Name:

Class Teacher:

Date:

OCR J276

GCSE Computer Science

REVISION BOOKLET

2.1 ALGORITHMS

Content in J276 GCSE Computer Science:

1.1 Systems Architecture
1.2 Memory
1.3 Storage
1.4 Wireless and Wired Networks
1.5 Network Topologies, Protocols and Layers
1.6 System Security
1.7 Systems Software
1.8 Ethical, Legal, Cultural and Environmental Concerns
2.1 Algorithms
2.2 Programming Techniques
2.3 Producing Robust Programs
2.4 Computational Logic
2.5 Translators and Facilities of Languages
2.6 Data Representation

2.1 ALGORITHMS

TOPIC	☹	😐	☺
Computational Thinking:			
Abstraction			
Decomposition			
Algorithmic Thinking			
Standard Searching Algorithms:			
Binary Search			
Linear Search			
Standard Sorting Algorithms:			
Bubble Sort			
Merge Sort			
Insertion Sort			
How to produce algorithms using:			
Pseudo code			
Flow diagrams			
Interpret, Correct or Complete algorithms			

2.1 ALGORITHMS

COMPUTATIONAL THINKING

ABSTRACTION

DECOMPOSITION

ALGORITHMIC THINKING

STANDARD SEARCHING ALGORITHMS

BINARY SEARCH

LINEAR SEARCH

STANDARD SORTING ALGORITHMS

BUBBLE SORT

MERGE SORT

INSERTION SORT

HOW TO PRODUCE ALGORITHMS USING:

PSEUDO CODE

FLOW DIAGRAMS

INTERPRET, CORRECT OR COMPLETE ALGORITHMS

EXAM QUESTIONS

QUESTION 1

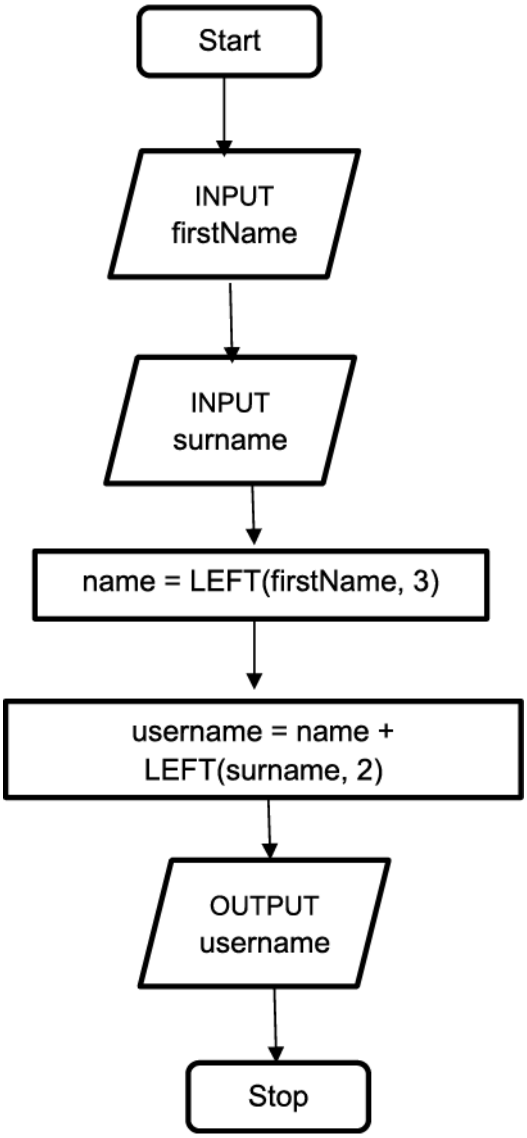
There is a subroutine, HEX(), that takes a denary number between 10 and 15 and returns the corresponding hexadecimal number, e.g. HEX(10) would return “A”, HEX(15) would return “F”. Write an algorithm, using the subroutine HEX(), to convert any whole decimal number between 0 and 255 into a 2 digit hexadecimal number.

