• …to display information about errors / location of errors / suggest solutions
• Run-time environment
• …to enable program to be run / to check for runtime errors / test the program</td><td>4</td><td>Allow other tools available in an IDE with suitable expansion (e.g. breakpoints, watch window, stepping, pretty printing, etc)</td></tr>
<tr><td></td><td>b</td><td></td><td>One mark per row

<table><tr><th>Statement</th><th>High-level language</th><th>Low-level language</th></tr><tr><td>Uses English-like keywords such as <code>print</code> and <code>while</code></td><td>✓</td><td></td></tr><tr><td>Must be translated before the processor can execute code</td><td>✓</td><td></td></tr><tr><td>Code written is portable between different processors</td><td>✓</td><td></td></tr><tr><td>Requires the programmer to understand the processor’s registers and structure</td><td></td><td>✓</td></tr></table></td><td>4</td><td>Accept other markings that indicate a choice has been made (e.g. a cross, etc)</td></tr>
<tr><td></td><td></td><td></td><td>Total</td><td>8</td><td></td></tr>
</table>

Mark Scheme

Question			Answer/Indicative content	Marks	Guidance
5	a		• To convert it to binary/machine code • The processor can only understand machine code	1 (AO1 1a)	Maximum 1 mark
	b		• Compiler translates all the code in one go… • …whereas an interpreter translates one line at a time • Compiler creates an executable… • …whereas an interpreter does not/executes one line at a time • Compiler reports errors at the end… • …whereas an interpreter stops when it finds an error	4 (AO1 1b)	1 mark to be awarded for the correct identification and one for a valid description up to a maximum of 4 marks. No more than 2 marks for answers relating only to interpreters and no more than 2 marks for answers only relating to compilers.
			Total	5	