

2024 ASSESSMENT REPORT

ITC315118 COMPUTER SCIENCE

Written Component

Section A – Criterion 1

Question 1

- a) Almost all students got full marks.
- b) Almost all students got full marks.
- c) This was generally well answered.
For full marks, candidates were expected to handle both boundary cases. Some students described the condition as “more than 6” and “less than 100” people attending, rather than “at least 6” and “no more than 100”.
- d)
 - i. The best responses described the problem in general terms.
Partial marks were awarded for responses that described a specific situation (e.g. an odd number of people attending, fewer people attending than the items in a single pack).
 - ii. This was generally answered well, although some responses focused on changing the scenario (e.g. only allowing multiples of 6 to attend) rather than changes to the algorithm.

Question 2

- a) Generally answered well.
- b) Less than half of the responses correctly identified the logic error.
A significant number of candidates left this question blank.
Partial marks were awarded for responses that identified one of the spelling mistakes in the variable names.
- c) Very few candidates obtained full marks for this part. The best responses included:
 - time input (1 mark)
 - time validation (1 mark)
 - calculate and display total cost.

Question 3

- a) This was attempted by most students, with about half the candidates earning full marks.
In the best responses, the components (buttons, text fields, dropdown, etc.) were clearly identified and it was obvious how a user would enter the required information.
- b) This was attempted by only about half of candidates, with a wide spread of all possible marks, and only two responses receiving full marks.
The most successful responses had algorithms that typically used one of two approaches:
 - the interface has inputs for a single treasure, and the algorithm updates the score as the treasures are entered one by one

- the interface has inputs for each of the nice different combinations of type and size and calculates the score accordingly.

The best responses included these features:

- initially/when structure (1 mark)
- appropriate variable names (1 mark)
- type and size points calculated correctly (3 marks)
- bonus points calculated correctly (3 marks)
- required values and messages displayed (2 marks)
- the algorithm works correctly no matter what order the buttons or text fields, etc, are used (1 mark).

Section B – Criterion 2

Question 4

- a)
- Completed correctly by almost all candidates.
 - Completed correctly by almost all candidates; very few omitted an answer for each question.
- b)
- About two thirds of candidates successfully completed this part.
 - About half successfully answered this part.
- c) Most candidates received full marks for this part.
Those who didn't selected an incorrect response but then did not show any working out so they could not be awarded partial marks.
- d) As with Question 4c, most candidates received full marks for this question.
Those who didn't selected an incorrect response but then did not show any working out so they could not be awarded partial marks.
- e) About 20% of candidates received full marks for this question.
Some circled the error(s) but did not provide an explanation, so partial marks were awarded.
- f) Very few candidates received full marks for this question. Most gave an incorrect final value for **i** and were awarded partial marks.
- g) About half of the candidates received full marks for this question.
Some students attempted the code but were unable to produce the correct code to test when input Number was even (or odd) and were awarded partial marks.
About a quarter of students made no attempt at this question.

Question 5

- a)
- Was completed correctly by most candidates.
 - Most candidates received full marks for this question. Some candidates gave the correct answer 42.0 but no explanation or an incorrect explanation and were awarded partial marks.
- b) A little under a half of all candidates received full marks for this question. Some candidates did not recognise that `secondArray` was of type **long** which stores integer, hence the rounded-up

values needed to be whole numbers and received partial marks. Many students did not attempt this question.

- c) Completed correctly by only one candidate. Some candidates were able to successfully provide the value of newWord but did not complete the trace table correctly as there was some misunderstanding about the result of new String(charArray, 0 , i) which in the first iteration of the for loop would return zero characters.
Those who attempted the trace were awarded partial marks. About one third of candidates did not attempt this question.

Question 6

- a) About half of all candidates attempted these four questions.
About 25% of candidates received full marks for each of the four questions.
Marks were deducted for incorrect return types not corresponding to the method call from the main class or other incorrect coding.
- b) Only eight candidates were able to gain full marks for this question.
A further five provided solutions which were awarded partial marks.
About 70% of candidates did not attempt this question.