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.

QUESTION 2

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

2	a		• RebEl	1	Correct Answer Only (allow any case)

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?

	b	i	• UitFr	1	Correct Answer Only (allow any case)

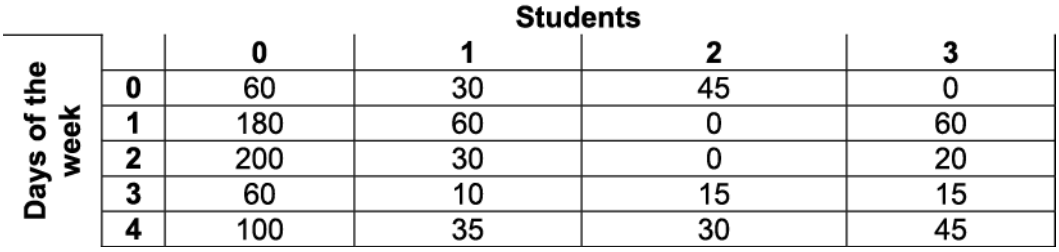
Write an algorithm for Johnny to output a username using the updated process.

		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

QUESTION 3

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



		Students			
		0	**1**	**2**	**3**
Days of the week	**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. Write the code to output the number of minutes student 0 played computer games on Wednesday.

3	a	i	print (hoursPlayed[0,2])	1	Correct Answer Only

[1]

State the output if Heath runs the code: **`print(hoursPlayed[2,1])`**.

		ii	0	1	Correct Answer Only

State the output if Heath runs the code:

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

		iii	80	1	Correct Answer Only

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

		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

Heath has the day of the week stored as a number, e.g. 0 = Monday, 1 = Tuesday. Write a sub-program that takes the number as a parameter and returns the day of the week as a string.

	b		• Appropriate declaration of a function that takes day number as parameter and returns day • Use of selection (if / switch) • Appropriate comparison • Correct identification of each day • Case default e.g. function returnDay(dayNo As String) As String switch dayNo case 0: returnDay = “Monday” case 1: returnDay = “Tuesday” case 2: returnDay = “Wednesday” case 3: returnDay = “Thursday” case 4: returnDay = “Friday” case default: returnDay = “Invalid” endswitch endfunction	5	1 mark per bullet to a maximum of 5. If used, a flowchart should represent the bulleted steps in the answer column.

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.

<table>
<tr><td></td><td>c</td><td></td><td>• 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)</td><td>6</td><td>Accept any type of average calculation (mean, median, mode).

If used, a flowchart should represent the bulleted steps in the answer column.</td></tr>
</table>

QUESTION 4

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

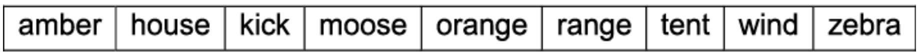
crime	bait	fright	victory	nymph	loose

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

<table>
<tr><td>4</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>
</table>

Willow has created a hangman program that uses a file to store words the program can select from. A second sample of data is shown in **Fig 4**.

Fig. 4



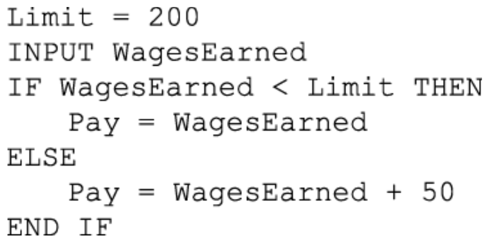
amber	house	kick	moose	orange	range	tent	wind	zebra

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

	b		• 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	4	1 mark per bullet (multiple ways through, marks awarded for appropriate comparison and creation of sub groups).

QUESTION 5

Jim is writing a program to calculate the wages of workers in a teddy bear factory. Workers sometimes get a £50 bonus. Here is an 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
```

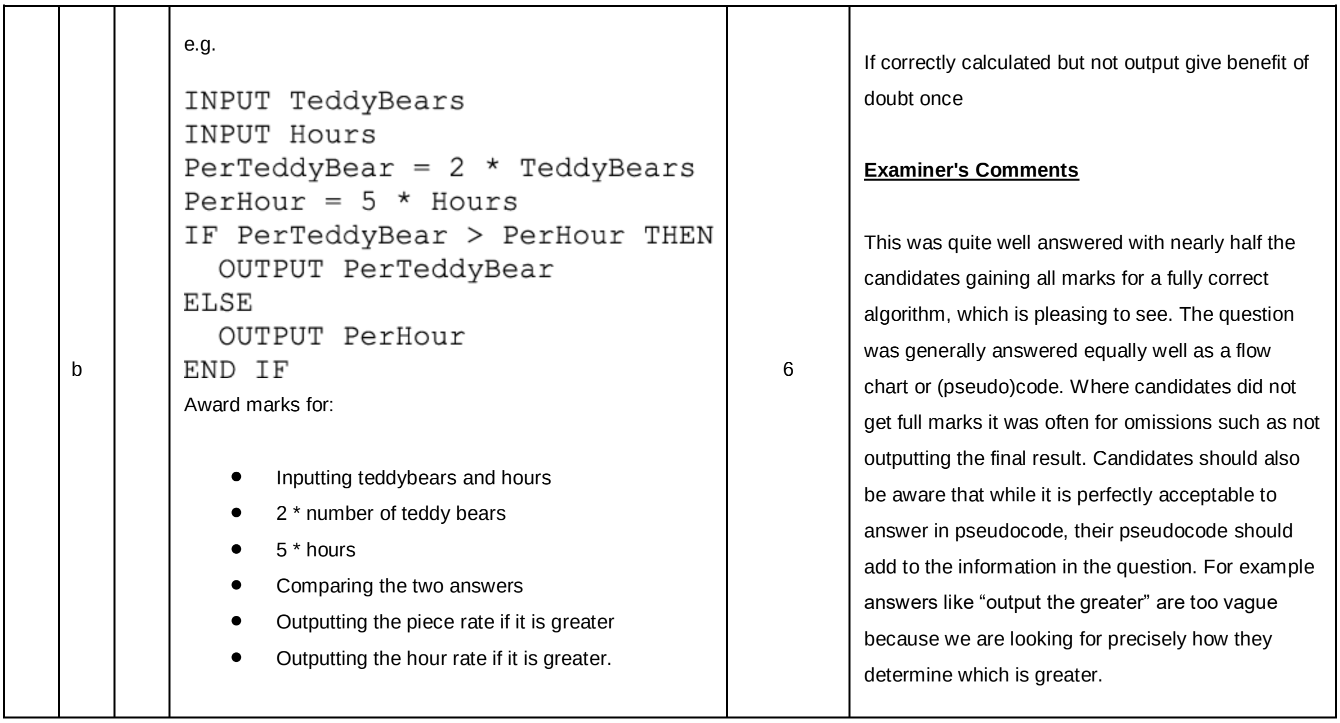
State the value of Pay after this code is executed for each of the following values of WagesEarned.

- WagesEarned = 50 Pay = ...
- WagesEarned = 200 Pay = ...

5	a		• 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.

The wages earned by a worker is either £2 for every teddy bear they have made or £5 for every house they have worked, whichever is larger. Write an algorithm that:

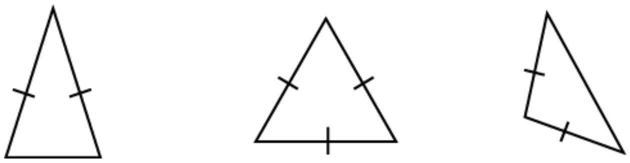
- allows the user to input the number of teddy bears made and the number of hours worked
- calculates the wages for the number of teddy bears made
- calculates the wages for the number of hours worked
- outputs the larger of the two results



	b		e.g.
			`INPUT TeddyBears`
			`INPUT Hours`
			`PerTeddyBear = 2 * TeddyBears`
			`PerHour = 5 * Hours`
			`IF PerTeddyBear > PerHour THEN`
			`  OUTPUT PerTeddyBear`
			`ELSE`
			`  OUTPUT PerHour`
			`END IF`
			Award marks for:
			• Inputting teddybears and hours
			• 2 * number of teddy bears
			• 5 * hours
			• Comparing the two answers
			• Outputting the piece rate if it is greater
			• Outputting the hour rate if it is greater.
				6	If correctly calculated but not output give benefit of doubt once
					Examiner's Comments
					This was quite well answered with nearly half the candidates gaining all marks for a fully correct algorithm, which is pleasing to see. The question was generally answered equally well as a flow chart or (pseudo)code. Where candidates did not get full marks it was often for omissions such as not outputting the final result. Candidates should also be aware that while it is perfectly acceptable to answer in pseudocode, their pseudocode should add to the information in the question. For example answers like "output the greater" are too vague because we are looking for precisely how they determine which is greater.

QUESTION 6

An isosceles triangle is a triangle that has at least two equal sides. The diagram below shows examples of isosceles triangles. In each diagram, the marked sides are equal.



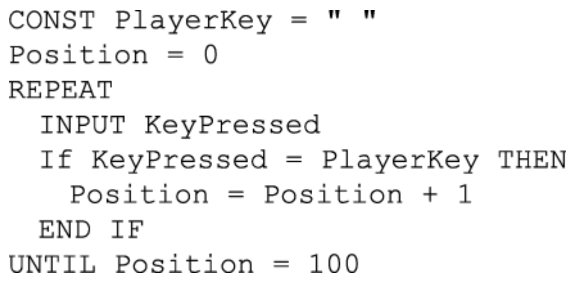
Write an algorithm for a computer program that determines whether a triangle is an isosceles triangle.

- The user inputs the length of the three sides as Length1, Length2 and Length3
- If any two sides have the same length the program outputs “Isosceles”
- Otherwise the program outputs “Not Isosceles”

6			Example INPUT Length1 INPUT Length2 INPUT Length3 IF Length1 = Length2 THEN Output “Isosceles” ELSE IF Length1 = Length3 THEN Output “Isosceles” ELSE IF Length2 = Length 3 THEN Output “Isosceles” ELSE OUTPUT “Not Isosceles” END IF END IF END IF *Award marks for:* • Inputting three lengths • Comparing lengths in pairs • … for all three ways correctly • … outputting “Isosceles” for all valid cases • … outputting “Not Isosceles” for all cases and only in cases where the three lengths are different.	5	There are various ways to implement this but the two most common methods will be the method shown or one disjuncted IF statement (ie IF Length1 = Length2 OR Length1 = Length3 OR Length2 = Length3). In all cases, apply the criteria in the last 4 bullet points to the whole algorithm to determine the mark. **<u>?Examiner's Comments</u>?