Section C – Criterion 3

Question 7

- a)
- About half of all students selected “6” as their answer, the other half “7”. Whether a constructor is considered a metho depends on your definition of a constructor. Both answers were accepted.
 - Well done by almost all candidates.
- b)
- Well done by almost all candidates.
 - Well done by almost all candidates.
- c) Candidates found this typical “C” question more difficult than 7d. About half of candidates used incorrect syntax, referenced the instance as “Tim” rather than athlete1, or did not answer the question at all.
- d) More than three quarters of candidates gained full marks here. Well done!
- e) Half of candidates achieved full marks. Some made reference to 7d and said that they had already completed a method similar to the setCurrentMedals() method. There was confusion as to whether currentMedals and extraMedals were to be separate variables, or just the single combined variable medals, with candidates equally divided as to how to interpret the question.

Question 8

- a)
- Half of all candidates gained full marks for this question. Common errors were incorrect syntax and not using book1 and book2 for the instances.
 - Half of candidates gained full marks. The main error was not using the “.” notation correctly.

- b) This question was well answered with almost half of candidates gaining 3 marks. Many candidates only included the parameter and variable for AuthorCountry, but all the other variables needed to be included for full marks.
- c) To receive 3 marks candidates needed to explain that a 2D array of Book had been created (declared) and 4 book instances had been placed in the array (instantiated).
- d) Many answers were awarded full marks here. These included:
 - Creating a GUI to display the books in a bookshelf style.
 - Using nested for loops to access the bookArray elements so they could be displayed.
 - Writing a displayBookArray() method that could accept the bookArray and print all the books within.
 - Some sort of database library system so that values in the bookArray could be examined, perhaps using search fields and Button.

Question 9

Around half of all candidates did not attempt this question. Among those who did, there were some excellent answers with a wide variety of approaches to handling the AppGUI class and how the active team was stored and used. Well done!

Marks were awarded for the following:

- correct class structure, including initialising variables and correct use of methods (2 marks)
- correct passing and use of parameters, within the AFL class and in the main method (3 marks)
- writing the calculateScore() method correctly (1 mark)
- writing the printDetails() method correctly (2 marks)
- writing the main method, including getting information from the AppGUI class (2 marks)
- dealing with whichever team was active when adding goals, behinds, or calculating a team's score (2 marks).

Section D – Criterion 4

Question 10

- a) Answered correctly by almost all candidates.
- b) Answered correctly by almost all candidates.
- c)
 - i. Answered correctly by almost all candidates.
 - ii. Answered correctly by most candidates.
- d)
 - i. If ii) was correct then this answer was correct.
 - ii. Some candidates stopped halfway with their simplifying and could have gone further.
- e) Some candidates lost marks for incorrectly completing the truth table.
- f)
 - i. Candidates lost marks for not mentioning “visual method” in their answer.
 - ii. Candidates lost marks here for just giving the answer and not doing the Karnaugh map. Some got confused and put in truth tables instead which also lost marks.

- g)
 - i. Needed to see the wording “transferring data” in the explanation.
 - ii. Needed to see bottleneck/slow down/decreased speed or something discussing the slowing down.

Question 11

- a)
 - i. Candidates lost some marks for not fully simplifying the logic expressions and some lost marks for using brackets but not adjusting the AND/OR statement when placing the NOT outside of the brackets.
 - ii. Candidates lost some marks for not fully simplifying the logic expressions and some lost marks for using brackets but not adjusting the AND/OR statement when placing the NOT outside of the brackets.
- b) Many partial marks awarded here.
- c)
 - i. Majority of the candidates attempted this question with about 80% getting it correct.
 - ii. Almost 70% of the candidates got some marks in this question with about 50% of candidates getting full marks.
 - iii. If the candidate went well in the previous question, they usually got this one correct.
 - iv. About 70% of candidates got this answer correct.

Question 12

- a)
 - i. Of those who attempted, most were able to trace R[1] and R[2] correct but then lost marks for the final part.
 - ii. Of those who attempted, most were able to trace R[1] and R[2] correct but then lost marks for the final part.
- b) Almost half the candidates attempted this in some form yet not many got full marks.
- c) Less than half of the candidates got this correct with a large percentage not even trying.

Section E – Criterion 5

Question 13

- a)
 - i. Almost all students got full marks.
 - ii. Almost all students got full marks.
- b)
 - i. Almost all students got full marks. Partial marks were given for calculating the correct number but circling the negative.
 - ii. Almost all students got full marks.
- c)
 - i. Almost all students got full marks.

- ii. Almost all students got full marks.
- d) Partial mark given for correctly circled answer and full marks for including explanation.
- e) Almost all students got full marks.

Question 14

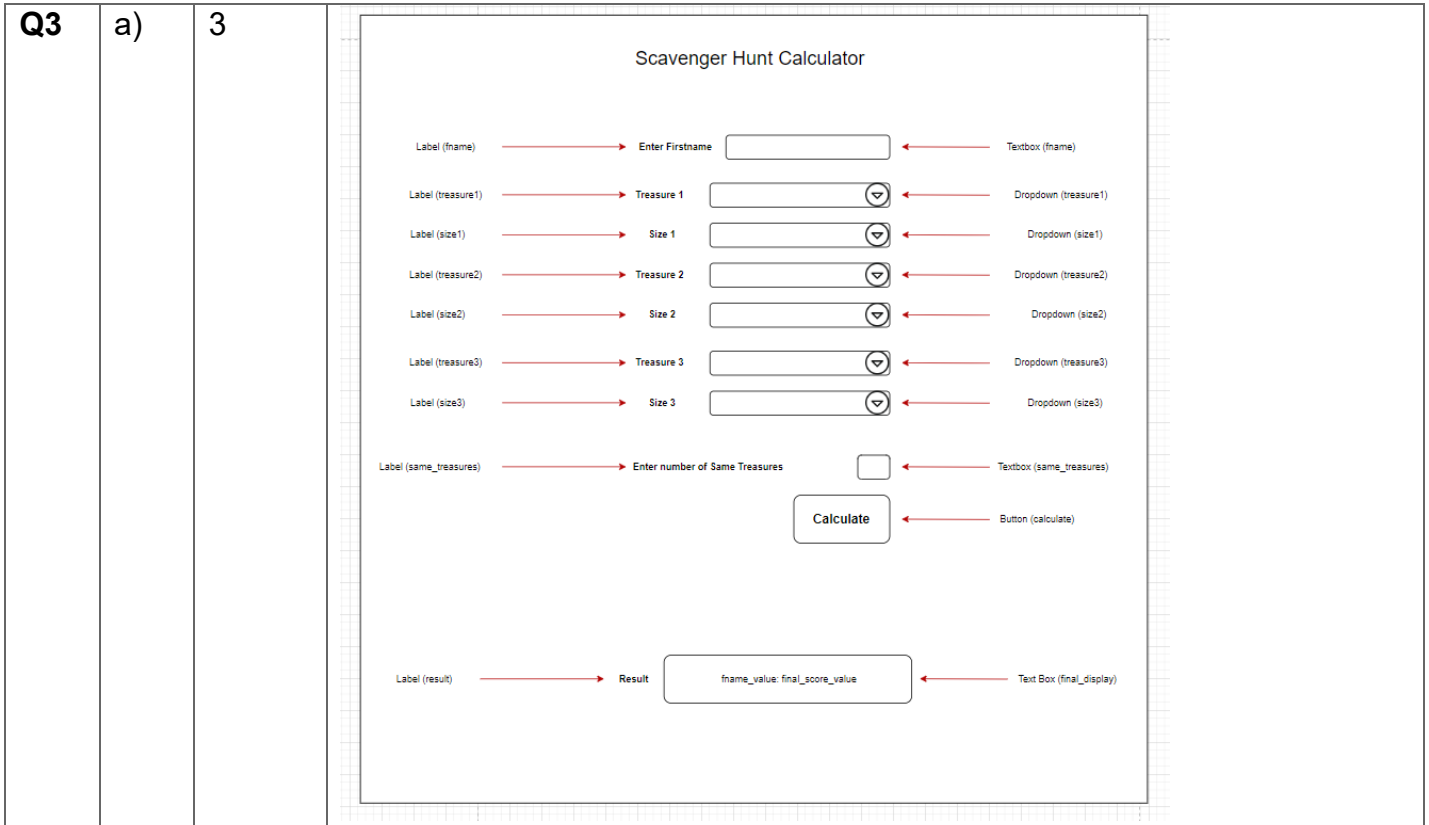
- a) Students were expected to explain the issue of overflow using the example. Most student met the requirement while some just attempted the calculation.
- b) Generally answered well.
- c) Partial marks for showing the results of the example. Full marks explaining the shift-right effect.
- d) Partial marks were given for finding the fraction only.