**? This question was generally well answered with most candidates obtaining 4 or 5 marks out of 5. Candidates not gaining the highest marks often made errors in writing an imprecise condition for the IF statements such as “Length1 = Length2 OR Length3”. While algorithms were acceptable in pseudocode, flowchart or code, the pseudocode of some of the candidates was so vague that it did not add anything to the specification in the question. Several candidates had innovative ways of determining whether the sides were equal and it was pleasing to see this creativity. Centres should advise candidates that when asked to give an algorithm to a specification, they read and follow the specification carefully. Some people do not class an equilateral triangle as an isosceles triangle, but the specification in this question made it clear that they should. Some candidates added additional constraints and while, on the whole they were not disadvantaged in this case from deviating from the specification, it is important that as programmers they learn to stick to a specification given.

QUESTION 7

Julie is writing a computer game that simulates a 100m 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.



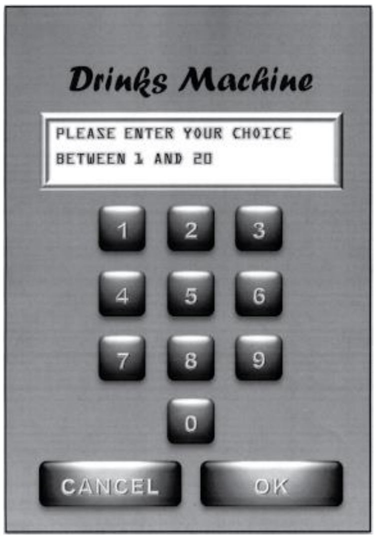
```
CONST PlayerKey = " "
Position = 0
REPEAT
  INPUT KeyPressed
  If KeyPressed = PlayerKey THEN
    Position = Position + 1
  END IF
UNTIL Position = 100
```

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.

7			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 **?<u>Examiner's Comments</u>?**? 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).

QUESTION 8

A free drinks machine in an office provides 20 different drinks.



The machine has a small keypad with keys 0 to 9, OK and CANCEL. It also has a small LCD screen, which can display a short message. To get a drink, users select an item number between 1 and 20 with the keypad and confirm their choice by pressing OK. If they make a mistake they can press the CANCEL button and start again. If the selection is valid and the drink is available it dispenses the drink. The display screen is used to show suitable short messages throughout the process. Write an algorithm for the process described above. The quality of written communication will be assessed in your answer.

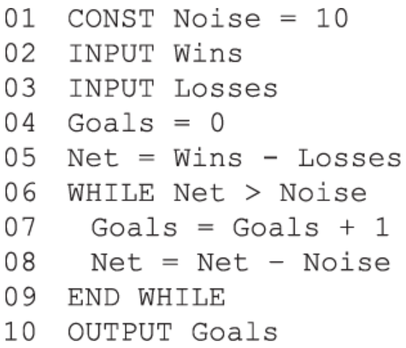
8			Example:
			`Choice = “”`
			
			`REPEAT`
			`  INPUT Button`
			`  IF Button is between 0 and 9 THEN`
			`    Choice = Choice & Button`
			`    OUTPUT Choice`
			`  ELSE IF Button = CANCEL THEN`
			`    Choice = “”`
			`  END IF`
			`UNTIL Button = OK`
			
			`IF Choice is between 1 and 20 THEN`
			`  IF drink chosen available THEN`
			`    Dispense drink`
			`    OUTPUT “Collect your drink”`
			`  ELSE`
			`    OUTPUT “Drink not available”`
			`  END IF`
			`ELSE`
			`  OUTPUT “Invalid selection”`
			`END IF`
			
			`Wait`
			`OUTPUT “Ready”`
				6	High Level Response (5/6):
					A clear and complete algorithm with correct input, validation and reasonable output / outcome (accept minor errors).
					Algorithm presented in algorithm in code, pseudocode or as a flowchart with correct conventions used to make it clear (e.g. indentation, shapes of flow chart objects). Technical terms are used correctly and there are few, if any, errors in spelling.
					