Question 15

- a) Partial marks were given for identifying and calculating parts of the floating-point value.
- b) Many students attempted this problem though few correctly competed it. Some students were thrown by the comma in the filter range value.
- c)
 - i. Diagram copied from the Information Book was sufficient.
 - ii. Most students who correctly drew the diagram from part i were able to correctly place the value.
 - iii. This was attempted by many students but only 3 received full marks. Partial marks were given for showing understanding of addressing.

Sample Solutions

Question	Marks	Sample Solutions	
Q1	a) i	1	6
	a) ii	1	sausage_allowance
	b) i	1	2
	b) ii	1	4
	c)	2	Warning = "You cannot have less than 6 people or more than 100 people attending the barbeque"
	d) i	2	The number of packs of sausages, loaves of bread, or cartons of drink do not divide equally into the total amount of each item required.
	d) ii	2	The returned number(s) would need to be rounded up to ensure that enough supplies are purchased.
Q2	a)	2	<p>Replace the "Extras" dropdown with one for each of required extras. E.g.</p> <p>When value selected from "Locker Hire" dropdown set locker_hire to value in "Locker Hire" dropdown</p> <p>Repeat above for towel and key hire, and add the following line 4.5 (or 27.5)</p> <pre>extras_fee = locker_hire + towel_hire + key_hire</pre>
	b)	2	<p>The variable initial_cost is only determined (and initialised) if coupon_value equals coupon_code.</p> <p>If they're not equal, line 37 has nothing to display.</p> <p>Line 32 should come before line 31.</p>
	c)	4	<p>After line 17 (in initially)...</p> <pre>set time = 9am set extras_fee = locker_hire + towel_hire</pre> <p>After line 24...</p> <p>When value selected from "Time" textfield</p> <pre>set time to value in the "Time" textfield if time is invalid display error message else if time >= 6am AND time <= 6pm set extras_fee = locker_hire + towel_hire else set extras_fee = security_key_hire endif endif</pre>



b)	11	<p>Initially</p> <pre> set name = "" set count = 0 set points = 0 set bonus = 0 set same = 0 set type1 = "" set size1 = "" (repeat for 2 and 3) When a value is entered into the "name" textfield set name to value in "name" textfield When a value is entered into the "treasure1" dropdown set treasure1 to value in "treasure1" dropdown (repeat for all dropdowns) When "calculate" button is pressed set count = 0 set points = 0 for each treasure if type <> "" count++ switch type case "ball": points += 10 case "coin": points += 20 case "marble": points += 30 end switch </pre>
----	----	--

			<pre> switch size case "small": points += 5 case "medium": points += 8 case "large": points += 11 end switch end if end for loop if (type1 == type 2 and size1 == size2 and type1 == type3 and size1 == size3) bonus = points * 0.3 same = 3 else if ((type1 == type2 and size1 == size2) or (type1 == type1 and size1 == size3) or (type2 == type3 and size2 == size3)) bonus = points * 0.2 same = 2 else bonus = 0 same = 0 end if points += bonus display name, count, same, points if bonus == 0 display "no bonus points" end if </pre>																								
Q4	a) i	1	7																								
	a) ii	1	8.0																								
	b) i	1	16																								
	b) ii	1	"Computer Science Exam"																								
	c)	2	10																								
	d)	2	<table border="1"> <tr><td></td><td></td></tr> <tr><td></td><td>25</td></tr> <tr><td>0</td><td></td></tr> <tr><td>1</td><td></td></tr> <tr><td>2</td><td></td></tr> <tr><td>3</td><td></td></tr> <tr><td>4</td><td></td></tr> <tr><td>5</td><td>23</td></tr> <tr><td>6</td><td>21</td></tr> <tr><td>7</td><td>19</td></tr> <tr><td>8</td><td></td></tr> <tr><td>9</td><td></td></tr> </table>				25	0		1		2		3		4		5	23	6	21	7	19	8		9	
	25																										
0																											
1																											
2																											
3																											
4																											
5	23																										
6	21																										
7	19																										
8																											
9																											
	e)	2	<p>Error 1: int 4g Explanation: A variable name cannot start with a digit.</p>																								

			<p>Error 2: <code>if (4g > 50);</code> Explanation: The semicolon terminates the if statement leading to a syntax error at following open brace.</p>																								
	f)	3	<table border="1"> <tr><td></td><td></td></tr> <tr><td>10</td><td>5</td></tr> <tr><td>20</td><td>10</td></tr> <tr><td>30</td><td>15</td></tr> <tr><td>40</td><td>20</td></tr> <tr><td>50</td><td>25</td></tr> <tr><td>60</td><td>120</td></tr> <tr><td>70</td><td>140</td></tr> <tr><td>80</td><td>160</td></tr> <tr><td>90</td><td>180</td></tr> <tr><td>100</td><td>200</td></tr> <tr><td>110</td><td></td></tr> </table> <p>Final value of i: 110 Final value of z: 200</p>			10	5	20	10	30	15	40	20	50	25	60	120	70	140	80	160	90	180	100	200	110	
10	5																										
20	10																										
30	15																										
40	20																										
50	25																										
60	120																										
70	140																										
80	160																										
90	180																										
100	200																										
110																											
	g)	3	<pre>public int question4g (int inputNumber) { int output; output = inputNumber; if ((output % 2) == 0) { output = output * 3; } else { output = output * 2; } return output; }</pre>																								
Q5	a) i	1	<p>Answer: A Explanation: The integer value "65" is cast to the Unicode / ASCII character "A"</p>																								
	a) ii	2	<p>Answer: 42.0 Explanation: The code has performed implicit casting (widening) from an integer value to a double.</p>																								
	b)	2	<p>Answer: 1 3 4 4 6 Explanation: The <code>Math.round()</code> method is used to process an array of floating-point numbers. Each number is rounded to the nearest integer and stored in a separate array, <code>secondArray</code>.</p>																								
	c)	3	<table border="1"> <tr><td></td><td></td></tr> <tr><td>0</td><td></td></tr> <tr><td>1</td><td>H</td></tr> <tr><td>2</td><td>HE</td></tr> <tr><td>3</td><td>HEL</td></tr> <tr><td>4</td><td>HELL</td></tr> </table> <p>Final value of <code>newWord</code> = "HELL"</p>			0		1	H	2	HE	3	HEL	4	HELL												
0																											
1	H																										
2	HE																										
3	HEL																										
4	HELL																										

Q6	a) i	2	<pre>public static void printScores(int[] scores) { System.out.print("Scores: "); for (int i = 0; i < scores.length; i++) { System.out.print(scores[i] + " "); } }</pre>
	a) ii	2	<pre>public static int calculateTotalScore(int[] scores) { int total = 0; for (int i=0; i < scores.length; i++){ total += scores[i]; } return total; }</pre>
	a) iii	2	<pre>public static int findHighestScore(int[] scores) { int highest = scores[0]; for (int i=0; i < scores.length; i++) { if (scores[i] > highest) { highest = scores[i]; } } return highest; }</pre>
	a) iv	2	<pre>public static double calculateAverageScore(int[] scores) { int total = calculateTotalScore(scores); return (double) total / scores.length; }</pre>
	b)	4	<p>This method sorts the values into ascending order and returns the median. It does this by stepping through the array comparing each element, <code>scoreArray[i]</code>, with each subsequent element, starting with <code>scoreArray[i+1]</code>, and swaps them if the other element is smaller. If the array has an odd number of values, it returns the middle value, otherwise it returns the average of the middle two values. In this example, it returns 56, since <code>scoreArray[7/2] = scoreArray[3] = 56</code>.</p>
Q7	a) i	1	6 or 7
	a) ii	1	3
	b) i	1	<code>Athlete a = new Athlete("Mary", "Jones", 20, "Female");</code>
	b) ii	1	<code>public Athlete(String newSurname, int newAge)</code>
	c)	2	<code>athlete1.setSurname("Jones");</code> <code>athlete1.setAge(25);</code>
	d)	2	<code>public void setMedals(int newMedals) {</code> <code> medals = newMedals;</code> <code>}</code>
	e)	4	<code>public void setCurrentMedals(int newMedals) {</code> <code> medals = newMedals;</code> <code>}</code> <code>public void setExtraMedals(int moreMedals) {</code> <code> medals += moreMedals;</code>