					Medium Level Response (3/4):
					An algorithm that deals with input, validation and reasonable output / outcome but there may be some logical errors. Algorithm may be in code, pseudocode, flowchart, or very well structured English (e.g. clear bulleted steps) using some accepted conventions, although this may not be consistent. Technical terms are mainly correct and there may be occasional spelling errors.
					
					Low level response (1/2):
					A description of the Input, validation and output

					required, but some may be missing. Response may be in English or a poorly structured code / flowchart. Limited, if any, use of technical terms and errors of spelling may be intrusive. 0: Response not worthy of credit **Examiner's Comments** Once again, it is important to emphasise that QWC does not mean candidates are required to write essays. As part of a computer science qualification, it is important to assess the candidates’ technical writing skills as relevant to computer science, including their ability to select and use the most appropriate form and register of written communication for the question set. In a QWC question, this is assessed in a holistic manner alongside the correctness and accuracy of the answer using levels of response. In this question, the most appropriate form of written communication for an algorithm was evidently pseudocode or a flowchart. A few candidates attempted to give their algorithms in prose which reduced the overall quality of the response, even when the logic of the algorithm was largely correct. That said, most candidates did answer in a flow chart, pseudocode or code in a language they have studied. Candidates who used a flow chart seemed to score better. They were less likely to omit parts of the specification or to create errors in their logic by incorrectly nesting branching structures. Overall, the question discriminated well between candidates of different abilities, with weaker candidates tending to either make an error in their logic, usually with the validation, or ignore parts of the question. Most often, candidates omitted checking that drinks were available before dispensing them and/or omitted to actually dispense the drink. Checking the algorithm with the requirements of the question may have prevented this. As well as the correctness of the algorithm, examiners considered the effectiveness of the written communication including for example the use of meaningful identifiers, consistent and clearly labelled symbols in flow charts or indentation in pseudocode. Centre’s should also note that, as is often the case with QWC question, this question was intentionally open ended and candidates could adopt a variety of approaches such as considering the input of the system as a continuous input stream (as in the example given in the published mark scheme) or a completely event driven system with each key having its own logic or a hybrid of these approaches mimicking an

					interactive console application with the OK button serving as an Enter key and the Cancel button clearing the input buffer. All of these approaches were equally valid.

QUESTION 9

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.