			<pre> } public int getTotalMedals() { return medals; } </pre>
Q8	a) i	2	<pre> Book book1 = new Book(1, "Cloudstreet", "Tim Winton", 1991); Book book2 = new Book(2, "True History of the Kelly Gang", "Peter Carey", 2000); </pre>
	a) ii	1	<pre> book1.displayDetails(); </pre>
	b)	3	<pre> public Book(int newBookID, String newTitle, String newAuthor, int newPublicationYear, String new AuthorCountry) { bookID = newBookID; title = newTitle; author = newAuthor; publicationYear = newPublicationYear; AuthorCountry = newAuthorCountry; } </pre>
	c)	3	<p>The code creates (declares) a 2D array of <code>Book</code> objects called <code>bookArray</code> with 2 rows and 2 columns, for a total of 4 elements. The code then fills (instantiates) the <code>bookArray</code> with 4 books.</p>
	d)	3	<p>You would need to create a new method in the named <code>displayBookArray</code> and pass through the newly created <code>bookArray</code> for processing. You would call up the method as:</p> <pre> displayBookArray(bookArray); </pre>
Q9		12	<pre> public class AFLScoreboard { private int team1Goals; private int team1Behinds; private int team2Goals; private int team2Behinds; private String team1Name; private String team2Name; public AFLScoreboard(String team1, String team2) { this.team1Goals = 0; this.team1Behinds = 0; this.team2Goals = 0; this.team2Behinds = 0; this.team1Name = team1; this.team2Name = team2; } public void addGoal(int team) { if (team == 1) { team1Goals++; } else if (team == 2) { team2Goals++; } } public void addBehind(int team) { </pre>

			<pre> if (team == 1) { team1Behinds++; } else if (team == 2) { team2Behinds++; } public int calculateTotalScore(int team) { if (team == 1) { return (team1Goals * 6) + team1Behinds; } else if (team == 2) { return (team2Goals * 6) + team2Behinds; } return 0; } public String displayDetails(int team) { if (team == 1) { return "Goals: " + team1Goals + ", Behinds: " + team1Behinds + ", Total: " + calculateTotalScore(1); } else if (team == 2) { return "Goals: " + team2Goals + ", Behinds: " + team2Behinds + ", Total: " + calculateTotalScore(2); } return ""; } </pre>					
Q10	a) i	1	$\sim A \vee B$					
	a) ii	1	$(\sim A \vee B) \wedge \sim C \equiv D$					
	b)	1						
	c) i	1	<table border="1"> <tr><td>\wedge</td></tr> <tr><td>F</td></tr> <tr><td>F</td></tr> <tr><td>F</td></tr> <tr><td>T</td></tr> </table>	\wedge	F	F	F	T
\wedge								
F								
F								
F								
T								
	c) ii	1	$\sim(A \wedge \sim B)$					
	d) i	1	A					
	d) ii	1	$(A \wedge A) \vee (A \wedge C)$ $\equiv A \vee (A \wedge C)$ $\equiv A$					
			L12 L34					