```
CONST Noise = 10
INPUT Wins
INPUT Losses
Goals = 0
Net = Wins - Losses
WHILE Net > Noise
  Goals = Goals + 1
  Net = Net - Noise
END WHILE
OUTPUT Goals
```

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

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. **Examiner's Comments** 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, for example by assuming that "Net" means the ball has touched the net and equating it to the number of goals.

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

	b		• A location in memory to store / a value that may change (as the program is running) • eg Wins / Losses/ Net / Goals	2	

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

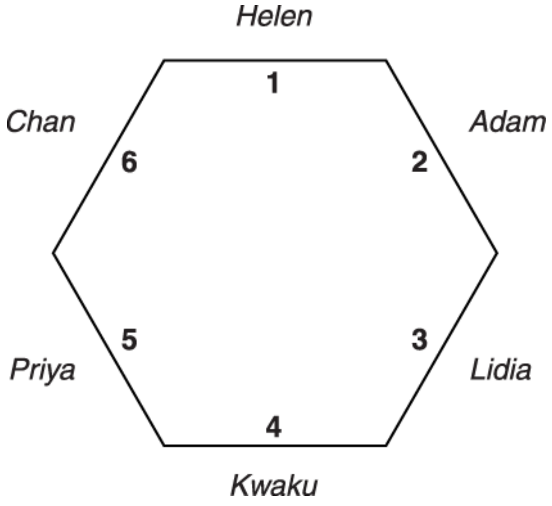
	c		• Net = 5 which is less than Noise • Goals = 0	2	1 mark for the subtraction **and** result of the comparison 1 mark for correct result

- Wins = 20 Losses = 5

			• Net = 15 which is greater than Noise • Runs Loop once {Goals = Goals + 1, Net = Net ? Noise}… • Goals = 1	3	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.

QUESTION 10

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).

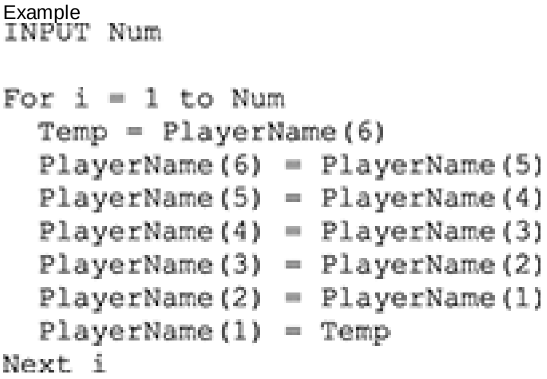
10	a		• Lidia	1	Accept incorrect spelling if intention is clear.

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

	b		• 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	2	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.

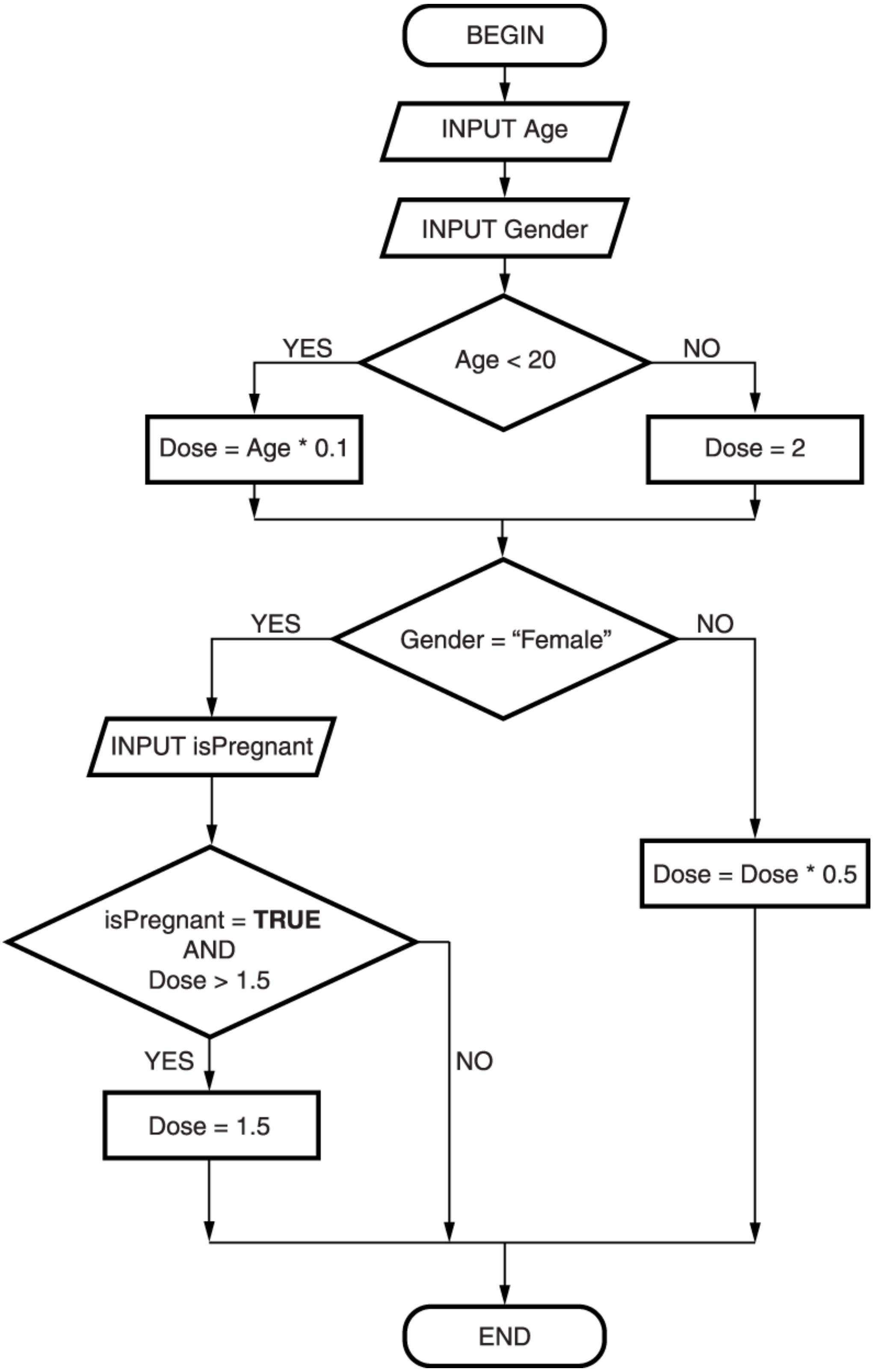
During the game, each player sometimes moves clockwise by a given number of places. For example, if the number of places is 2, Helen will move to seat 3, Priya will move to seat 1 etc. Write an algorithm that will update the contents of the array PlayerName after a move has occurred. Your algorithm should:

- 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

	c		Example  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 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)	6	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.

QUESTION 11

A computer program calculates the correct dose in grams of a type of medicine. The algorithm used is shown by the flow diagram below.



Use the flow diagram to calculate the correct dose of medication for a male aged 30. You must show your working.

11	a		• (Age < 20 is FALSE so) Dose = 2 • (Gender = "Female" is FALSE) so Dose = Dose * 0.5 • … therefore Dose = 1	3	**Award mark for first bullet only if 2 clearly refers to the dose.** Allow follow through error for second and third bullet. i.e. if candidate has the wrong dose they can still get a mark for Dose * 0.5 and for doing this calculation correctly. (Typically 3 * 0.5 = 1.5 which is therefore worth 2 marks)

Use the flow diagram to calculate the correct dose of medicine for a pregnant female aged 19. You must show your working.

	b		• (Age is less than 20 = true) so Dose = 0.1 * Age • 1.9 • [is Pregnant AND Dose > 1.5] is TRUE • Dose = 1.5	4	**Candidates do not need to refer to dose, provided it is clear that they are performing the correct operation.** For 3rd bullet it is sufficient if the candidate has shown that both isPregnant and (Dose > 1.5) are TRUE (This may not be at the same point in the answer and they do not need to explicitly state the result of the AND).

QUESTION 12

A memory game is played where:

- three players (A, B and C) have to choose a number between 0 and 100
- if the number has already been chosen, a message is displayed that says "taken"
- if the number has not already been chosen, the players letter is placed next to it
- the quantity of numbers that have not yet been chosen is displayed

The winner is the player who has chosen the most unique numbers by the end of the game. The numbers are stored in an array; **`numbers()`**. A number that has not yet been chosen is stored as an empty string `""`. The players are represented by **`"A"`**, **`"B"`** and **`"C"`**. Fig 5 shows an extract from the array.

Number:	0	1	2	3	4	...	...	99	100
Player:	A	C	B		A			B	

You have been asked to program part of the game. Write an algorithm for player A's turn, which:

- takes as an input the number that player A chooses