	e)	1																																													
			0	0	0	1	0	1	1																																						
			0	1	0	1	1	0	0																																						
			1	0	0	1	1	0	0																																						
			1	1	1	0	1	0	0																																						
	f) i	1	It helps to simplify Boolean expressions, reducing the number of logic gates needed in the design of a circuit.																																												
	f) ii	1	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td colspan="2"></td> <td colspan="2" style="text-align: center;">B</td> <td colspan="2"></td> </tr> <tr> <td colspan="2"></td> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td colspan="2"></td> </tr> <tr> <td rowspan="2" style="vertical-align: middle;">A</td> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td colspan="2"></td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td colspan="2"></td> </tr> </table> <p style="text-align: center;">$E \equiv B$</p>							B						0	1			A	0	0	1			1	0	1																			
		B																																													
		0	1																																												
A	0	0	1																																												
	1	0	1																																												
	g) i	1	A bus transfers data from one section of a computer to another.																																												
	g) ii	1	It limits the speed of data transfer.																																												
Q11	a) i	2	$F \equiv (\sim C \wedge D) \vee (\sim A \wedge \sim B \wedge \sim C) \vee (A \wedge \sim B \wedge D)$																																												
	a) ii	2	$F \equiv (\sim A \wedge \sim B) \vee (\sim B \wedge C \wedge D)$																																												
	b)	2	<p>Registers are temporary storage areas within the CPU that can be accessed directly by the ALU. Memory cells are located outside of the CPU.</p> <p>To perform arithmetic on a value in main memory, it must be first transferred to a register.</p>																																												
	c) i	1	10																																												
	c) ii	3	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td colspan="4" style="background-color: black;"></td> </tr> <tr> <td style="text-align: center;">01</td> <td style="text-align: center;">0008</td> <td style="text-align: center;">Data</td> <td>Mem[01] contains value 8</td> </tr> <tr> <td style="text-align: center;">02</td> <td style="text-align: center;">0005</td> <td style="text-align: center;">Data</td> <td>Mem[02] contains value 5</td> </tr> <tr> <td style="text-align: center;">03</td> <td style="text-align: center;">0000</td> <td style="text-align: center;">Data</td> <td>Mem[03] contains value 0</td> </tr> <tr> <td colspan="4" style="background-color: #f2f2f2;"></td> </tr> <tr> <td style="text-align: center;">10</td> <td style="text-align: center;">8A00</td> <td style="text-align: center;">RA f mem[00]</td> <td>Load the contents of memory address 01 into register A</td> </tr> <tr> <td style="text-align: center;">11</td> <td style="text-align: center;">8B01</td> <td style="text-align: center;">RB f mem[01]</td> <td>Load the contents of memory 02 into register B</td> </tr> <tr> <td style="text-align: center;">12</td> <td style="text-align: center;">1CAB</td> <td style="text-align: center;">RC f RA + RB</td> <td>Add the contents of registers A and B. Put the result into register C</td> </tr> <tr> <td style="text-align: center;">13</td> <td style="text-align: center;">9C02</td> <td style="text-align: center;">mem[03] f RC</td> <td>Store the contents of Register C into memory 03</td> </tr> <tr> <td style="text-align: center;">14</td> <td style="text-align: center;">0000</td> <td style="text-align: center;">halt</td> <td>Stop the machine</td> </tr> </table>									01	0008	Data	Mem[01] contains value 8	02	0005	Data	Mem[02] contains value 5	03	0000	Data	Mem[03] contains value 0					10	8A00	RA f mem[00]	Load the contents of memory address 01 into register A	11	8B01	RB f mem[01]	Load the contents of memory 02 into register B	12	1CAB	RC f RA + RB	Add the contents of registers A and B. Put the result into register C	13	9C02	mem[03] f RC	Store the contents of Register C into memory 03	14	0000	halt	Stop the machine
01	0008	Data	Mem[01] contains value 8																																												
02	0005	Data	Mem[02] contains value 5																																												
03	0000	Data	Mem[03] contains value 0																																												
10	8A00	RA f mem[00]	Load the contents of memory address 01 into register A																																												
11	8B01	RB f mem[01]	Load the contents of memory 02 into register B																																												
12	1CAB	RC f RA + RB	Add the contents of registers A and B. Put the result into register C																																												
13	9C02	mem[03] f RC	Store the contents of Register C into memory 03																																												
14	0000	halt	Stop the machine																																												
	c) iii	1	000D																																												
	c) iv	1	Register[0] always contains the value 0.																																												
Q12	a) i	2																																													
			10	0006																																											