- if it has not already been chosen, stores an **`"A"`** in that array element
- if it has already been chosen, outputs **`"taken"`**
- counts and outputs the quantity of numbers left that have not been chosen

12			1 mark per bullet • Taking the move as input • Checking if array element input is free … ○ …Outputting if it is taken • Writing "A" to the correct array element • Counting how many free space there are… ○ …Outputting the number of free spaces (if good attempt at counting free spaces)	6	The output mark can only be awarded if a reasonable attempt at adding the free spaces have been performed Counting how many free spaces there are can be done by either: • Looping through each element of the array and updating a variable if free / taken • Subtracting 1 each time an element is taken (this must work, i.e. there is no initialisation of the variable e.g. to 101, as that would run every time and reset the variable). If Initialisation is used, this must be outside a loop and must be 101. **Examiner's Comments**

			e.g. INPUT move IF numbers(move) = “” then numbers(move) = “A” ELSE output “taken” ENDIF free = 0 FOR x = 0 TO 100 IF numbers(x) = “” then free = free + 1 ENDIF NEXT x OUTPUT free e.g. INPUT move IF numbers(move) = “” then numbers(move) = “A” numberFree = numberFree - 1 ELSE output “taken” ENDIF OUTPUT numberfree		Candidates were required to write an algorithm to access specific array elements and then either keep track of the number of taken elements, or to loop through and count the number not taken. Most candidates were able to take the number as input. Few candidates had a good understanding of arrays and how to access specific array elements. Some candidates attempted to keep track of the number of spaces taken by adding 1 to a variable each time through, but a common mistake was to also reset this value each time so that it was not actually keeping track correctly. Many candidates who tackled this question used pseudo code and often made a better attempt at the question. When a flow chart was used, there was rarely any use of arrays and accessing the array elements.