			<table border="1"> <tr> <td>11</td> <td></td> <td>0009</td> <td></td> </tr> <tr> <td>12</td> <td></td> <td></td> <td>0000</td> </tr> </table> <p>The content of memory location 3 will be 0000₁₆</p>	11		0009		12			0000				
11		0009													
12			0000												
	a) ii	2	<table border="1"> <tr> <td>10</td> <td>000C</td> <td></td> <td></td> </tr> <tr> <td>11</td> <td></td> <td>0002</td> <td></td> </tr> <tr> <td>12</td> <td></td> <td></td> <td>0030</td> </tr> </table> <p>The contents of memory location 3 will be 0030₁₆ : $(12_{10} \times 2_{10} \times 2_{10}) = 48_{10} = 30_{16}$.</p>	10	000C			11		0002		12			0030
10	000C														
11		0002													
12			0030												
	b)	4	<pre>int a = 0; int b = 0; if ((a > 3) && (a < 5)) { b = 2; }</pre>												
	c)	4	<pre>01: 0000 10: 8A00 RA <- mem[00] 11: CA15 if (RA = 0) PC <-15 12: 9AFF write RA 13: 1AAA RA <- RA + RA 14: DA11 if (RA > 0) goto 11 (or C011 - PC <- 11) 15: 0000 halt</pre>												
Q13	a) i	1	100 1111 ₂												
	a) ii	1	193 ₁₀												
	b) i	2	428 ₁₀												
	b) ii	2	11.65625 ₁₀												
	c) i	2	1 0000 0100 ₂												
	c) ii	1	102 ₁₀												
	d)	2	0001 1100 ₂ Non-negative integers always start with a '0' and will have as many leading zeros as necessary to pad them out to the required number of bits.												
	e)	2	01111 1111 The most significant bit (MSB) in the 8-bit string is reserved for the sign. As we are referring to a positive integer the MSB is '0'. The remainder bits are completed as '1's giving a binary total of 127 ₁₀ .												
Q14	a)	2	-60 + -75 = -135 The calculation results in an integer overflow (giving a positive result), as the max/min range that can be stored in an 8-bit twos complement is -128 to +127 .												
	b)	3	56 - 12 = 56 + (-12) To get -12, take the binary representation of 12, invert, and add one. 12 in binar → 0000 1100 Invert the digits: 1's complement → 1111 0011 Add one. 2's complement → 1111 0100												

			<p>The last is the binary representation for -12. Add the numbers together.</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td style="background-color: black; color: white; text-align: center;">0011</td> <td style="background-color: black; color: white; text-align: center;">56</td> </tr> <tr> <td style="background-color: black; color: white; text-align: center;">1000</td> <td></td> </tr> <tr> <td style="background-color: #cccccc; text-align: center;">+ 1111</td> <td style="background-color: #cccccc; text-align: center;">-</td> </tr> <tr> <td style="background-color: #cccccc; text-align: center;">0100</td> <td style="background-color: #cccccc; text-align: center;">12</td> </tr> <tr> <td style="text-align: center;">0010</td> <td style="text-align: center;">44</td> </tr> <tr> <td style="text-align: center;">1100</td> <td></td> </tr> </table> <p>The result is 44, which is 56 - 12.</p>	0011	56	1000		+ 1111	-	0100	12	0010	44	1100	
0011	56														
1000															
+ 1111	-														
0100	12														
0010	44														
1100															
	c)	2	<p>Shifting right by n bits on an unsigned binary number has the effect of dividing it by 2^n (rounding towards 0).</p> <p>In this example the original number of 20_{10} becomes 10_{10} after the right shift.</p> <p>If the initial number is odd it will round down, as the least significant bit in the original is discarded.</p>												
	d)	2	<p>0.390625</p> $0.011001_2 = (0 \times 2^0) + (0 \times 2^{-1}) + (1 \times 2^{-2}) + (1 \times 2^{-3}) + (0 \times 2^{-4}) + (0 \times 2^{-5}) + (1 \times 2^{-6}) = (0.390625)_{10}$												
Q15	a)	3	<p>sign = 1</p> <p>exponent = 000101 = 5</p> <p>mantissa = 0.101... = 0.625</p> <p>decimal = -0.625×2^5 $= -0.625 \times 32$ $= -20$</p>												
	b)	4	<p>Values from 0 to 8120 requires 13 bits, since $2^{13} = 8192$</p> <p>Each pixel consists of $16 + 16 + 32 = 64$ squares</p> <p>Therefore, each pixel requires 64×13 bits = 832 bits</p> <p>The image will use $896 \times 640 \times 480 = 255\,590\,400$ bits</p>												
	c) i	2													
	c) ii	1	See above												

c) iii 3

AC00	3
AC01	AC04
AC02	AC07
AC03	AC0A
AC04	2
AC05	1
AC06	2
AC07	2
AC08	4
AC09	5
AC0A	2
AC0B	7
AC0C	9